

# Package ‘dashHtmlComponents’

October 13, 2022

**Title** Vanilla HTML Components for 'Dash'

**Version** 1.0.3

**Description** 'Dash' is a web application framework that provides pure Python and R abstraction around HTML, CSS, and JavaScript. Instead of writing HTML or using an HTML templating engine, you compose your layout using R functions within the 'dashHtmlComponents' package. The source for this package is on GitHub: [plotly/dash-html-components](https://github.com/plotly/dash-html-components).

**Depends** R (>= 3.0.2)

**Imports**

**Suggests** dash, dashCoreComponents, knitr, rmarkdown

**License** MIT + file LICENSE

**Copyright** Plotly Technologies, Inc.

**URL** <https://github.com/plotly/dash-html-components>

**BugReports** <https://github.com/plotly/dash-html-components/issues>

**Encoding** UTF-8

**LazyData** true

**VignetteBuilder** knitr

**KeepSource** true

**NeedsCompilation** no

**Author** Chris Parmer [aut],  
Ryan Patrick Kyle [cre] (<<https://orcid.org/0000-0002-4958-2844>>),  
Plotly Technologies, Inc. [cph]

**Maintainer** Ryan Patrick Kyle <[ryan@plotly.com](mailto:ryan@plotly.com)>

**Repository** CRAN

**Date/Publication** 2020-05-06 22:00:02 UTC

**R topics documented:**

dashHtmlComponents-package . . . . .	5
htmlA . . . . .	5
htmlAbbr . . . . .	7
htmlAcronym . . . . .	9
htmlAddress . . . . .	11
htmlArea . . . . .	12
htmlArticle . . . . .	15
htmlAside . . . . .	17
htmlAudio . . . . .	19
htmlB . . . . .	21
htmlBase . . . . .	23
htmlBasefont . . . . .	25
htmlBdi . . . . .	27
htmlBdo . . . . .	28
htmlBig . . . . .	30
htmlBlink . . . . .	32
htmlBlockquote . . . . .	34
htmlBr . . . . .	36
htmlButton . . . . .	38
htmlCanvas . . . . .	40
htmlCaption . . . . .	42
htmlCenter . . . . .	44
htmlCite . . . . .	46
htmlCode . . . . .	48
htmlCol . . . . .	50
htmlColgroup . . . . .	52
htmlCommand . . . . .	54
htmlContent . . . . .	56
htmlData . . . . .	57
htmlDatalist . . . . .	59
htmlDd . . . . .	61
htmlDel . . . . .	63
htmlDetails . . . . .	65
htmlDfn . . . . .	67
htmlDialog . . . . .	69
htmlDiv . . . . .	70
htmlDl . . . . .	72
htmlDt . . . . .	74
htmlElement . . . . .	76
htmlEm . . . . .	77
htmlEmbed . . . . .	79
htmlFieldset . . . . .	81
htmlFigcaption . . . . .	83
htmlFigure . . . . .	85
htmlFont . . . . .	87
htmlFooter . . . . .	88

htmlForm	90
htmlFrame	93
htmlFrameset	94
htmlH1	96
htmlH2	98
htmlH3	99
htmlH4	101
htmlH5	103
htmlH6	105
htmlHeader	107
htmlHgroup	109
htmlHr	110
htmlI	112
htmlIframe	114
htmlImg	116
htmlIns	118
htmlIsindex	120
htmlKbd	121
htmlKeygen	123
htmlLabel	125
htmlLegend	127
htmlLi	129
htmlLink	131
htmlListing	133
htmlMain	135
htmlMapEl	137
htmlMark	139
htmlMarquee	141
htmlMeta	143
htmlMeter	145
htmlMulticol	147
htmlNav	148
htmlNextid	150
htmlNobr	152
htmlNoscript	154
htmlObjectEl	155
htmlOl	158
htmlOptgroup	160
htmlOption	162
htmlOutput	164
htmlP	165
htmlParam	167
htmlPicture	169
htmlPlaintext	171
htmlPre	173
htmlProgress	175
htmlQ	177
htmlRb	178

htmlRp	180
htmlRt	182
htmlRtc	184
htmlRuby	186
htmlS	188
htmlSamp	190
htmlScript	191
htmlSection	193
htmlSelect	195
htmlShadow	197
htmlSlot	199
htmlSmall	200
htmlSource	202
htmlSpacer	204
htmlSpan	206
htmlStrike	208
htmlStrong	210
htmlSub	211
htmlSummary	213
htmlSup	215
htmlTable	217
htmlTbody	219
htmlTd	221
htmlTemplate	223
htmlTextarea	225
htmlTfoot	228
htmlTh	230
htmlThead	232
htmlTime	234
htmlTitle	236
htmlTr	238
htmlTrack	240
htmlU	242
htmlUl	244
htmlVar	246
htmlVideo	247
htmlWbr	250
htmlXmp	252

---

dashHtmlComponents-package

*Vanilla HTML Components for 'Dash'*


---

### Description

'Dash' is a web application framework that provides pure Python and R abstraction around HTML, CSS, and JavaScript. Instead of writing HTML or using an HTML templating engine, you compose your layout using R functions within the 'dashHtmlComponents' package. The source for this package is on GitHub: [plotly/dash-html-components](https://github.com/plotly/dash-html-components).

### Author(s)

**Maintainer:** Ryan Patrick Kyle <[ryan@plotly.com](mailto:ryan@plotly.com)>

---

htmlA

*A component*


---

### Description

A is a wrapper for the <a> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/a>

### Usage

```
htmlA(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, download=NULL,
href=NULL, hrefLang=NULL, media=NULL, rel=NULL, shape=NULL,
target=NULL, accessKey=NULL, className=NULL,
contentEditable=NULL, contextMenu=NULL, dir=NULL,
draggable=NULL, hidden=NULL, lang=NULL, spellCheck=NULL,
style=NULL, tabIndex=NULL, title=NULL, loading_state=NULL,
...)
```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.

n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
download	Character. Indicates that the hyperlink is to be used for downloading a resource.
href	Character. The URL of a linked resource.
hrefLang	Character. Specifies the language of the linked resource.
media	Character. Specifies a hint of the media for which the linked resource was designed.
rel	Character. Specifies the relationship of the target object to the link object.
shape	Character.
target	Character.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlA(children='Link to external site',
             href='https://plotly.com',
             target='_blank')
    )
  )
)

app$run_server()
}
```

---

htmlAbbr

*Abbr component*


---

**Description**

Abbr is a wrapper for the <abbr> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/abbr>

**Usage**

```
htmlAbbr(children=NULL, id=NULL, n_clicks=NULL,
          n_clicks_timestamp=NULL, key=NULL, role=NULL,
          accessKey=NULL, className=NULL, contentEditable=NULL,
          contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
          lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
          title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.

n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()
```



```

    app$layout(
      htmlDiv(list(
        htmlAbbr(children='Hello! htmlAbbr at work!',
          title='\U{1F50D} Hover over this line for a few seconds and see the text box appear...')
        )
      )
    )

    app$run_server()
  }

```

---

htmlAcronym

*Acronym component*


---

## Description

Acronym is a wrapper for the `<acronym>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/acronym>

## Usage

```

htmlAcronym(children=NULL, id=NULL, n_clicks=NULL,
  n_clicks_timestamp=NULL, key=NULL, role=NULL,
  accessKey=NULL, className=NULL, contentEditable=NULL,
  contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
  lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
  title=NULL, loading_state=NULL, ...)

```

## Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.

contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlAcronym(children='ASAP',
        title='Mouse over these words to see the acronym for \'as soon as possible\'.')
    )
  )
)

app$run_server()
}

```

---

htmlAddress	<i>Address component</i>
-------------	--------------------------

---

### Description

Address is a wrapper for the <address> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/address>

### Usage

```
htmlAddress(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.

lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlAddress(children='5555 Avenue de Gaspe, Montreal QC H2T 2A3')
    )
  )
)

app$run_server()
}

```

---

htmlArea

*Area component*


---

**Description**

Area is a wrapper for the <area> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/area>

**Usage**

```
htmlArea(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, alt=NULL,
coords=NULL, download=NULL, href=NULL, hrefLang=NULL,
media=NULL, rel=NULL, shape=NULL, target=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in call-backs. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
alt	Character. Alternative text in case an image can't be displayed.
coords	Character. A set of values specifying the coordinates of the hot-spot region.
download	Character. Indicates that the hyperlink is to be used for downloading a resource.
href	Character. The URL of a linked resource.
hrefLang	Character. Specifies the language of the linked resource.
media	Character. Specifies a hint of the media for which the linked resource was designed.
rel	Character. Specifies the relationship of the target object to the link object.
shape	Character.
target	Character.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.

<code>dir</code>	Character. Defines the text direction. Allowed values are <code>ltr</code> (Left-To-Right) or <code>rtl</code> (Right-To-Left)
<code>draggable</code>	Character. Defines whether the element can be dragged.
<code>hidden</code>	A value equal to: <code>'hidden'</code> , <code>'hidden'   logical</code> . Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
<code>lang</code>	Character. Defines the language used in the element.
<code>spellCheck</code>	Character. Indicates whether spell checking is allowed for the element.
<code>style</code>	Named list. Defines CSS styles which will override styles previously set.
<code>tabIndex</code>	Character. Overrides the browser's default tab order and follows the one specified instead.
<code>title</code>	Character. Text to be displayed in a tooltip when hovering over the element.
<code>loading_state</code>	Lists containing elements <code>'is_loading'</code> , <code>'prop_name'</code> , <code>'component_name'</code> . those elements have the following types: - <code>is_loading</code> (logical; optional): determines if the component is loading or not - <code>prop_name</code> (character; optional): holds which property is loading - <code>component_name</code> (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
<code>...</code>	wildcards allowed have the form: <code>'data-*'</code> , <code>'aria-*'</code>

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
# The URL below has been chunked to comply with CRAN
# requirements; the use of file.path is optional and not required
# for this component.
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app$layout(
    htmlDiv(list(
      htmlImg(src = file.path('https://upload.wikimedia.org',
        'wikipedia/commons/0/0c',
        'PIA17351-ApparentSizes-MarsDeimosPhobos-EarthMoon.jpg'),
        fsep = '/'),
      useMap = '#image-map'),
    htmlMapEl(list(
      htmlArea(target='_blank',
        alt='Deimos',
        title='Deimos',
        href='https://en.wikipedia.org/wiki/Deimos_(moon)',
        coords='5,114,32,147',
        shape='rect'),
      htmlArea(target='_blank',
        alt='Phobos',
```

```

        title='Phobos',
        href='https://en.wikipedia.org/wiki/Phobos_(moon)',
        coords='113,196,32,103',
        shape='rect'),
    htmlArea(target='_blank',
            alt='Moon',
            title='Moon',
            href='https://en.wikipedia.org/wiki/Moon',
            coords='127,285,294,1',
            shape='rect')
    ),
    name = 'image-map'
  ),
  htmlDiv(children = 'Click on the image to visit a Wikipedia article',
         id = 'object-name')
  )
)
)
app$run_server()
}

```

---

htmlArticle

*Article component*


---

## Description

Article is a wrapper for the `<article>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/article>

## Usage

```

htmlArticle(children=NULL, id=NULL, n_clicks=NULL,
            n_clicks_timestamp=NULL, key=NULL, role=NULL,
            accessKey=NULL, className=NULL, contentEditable=NULL,
            contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
            lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
            title=NULL, loading_state=NULL, ...)

```

## Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.

n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()
```



```

app$layout(
  htmlDiv(list(
    htmlArticle(list(
      htmlH2('Dash for R launched!'),
      htmlP('Dash is a user interface library for creating analytical\n
web applications. Those who use R for data analysis, data\n
exploration, visualization, modelling, instrument control,\n
and reporting will find immediate use for Dash for R.'),
      htmlAside('Plotly is a technical computing company with offices\n
in Montreal, Canada and Cambridge, Massachusetts.')
    )
  )
)
)
)
)
app$run_server()
}

```

---

htmlAside

*Aside component*


---

### Description

Aside is a wrapper for the <aside> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/aside>

### Usage

```

htmlAside(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.

key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlArticle(list(
        htmlH2('Dash for R launched!'),
```

```

        htmlP('Dash is a user interface library for creating analytical\n
              web applications. Those who use R for data analysis, data\n
              exploration, visualization, modelling, instrument control,\n
              and reporting will find immediate use for Dash for R. '),
        htmlAside('Plotly is a technical computing company with offices\n
                  in Montreal, Canada and Cambridge, Massachusetts. ')
    )
  )
)
)
)
app$run_server()
}

```

---

htmlAudio

*Audio component*


---

### Description

Audio is a wrapper for the <audio> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/audio>

### Usage

```

htmlAudio(children=NULL, id=NULL, n_clicks=NULL,
           n_clicks_timestamp=NULL, key=NULL, role=NULL, autoPlay=NULL,
           controls=NULL, crossOrigin=NULL, loop=NULL, muted=NULL,
           preload=NULL, src=NULL, accessKey=NULL, className=NULL,
           contentEditable=NULL, contextMenu=NULL, dir=NULL,
           draggable=NULL, hidden=NULL, lang=NULL, spellCheck=NULL,
           style=NULL, tabIndex=NULL, title=NULL, loading_state=NULL,
           ...)

```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.

key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
autoPlay	A value equal to: 'autoplay', 'autoplay', 'autoplay'   logical. The audio or video should play as soon as possible.
controls	A value equal to: 'controls', 'controls'   logical. Indicates whether the browser should show playback controls to the user.
crossOrigin	Character. How the element handles cross-origin requests
loop	A value equal to: 'loop', 'loop'   logical. Indicates whether the media should start playing from the start when it's finished.
muted	A value equal to: 'muted', 'muted'   logical. Indicates whether the audio will be initially silenced on page load.
preload	Character. Indicates whether the whole resource, parts of it or nothing should be preloaded.
src	Character. The URL of the embeddable content.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlAudio(src='https://www.nasa.gov/62284main_onesmall2.wav',
                controls=TRUE,
                title='Apollo 11 - July 16, 1969 - Neil Armstrong')
    )
  )
)

app$run_server()
}
```

---

htmlB

*B component*


---

**Description**

B is a wrapper for the <b> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/b>

**Usage**

```
htmlB(children=NULL, id=NULL, n_clicks=NULL,
       n_clicks_timestamp=NULL, key=NULL, role=NULL,
       accessKey=NULL, className=NULL, contentEditable=NULL,
       contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
       lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
       title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in call-backs. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.

n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()
```

```

    app$layout(
      htmlDiv(list(
        htmlB(children="This is a bold statement!"),
        htmlP(children="This is not so bold.")
      )
    )
  )
  app$run_server()
}

```

---

htmlBase

*Base component*


---

### Description

Base is a wrapper for the <base> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/base>

### Usage

```

htmlBase(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, href=NULL,
target=NULL, accessKey=NULL, className=NULL,
contentEditable=NULL, contextMenu=NULL, dir=NULL,
draggable=NULL, hidden=NULL, lang=NULL, spellCheck=NULL,
style=NULL, tabIndex=NULL, title=NULL, loading_state=NULL,
...)

```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
href	Character. The URL of a linked resource.

target	Character.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlBase(href="https://www.w3schools.com/"),
      htmlA(children="HTML base tag", href="tags/tag_base.asp")
    )
  )
)

app$run_server()
}

```



---

htmlBasefont	<i>Basefont component</i>
--------------	---------------------------

---

### Description

Basefont is a wrapper for the <basefont> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/basefont>

### Usage

```
htmlBasefont(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.

lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  # This feature is obsolete. It may still work in some
  # browsers, but could stop working at any time. Try to
  # avoid using this component.
  #
  # Instead, use CSS properties to set font, font-family,
  # font-size and color.
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlBasefont(color="FF0000",
                    face="Helvetica",
                    size="+2"),
      htmlP(children="If it works, this will be Helvetica but a couple point sizes larger.")
    )
  )
)

app$run_server()
}
```

---

htmlBdi	<i>Bdi component</i>
---------	----------------------

---

### Description

Bdi is a wrapper for the `<bdi>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/bdi>

### Usage

```
htmlBdi(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <code>&lt;menu&gt;</code> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are <code>ltr</code> (Left-To-Right) or <code>rtl</code> (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: <code>'hidden'</code> , <code>'hidden'   logical</code> . Prevents rendering of given element, while keeping child elements, e.g. script elements, active.

lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlP(children="This text is 'Aladdin', but in Arabic script:"),
      htmlBdi(children=paste0("\U{0639}\U{0644}\U{0627}\U{0621}",
                             "\U{0627}\U{0644}\U{062F}\U{064A}\U{0646}"))
    )
  )
)

app$run_server()
}

```

---

htmlBdo

*Bdo component*


---

**Description**

Bdo is a wrapper for the <bdo> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/bdo>

**Usage**

```
htmlBdo(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.

loading\_state Lists containing elements 'is\_loading', 'prop\_name', 'component\_name'. those elements have the following types: - is\_loading (logical; optional): determines if the component is loading or not - prop\_name (character; optional): holds which property is loading - component\_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer

... wildcards allowed have the form: "data-\*", "aria-\*"

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  # This element supports bidirectional text override.
  # We can force text to render from right to left instead
  # of left to right.
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlP(children="This text will print from left to right."),
      htmlP(children="Below, we use bidirectional override to print right to left:"),
      htmlBdo(children="This text will print from right to left.",
        dir="rtl")
    )
  )
)

app$run_server()
}
```

---

htmlBig

*Big component*

---

### Description

Big is a wrapper for the <big> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/big>

### Usage

```
htmlBig(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
```

contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL, lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL, title=NULL, loading\_state=NULL, ...)

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  # The <big> tag is not supported in HTML5.
  # Instead, use the font-size property in
  # CSS to enlarge text.
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlP(children="This text will display in standard size."),
      htmlBig(children="This text may or may not appear slightly larger.")
    )
  )
  )

  app$run_server()
}
```

---

htmlBlink

*Blink component*


---

**Description**

Blink is a wrapper for the `<blink>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/blink>

**Usage**

```
htmlBlink(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.



n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  # The blink tag is now obsolete and deprecated.
  # It may not function properly in all browsers,
```

```

# and it may cease working without warning.
#
# This element is generally unsupported on all
# modern browser releases.
library(dash)
library(dashHtmlComponents)

app <- Dash$new()

app$layout(
  htmlDiv(list(
    htmlP(children="Here is a bit of text."),
    htmlBlink(children="Here is a bit of blinking text.")
  )
)
)

app$run_server()
}

```

---

htmlBlockquote

*Blockquote component*


---

## Description

Blockquote is a wrapper for the <blockquote> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/blockquote>

## Usage

```

htmlBlockquote(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, cite=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

## Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.

key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
cite	Character. Contains a URI which points to the source of the quote or change.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlP("Here is some text."),
```

```

    htmlBlockquote(children=list(
      htmlP("And here is a quotation in block format.")
    )
  )
)
)
)
)
app$run_server()
}

```

htmlBr

*Br component***Description**

Br is a wrapper for the <br> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/br>

**Usage**

```

htmlBr(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.

contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlP("Here is some text."),
      htmlBr(),
      htmlP("Here is additional text."),
      htmlBr(),
      htmlP("See the gap in between the lines?")
    )
  )
)

app$run_server()
}

```

htmlButton

*Button component***Description**

Button is a wrapper for the `<button>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/button>

**Usage**

```
htmlButton(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
autoFocus=NULL, disabled=NULL, form=NULL, formAction=NULL,
formEncType=NULL, formMethod=NULL, formNoValidate=NULL,
formTarget=NULL, name=NULL, type=NULL, value=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
autoFocus	A value equal to: 'autofocus', 'autofocus', 'autofocus'   logical. The element should be automatically focused after the page loaded.
disabled	A value equal to: 'disabled', 'disabled'   logical. Indicates whether the user can interact with the element.
form	Character. Indicates the form that is the owner of the element.
formAction	Character. Indicates the action of the element, overriding the action defined in the <code>&lt;form&gt;</code> .

formEncType	Character. If the button/input is a submit button (type="submit"), this attribute sets the encoding type to use during form submission. If this attribute is specified, it overrides the enctype attribute of the button's form owner.
formMethod	Character. If the button/input is a submit button (type="submit"), this attribute sets the submission method to use during form submission (GET, POST, etc.). If this attribute is specified, it overrides the method attribute of the button's form owner.
formNoValidate	A value equal to: 'formnovalidate', 'formnovalidate', 'formnovalidate'   logical. If the button/input is a submit button (type="submit"), this boolean attribute specifies that the form is not to be validated when it is submitted. If this attribute is specified, it overrides the novalidate attribute of the button's form owner.
formTarget	Character. If the button/input is a submit button (type="submit"), this attribute specifies the browsing context (for example, tab, window, or inline frame) in which to display the response that is received after submitting the form. If this attribute is specified, it overrides the target attribute of the button's form owner.
name	Character. Name of the element. For example used by the server to identify the fields in form submits.
type	Character. Defines the type of the element.
value	Character. Defines a default value which will be displayed in the element on page load.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlButton("Click me!")
    ))
  )

  app$run_server()
}

```

---

htmlCanvas

*Canvas component*


---

**Description**

Canvas is a wrapper for the <canvas> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/canvas>

**Usage**

```

htmlCanvas(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, height=NULL,
width=NULL, accessKey=NULL, className=NULL,
contentEditable=NULL, contextMenu=NULL, dir=NULL,
draggable=NULL, hidden=NULL, lang=NULL, spellCheck=NULL,
style=NULL, tabIndex=NULL, title=NULL, loading_state=NULL,
...)

```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.



n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
height	Character   numeric. Specifies the height of elements listed here. For all other elements, use the CSS height property. Note: In some instances, such as <div>, this is a legacy attribute, in which case the CSS height property should be used instead.
width	Character   numeric. For the elements listed here, this establishes the element's width. Note: For all other instances, such as <div>, this is a legacy attribute, in which case the CSS width property should be used instead.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  # this component requires JavaScript code to draw on the canvas
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlCanvas(id="canvas-component")
    )
  )
)

app$run_server()
}

```

---

htmlCaption

*Caption component*


---

**Description**

Caption is a wrapper for the <caption> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/caption>

**Usage**

```

htmlCaption(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.

key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlTable(list(
        htmlCaption("Elevations of a few Cascade Range volcanoes"),
```

```

htmlHead(
  htmlTr(list(
    htmlTh("Mountain"),
    htmlTh("Elevation (m)"),
    htmlTh("Elevation (ft)")
  )
),
htmlBody(list(
  htmlTr(list(
    htmlTd("Mount Rainier"),
    htmlTd("4,392"),
    htmlTd("14,411")
  )
),
  htmlTr(list(
    htmlTd("Mount Hood"),
    htmlTd("3,429"),
    htmlTd("11,249")
  )
),
  htmlTr(list(
    htmlTd("Lassen Peak"),
    htmlTd("3,187"),
    htmlTd("10,457")
  )
),
  htmlTr(list(
    htmlTd("Mount St. Helens"),
    htmlTd("2,549"),
    htmlTd("8,363")
  )
)
), style = list(
  border = "1px black solid"
)
)
)
)
)
app$run_server()
}

```

**Description**

Center is a wrapper for the <center> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/center>

**Usage**

```
htmlCenter(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.

tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlCenter("Centered text!")
    )
  )
)

app$run_server()
}
```

---

htmlCite

*Cite component*


---

**Description**

Cite is a wrapper for the <cite> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/cite>

**Usage**

```
htmlCite(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlCite("Click me!")
    )
  )
)

app$run_server()
}

```

htmlCode

*Code component***Description**

Code is a wrapper for the `<code>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/code>

**Usage**

```

htmlCode(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.



key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(htmlDiv(list(
    htmlCode(
      children = 'cat("Hello world!")'
    )
  )))
}
```

```

    )
  )
)

app$run_server()
}

```

---

htmlCol

*Col component*


---

### Description

Col is a wrapper for the `<col>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/col>

### Usage

```
htmlCol(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, span=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
span	Character.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.

contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  # Used within htmlColgroup to define columns.
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlTable(list(
        htmlColgroup(
          list(
            htmlCol(span = 2, style = list("background-color"= "red"))
          )
        ),
        htmlTr(
          list(
            htmlTd("Cell A"),
            htmlTd("Cell B"),
            htmlTd("Cell C")
          )
        )
      )
    )
  )
}
```

```

    )
  ))
)
)
)
  app$run_server()
}

```

---

htmlColgroup

*Colgroup component*


---

### Description

Colgroup is a wrapper for the `<colgroup>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/colgroup>

### Usage

```

htmlColgroup(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, span=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
span	Character.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.

contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: ``data-*`, `aria-*``

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlTable(list(
        htmlColgroup(
          list(
            htmlCol(span = 2, style = list("background-color"= "red"))
          )
        ),
      htmlTr(
        list(
          htmlTd("Cell A"),
          htmlTd("Cell B"),
```

```

        htmlTd("Cell C")
      )
    ))
  )
)
app$run_server()
}

```

---

htmlCommand

*Command component*


---

### Description

Command is a wrapper for the `<command>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/command>

### Usage

```

htmlCommand(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, checked=NULL,
disabled=NULL, icon=NULL, radioGroup=NULL, type=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
checked	A value equal to: 'checked', 'checked'   logical. Indicates whether the element should be checked on page load.

disabled	A value equal to: 'disabled', 'disabled'   logical. Indicates whether the user can interact with the element.
icon	Character. Specifies a picture which represents the command.
radioGroup	Character.
type	Character. Defines the type of the element.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
# This component is deprecated and its use is no longer recommended.
```

htmlContent

*Content component***Description**

Content is a wrapper for the <content> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/content>

**Usage**

```
htmlContent(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.



lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
# This feature is obsolete and no longer supported. It is recommended
# that you use the htmlSlot component instead.
```

---

htmlData	<i>Data component</i>
----------	-----------------------

---

**Description**

Data is a wrapper for the <data> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/data>

**Usage**

```
htmlData(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, value=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
value	Character. Defines a default value which will be displayed in the element on page load.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: "data-*", "aria-*"

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlUL(list(
        htmlLi(list(htmlData(value = 398, "First Element"))),
        htmlLi(list(htmlData(value = 399, "Second Element"))),
        htmlLi(list(htmlData(value = 400, "First Element"))))
      ))
    )
  )

  # Include the following in a seperate CSS file in an
  # `assets` directory in the root of your app.
  #
  # data: hover::after {
  #   content: ' (ID ' attr(value) ')';
  #   font-size: .7em;
  # }

  app$run_server()
}

```

---

htmlDatalist

*Datalist component*


---

**Description**

Datalist is a wrapper for the <datalist> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/datalist>

**Usage**

```

htmlDatalist(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)
  library(dashCoreComponents)

  app <- Dash$new()

  app$layout(htmlDiv(list(
    dccInput(
      placeholder = 'Enter here',
      list = 'list-of-options'),
    htmlDatalist(id = 'list-of-options',
      children=list(
        htmlOption("Option 1"),
        htmlOption("Option 2"),
        htmlOption("Option 3")
      )
    )
  )
)
)
)
)

app$run_server()
}

```

htmlDd

*Dd component***Description**

Dd is a wrapper for the <dd> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/dd>

**Usage**

```

htmlDd(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.

<code>n_clicks</code>	Numeric. An integer that represents the number of times that this element has been clicked on.
<code>n_clicks_timestamp</code>	Numeric. An integer that represents the time (in ms since 1970) at which <code>n_clicks</code> changed. This can be used to tell which button was changed most recently.
<code>key</code>	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
<code>role</code>	Character. The ARIA role attribute
<code>accessKey</code>	Character. Keyboard shortcut to activate or add focus to the element.
<code>className</code>	Character. Often used with CSS to style elements with common properties.
<code>contentEditable</code>	Character. Indicates whether the element's content is editable.
<code>contextMenu</code>	Character. Defines the ID of a <code>&lt;menu&gt;</code> element which will serve as the element's context menu.
<code>dir</code>	Character. Defines the text direction. Allowed values are <code>ltr</code> (Left-To-Right) or <code>rtl</code> (Right-To-Left)
<code>draggable</code>	Character. Defines whether the element can be dragged.
<code>hidden</code>	A value equal to: <code>'hidden'</code> , <code>'hidden'   logical</code> . Prevents rendering of given element, while keeping child elements, e.g. <code>script</code> elements, active.
<code>lang</code>	Character. Defines the language used in the element.
<code>spellCheck</code>	Character. Indicates whether spell checking is allowed for the element.
<code>style</code>	Named list. Defines CSS styles which will override styles previously set.
<code>tabIndex</code>	Character. Overrides the browser's default tab order and follows the one specified instead.
<code>title</code>	Character. Text to be displayed in a tooltip when hovering over the element.
<code>loading_state</code>	Lists containing elements <code>'is_loading'</code> , <code>'prop_name'</code> , <code>'component_name'</code> . those elements have the following types: <code>- is_loading</code> (logical; optional): determines if the component is loading or not - <code>prop_name</code> (character; optional): holds which property is loading - <code>component_name</code> (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from <code>dash-renderer</code>
<code>...</code>	wildcards allowed have the form: <code>'data-*'</code> , <code>'aria-*'</code>

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)
```

```

app <- Dash$new()

app$layout(htmlDiv(list(
  htmlDl(
    children= list(htmlDt("Dash for R"),
                  htmlDd('HtmlDt and htmlDD must be used
                        within htmlDl'))
  )
)
)
)

app$run_server()
}

```

---

htmlDel

*Del component*


---

### Description

Del is a wrapper for the `<del>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/del>

### Usage

```

htmlDel(children=NULL, id=NULL, n_clicks=NULL,
         n_clicks_timestamp=NULL, key=NULL, role=NULL, cite=NULL,
         dateTime=NULL, accessKey=NULL, className=NULL,
         contentEditable=NULL, contextMenu=NULL, dir=NULL,
         draggable=NULL, hidden=NULL, lang=NULL, spellCheck=NULL,
         style=NULL, tabIndex=NULL, title=NULL, loading_state=NULL,
         ...)

```

### Arguments

<code>children</code>	A list of or a singular dash component, string or number. The children of this component
<code>id</code>	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
<code>n_clicks</code>	Numeric. An integer that represents the number of times that this element has been clicked on.
<code>n_clicks_timestamp</code>	Numeric. An integer that represents the time (in ms since 1970) at which <code>n_clicks</code> changed. This can be used to tell which button was changed most recently.

key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
cite	Character. Contains a URI which points to the source of the quote or change.
dateTime	Character. Indicates the date and time associated with the element.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(htmlDiv(list(
```



```

    htmlDel(
      children ="Deleted Hello"
    )
  )
)
)
app$run_server()
}

```

---

htmlDetails

*Details component*


---

### Description

Details is a wrapper for the <details> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/details>

### Usage

```

htmlDetails(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, open=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
open	A value equal to: 'open', 'open'   logical. Indicates whether the details will be shown on page load.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.

className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlDetails(
        children = list(
          htmlSummary(
            children = "Within a details element, the summary can act as a clickable description"
          ),
          "And the rest is hidden until the summary is clicked"
        )
      )
    )
  )
}

```

```

    )
  )

  app$run_server()
}

```

htmlDfn

*Dfn component***Description**

Dfn is a wrapper for the <dfn> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/dfn>

**Usage**

```

htmlDfn(children=NULL, id=NULL, n_clicks=NULL,
         n_clicks_timestamp=NULL, key=NULL, role=NULL,
         accessKey=NULL, className=NULL, contentEditable=NULL,
         contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
         lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
         title=NULL, loading_state=NULL, ...)

```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.

<code>dir</code>	Character. Defines the text direction. Allowed values are <code>ltr</code> (Left-To-Right) or <code>rtl</code> (Right-To-Left)
<code>draggable</code>	Character. Defines whether the element can be dragged.
<code>hidden</code>	A value equal to: <code>'hidden'</code> , <code>'hidden'   logical</code> . Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
<code>lang</code>	Character. Defines the language used in the element.
<code>spellCheck</code>	Character. Indicates whether spell checking is allowed for the element.
<code>style</code>	Named list. Defines CSS styles which will override styles previously set.
<code>tabIndex</code>	Character. Overrides the browser's default tab order and follows the one specified instead.
<code>title</code>	Character. Text to be displayed in a tooltip when hovering over the element.
<code>loading_state</code>	Lists containing elements <code>'is_loading'</code> , <code>'prop_name'</code> , <code>'component_name'</code> . those elements have the following types: - <code>is_loading</code> (logical; optional): determines if the component is loading or not - <code>prop_name</code> (character; optional): holds which property is loading - <code>component_name</code> (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
<code>...</code>	wildcards allowed have the form: <code>''data-*'</code> , <code>'aria-*'</code>

## Value

named list of JSON elements corresponding to React.js properties and their values

## Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(htmlDiv(list(
    htmlDfn(
      children ="Hello"
    )
  )
))
)

app$run_server()
}
```

---

htmlDialog	<i>Dialog component</i>
------------	-------------------------

---

### Description

Dialog is a wrapper for the <dialog> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/dialog>

### Usage

```
htmlDialog(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.

lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: "data-*", "aria-*"

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(htmlDiv(list(
    htmlDialog(
      children = htmlP('Greetings')
    )
  )
))
)
)

app$run_server()
}

```

---

htmlDiv

*Div component*


---

**Description**

Div is a wrapper for the <div> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/div>

**Usage**

```
htmlDiv(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.

loading\_state Lists containing elements 'is\_loading', 'prop\_name', 'component\_name'. those elements have the following types: - is\_loading (logical; optional): determines if the component is loading or not - prop\_name (character; optional): holds which property is loading - component\_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer

... wildcards allowed have the form: 'data-\*', 'aria-\*'

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlDiv('This Title is Wrapped inside an inner Div')
    )
  )
)

app$run_server()
}
```

---

htmlDl

*Dl component*

---

### Description

Dl is a wrapper for the <dl> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/dl>

### Usage

```
htmlDl(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```



**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(htmlDiv(list(
    htmlDl(
      children= list(htmlDt("Dash for R"),
                    htmlDd('HtmlDt and htmlDD must be used
                           within htmlDl'))
    )
  )
  )
  )
  app$run_server()
}

```

---

*htmlDt**Dt component*

---

**Description**

Dt is a wrapper for the <dt> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/dt>

**Usage**

```

htmlDt(children=NULL, id=NULL, n_clicks=NULL,
        n_clicks_timestamp=NULL, key=NULL, role=NULL,
        accessKey=NULL, className=NULL, contentEditable=NULL,
        contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
        lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
        title=NULL, loading_state=NULL, ...)

```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in call-backs. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.

n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()
```

```

app$layout(htmlDiv(list(
  htmlDl(
    children= list(htmlDt("Dash for R"),
                  htmlDd('HtmlDt and htmlDD must be used
                        within htmlDl'))
    )
  )
)
)

app$run_server()
}

```

---

htmlElement

*Element component*


---

### Description

Element is a wrapper for the <element> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/element>

### Usage

```

htmlElement(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.

className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
# This feature is obsolete. It may still work in some
# browsers, but could stop working at any time. Try to
# avoid using this component.
```

---

htmlEm

*Em component*


---

**Description**

Em is a wrapper for the <em> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/em>

**Usage**

```
htmlEm(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.

loading\_state Lists containing elements 'is\_loading', 'prop\_name', 'component\_name'. those elements have the following types: - is\_loading (logical; optional): determines if the component is loading or not - prop\_name (character; optional): holds which property is loading - component\_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer

... wildcards allowed have the form: 'data-\*', 'aria-\*'

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(htmlDiv(list(
    htmlH1(list('Dash is a very ',
               htmlEm(' important '),
               'framework')
    )
  )
)
)
app$run_server()
}
```

---

htmlEmbed

*Embed component*

---

### Description

Embed is a wrapper for the <embed> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/embed>

### Usage

```
htmlEmbed(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, height=NULL,
src=NULL, type=NULL, width=NULL, accessKey=NULL,
className=NULL, contentEditable=NULL, contextMenu=NULL,
dir=NULL, draggable=NULL, hidden=NULL, lang=NULL,
spellCheck=NULL, style=NULL, tabIndex=NULL, title=NULL,
loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in call-backs. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
height	Character   numeric. Specifies the height of elements listed here. For all other elements, use the CSS height property. Note: In some instances, such as <div>, this is a legacy attribute, in which case the CSS height property should be used instead.
src	Character. The URL of the embeddable content.
type	Character. Defines the type of the element.
width	Character   numeric. For the elements listed here, this establishes the element's width. Note: For all other instances, such as <div>, this is a legacy attribute, in which case the CSS width property should be used instead.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.



loading\_state Lists containing elements 'is\_loading', 'prop\_name', 'component\_name'. those elements have the following types: - is\_loading (logical; optional): determines if the component is loading or not - prop\_name (character; optional): holds which property is loading - component\_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer

... wildcards allowed have the form: 'data-\*', 'aria-\*'

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(htmlDiv(list(
    htmlEmbed(
      src = 'https://archive.org/embed/VintageCartoonsSet1Mp4',
      width = '500',
      height = '500')
    )
  )
)

  app$run_server()
}
```

---

htmlFieldset

*Fieldset component*

---

### Description

Fieldset is a wrapper for the <fieldset> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/fieldset>

### Usage

```
htmlFieldset(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, disabled=NULL,
form=NULL, name=NULL, accessKey=NULL, className=NULL,
contentEditable=NULL, contextMenu=NULL, dir=NULL,
draggable=NULL, hidden=NULL, lang=NULL, spellCheck=NULL,
style=NULL, tabIndex=NULL, title=NULL, loading_state=NULL,
...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
disabled	A value equal to: 'disabled', 'disabled'   logical. Indicates whether the user can interact with the element.
form	Character. Indicates the form that is the owner of the element.
name	Character. Name of the element. For example used by the server to identify the fields in form submits.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'



**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(htmlDiv(list(
    htmlFigure(children = list(
      htmlImg(src = 'https://brand.plotly.com/static/images/plotly-logo-01-stripe@2x.png'),
      htmlFigcaption(children = 'Plotly Logo'))
    )
  )
)

  app$run_server()
}

```

---

htmlFigure	<i>Figure component</i>
------------	-------------------------

---

**Description**

Figure is a wrapper for the <figure> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/figure>

**Usage**

```

htmlFigure(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.

key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(htmlDiv(list(
    htmlFigure(children = list(
      htmlImg(src = 'https://brand.plotly.com/static/images/plotly-logo-01-stripe@2x.png',
        width = '400',
```

```

        height = '150')
      )
    )
  )
)
app$run_server()
}

```

---

htmlFont

*Font component*


---

### Description

Font is a wrapper for the `<font>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/font>

### Usage

```

htmlFont(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.

<code>contentEditable</code>	Character. Indicates whether the element's content is editable.
<code>contextMenu</code>	Character. Defines the ID of a <code>&lt;menu&gt;</code> element which will serve as the element's context menu.
<code>dir</code>	Character. Defines the text direction. Allowed values are <code>ltr</code> (Left-To-Right) or <code>rtl</code> (Right-To-Left)
<code>draggable</code>	Character. Defines whether the element can be dragged.
<code>hidden</code>	A value equal to: <code>'hidden'</code> , <code>'hidden'   logical</code> . Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
<code>lang</code>	Character. Defines the language used in the element.
<code>spellCheck</code>	Character. Indicates whether spell checking is allowed for the element.
<code>style</code>	Named list. Defines CSS styles which will override styles previously set.
<code>tabIndex</code>	Character. Overrides the browser's default tab order and follows the one specified instead.
<code>title</code>	Character. Text to be displayed in a tooltip when hovering over the element.
<code>loading_state</code>	Lists containing elements <code>'is_loading'</code> , <code>'prop_name'</code> , <code>'component_name'</code> . those elements have the following types: - <code>is_loading</code> (logical; optional): determines if the component is loading or not - <code>prop_name</code> (character; optional): holds which property is loading - <code>component_name</code> (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
<code>...</code>	wildcards allowed have the form: <code>'data-*'</code> , <code>'aria-*'</code>

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
# Starting with HTML 4, HTML does not convey styling information
# anymore (outside the <style> element or the style attribute of each
# element). CSS should be used for styling instead.
```

---

htmlFooter

*Footer component*

---

**Description**

Footer is a wrapper for the `<footer>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/footer>



**Usage**

```
htmlFooter(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.

loading\_state Lists containing elements 'is\_loading', 'prop\_name', 'component\_name'. those elements have the following types: - is\_loading (logical; optional): determines if the component is loading or not - prop\_name (character; optional): holds which property is loading - component\_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer

... wildcards allowed have the form: 'data-\*', 'aria-\*'

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(htmlDiv(list(
    htmlFooter(list(
      htmlH1('Dash'),
      htmlLi('Pointer1'),
      htmlLi('Pointer2')
    )
  )
))
)
)

app$run_server()
}
```

---

htmlForm

*Form component*

---

### Description

Form is a wrapper for the <form> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/form>

### Usage

```
htmlForm(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, accept=NULL,
acceptCharset=NULL, action=NULL, autoComplete=NULL,
encType=NULL, method=NULL, name=NULL, noValidate=NULL,
target=NULL, accessKey=NULL, className=NULL,
```

```
contentEditable=NULL, contextMenu=NULL, dir=NULL,
draggable=NULL, hidden=NULL, lang=NULL, spellCheck=NULL,
style=NULL, tabIndex=NULL, title=NULL, loading_state=NULL,
...)
```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accept	Character. List of types the server accepts, typically a file type.
acceptCharset	Character. List of supported charsets.
action	Character. The URI of a program that processes the information submitted via the form.
autoComplete	Character. Indicates whether controls in this form can by default have their values automatically completed by the browser.
encType	Character. Defines the content type of the form data when the method is POST.
method	Character. Defines which HTTP method to use when submitting the form. Can be GET (default) or POST.
name	Character. Name of the element. For example used by the server to identify the fields in form submits.
noValidate	A value equal to: 'novalidate', 'novalidate', 'novalidate'   logical. This attribute indicates that the form shouldn't be validated when submitted.
target	Character.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)



```

    app$run_server()
  }

```

---

htmlFrame	<i>Frame component</i>
-----------	------------------------

---

### Description

Frame is a wrapper for the <frame> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/frame>

### Usage

```

htmlFrame(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)

draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
# htmlFrame is now deprecated. htmlIFrame is recommended instead.
```

---

htmlFrameset

*Frameset component*

---

**Description**

Frameset is a wrapper for the <frameset> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/frameset>

**Usage**

```
htmlFrameset(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

# htmlFrameset is now deprecated. htmlIFrame is recommended instead.

---

htmlH1

*H1 component*

---

**Description**

H1 is a wrapper for the <h1> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/h1>

**Usage**

```
htmlH1(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)



draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

## Value

named list of JSON elements corresponding to React.js properties and their values

## Examples

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlH1(list(
        'Dash Html',
        htmlBr(), #We can customize
        htmlSpan('Dash', style = list('opacity' = '0.8')),
        htmlSpan(' Core'))))
    )
  )
)

app$run_server()
}

```

htmlH2

*H2 component***Description**

H2 is a wrapper for the <h2> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/h2>

**Usage**

```
htmlH2(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.

lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlH2(list(
        'Dash Html',
        htmlBr(), #We can customize
        htmlSpan('Dash', style = list('opacity' = '0.8')),
        htmlSpan(' Core'))
      )
    )
  )

  app$run_server()
}

```

---

htmlH3

*H3 component*


---

**Description**

H3 is a wrapper for the <h3> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/h3>

**Usage**

```
htmlH3(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.

loading\_state Lists containing elements 'is\_loading', 'prop\_name', 'component\_name'. those elements have the following types: - is\_loading (logical; optional): determines if the component is loading or not - prop\_name (character; optional): holds which property is loading - component\_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer

... wildcards allowed have the form: "data-\*", "aria-\*"

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlH3(list(
        'Dash Html',
        htmlBr(), #We can customize
        htmlSpan('Dash', style = list('opacity' = '0.8')),
        htmlSpan(' Core'))
      )
    )
  )

  app$run_server()
}
```

---

htmlH4

*H4 component*

---

### Description

H4 is a wrapper for the <h4> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/h4>

### Usage

```
htmlH4(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlH4(list(
        'Dash Html',
        htmlBr(), #We can customize
        htmlSpan('Dash', style = list('opacity' = '0.8')),
        htmlSpan(' Core'))
      )
    )
  )

  app$run_server()
}

```

---

`htmlH5`*H5 component*

---

**Description**

H5 is a wrapper for the `<h5>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/h5>

**Usage**

```

htmlH5(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

**Arguments**

<code>children</code>	A list of or a singular dash component, string or number. The children of this component
<code>id</code>	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
<code>n_clicks</code>	Numeric. An integer that represents the number of times that this element has been clicked on.

<code>n_clicks_timestamp</code>	Numeric. An integer that represents the time (in ms since 1970) at which <code>n_clicks</code> changed. This can be used to tell which button was changed most recently.
<code>key</code>	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
<code>role</code>	Character. The ARIA role attribute
<code>accessKey</code>	Character. Keyboard shortcut to activate or add focus to the element.
<code>className</code>	Character. Often used with CSS to style elements with common properties.
<code>contentEditable</code>	Character. Indicates whether the element's content is editable.
<code>contextMenu</code>	Character. Defines the ID of a <code>&lt;menu&gt;</code> element which will serve as the element's context menu.
<code>dir</code>	Character. Defines the text direction. Allowed values are <code>ltr</code> (Left-To-Right) or <code>rtl</code> (Right-To-Left)
<code>draggable</code>	Character. Defines whether the element can be dragged.
<code>hidden</code>	A value equal to: <code>'hidden'</code> , <code>'hidden'   logical</code> . Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
<code>lang</code>	Character. Defines the language used in the element.
<code>spellCheck</code>	Character. Indicates whether spell checking is allowed for the element.
<code>style</code>	Named list. Defines CSS styles which will override styles previously set.
<code>tabIndex</code>	Character. Overrides the browser's default tab order and follows the one specified instead.
<code>title</code>	Character. Text to be displayed in a tooltip when hovering over the element.
<code>loading_state</code>	Lists containing elements <code>'is_loading'</code> , <code>'prop_name'</code> , <code>'component_name'</code> . those elements have the following types: - <code>is_loading</code> (logical; optional): determines if the component is loading or not - <code>prop_name</code> (character; optional): holds which property is loading - <code>component_name</code> (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
<code>...</code>	wildcards allowed have the form: <code>'data-*'</code> , <code>'aria-*'</code>

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()
```



```

    app$layout(
      htmlDiv(list(
        htmlH5(list(
          'Dash Html',
          htmlBr(), #We can customize
          htmlSpan('Dash', style = list('opacity' = '0.8')),
          htmlSpan(' Core')))
        )
      )
    )
  app$run_server()
}

```

---

htmlH6

*H6 component*


---

### Description

H6 is a wrapper for the <h6> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/h6>

### Usage

```

htmlH6(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute

<code>accessKey</code>	Character. Keyboard shortcut to activate or add focus to the element.
<code>className</code>	Character. Often used with CSS to style elements with common properties.
<code>contentEditable</code>	Character. Indicates whether the element's content is editable.
<code>contextMenu</code>	Character. Defines the ID of a <code>&lt;menu&gt;</code> element which will serve as the element's context menu.
<code>dir</code>	Character. Defines the text direction. Allowed values are <code>ltr</code> (Left-To-Right) or <code>rtl</code> (Right-To-Left)
<code>draggable</code>	Character. Defines whether the element can be dragged.
<code>hidden</code>	A value equal to: <code>'hidden'</code> , <code>'hidden'   logical</code> . Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
<code>lang</code>	Character. Defines the language used in the element.
<code>spellCheck</code>	Character. Indicates whether spell checking is allowed for the element.
<code>style</code>	Named list. Defines CSS styles which will override styles previously set.
<code>tabIndex</code>	Character. Overrides the browser's default tab order and follows the one specified instead.
<code>title</code>	Character. Text to be displayed in a tooltip when hovering over the element.
<code>loading_state</code>	Lists containing elements <code>'is_loading'</code> , <code>'prop_name'</code> , <code>'component_name'</code> . those elements have the following types: <code>- is_loading</code> (logical; optional): determines if the component is loading or not - <code>prop_name</code> (character; optional): holds which property is loading - <code>component_name</code> (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
<code>...</code>	wildcards allowed have the form: <code>'data-*'</code> , <code>'aria-*'</code>

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlH6(list(
        'Dash Html',
        htmlBr(), #We can customize
        htmlSpan('Dash', style = list('opacity' = '0.8')),
        htmlSpan(' Core')))
    )
  )
}
```

```

    )
    app$run_server()
}

```

---

htmlHeader

*Header component*


---

### Description

Header is a wrapper for the <header> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/header>

### Usage

```

htmlHeader(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.

<code>dir</code>	Character. Defines the text direction. Allowed values are <code>ltr</code> (Left-To-Right) or <code>rtl</code> (Right-To-Left)
<code>draggable</code>	Character. Defines whether the element can be dragged.
<code>hidden</code>	A value equal to: <code>'hidden'</code> , <code>'hidden'   logical</code> . Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
<code>lang</code>	Character. Defines the language used in the element.
<code>spellCheck</code>	Character. Indicates whether spell checking is allowed for the element.
<code>style</code>	Named list. Defines CSS styles which will override styles previously set.
<code>tabIndex</code>	Character. Overrides the browser's default tab order and follows the one specified instead.
<code>title</code>	Character. Text to be displayed in a tooltip when hovering over the element.
<code>loading_state</code>	Lists containing elements <code>'is_loading'</code> , <code>'prop_name'</code> , <code>'component_name'</code> . those elements have the following types: - <code>is_loading</code> (logical; optional): determines if the component is loading or not - <code>prop_name</code> (character; optional): holds which property is loading - <code>component_name</code> (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
<code>...</code>	wildcards allowed have the form: <code>''data-*'</code> , <code>'aria-*'</code>

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlHeader("This is a header"),
      htmlP("And here is some text.")
    ))
  )
)
}

app$run_server()
```

htmlHgroup

*Hgroup component***Description**

Hgroup is a wrapper for the <hgroup> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/hgroup>

**Usage**

```
htmlHgroup(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.

lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: ``data-*`, `aria-*``

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlHgroup(list(
        htmlH1('MultiLevel Title'),
        htmlHr(),
        htmlH2('Header')
      )
    )
  )
)
)
app$run_server()
}

```

---

htmlHr

*Hr component*


---

**Description**

Hr is a wrapper for the <hr> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/hr>

**Usage**

```
htmlHr(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.

loading\_state Lists containing elements 'is\_loading', 'prop\_name', 'component\_name'. those elements have the following types: - is\_loading (logical; optional): determines if the component is loading or not - prop\_name (character; optional): holds which property is loading - component\_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer

... wildcards allowed have the form: 'data-\*', 'aria-\*'

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlDiv(list(
        htmlH1('Dash'),
        htmlHr(),
        htmlH2('Components')
      )
    )
  )
)

  app$run_server()
}
```

---

htmlI

*I component*

---

### Description

I is a wrapper for the `<i>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/i>

### Usage

```
htmlI(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
```



lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,  
title=NULL, loading\_state=NULL, ...)

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: ``data-*``, ``aria-*``

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlI('Italicized Text')
    )
  )
)

app$run_server()
}

```

---

htmlIframe

*Iframe component*


---

**Description**

Iframe is a wrapper for the <iframe> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/iframe>

**Usage**

```

htmlIframe(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, height=NULL,
name=NULL, sandbox=NULL, src=NULL, srcDoc=NULL, width=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.

n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
height	Character   numeric. Specifies the height of elements listed here. For all other elements, use the CSS height property. Note: In some instances, such as <div>, this is a legacy attribute, in which case the CSS height property should be used instead.
name	Character. Name of the element. For example used by the server to identify the fields in form submits.
sandbox	Character. Stops a document loaded in an iframe from using certain features (such as submitting forms or opening new windows).
src	Character. The URL of the embeddable content.
srcDoc	Character.
width	Character   numeric. For the elements listed here, this establishes the element's width. Note: For all other instances, such as <div>, this is a legacy attribute, in which case the CSS width property should be used instead.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(htmlDiv(list(
    htmlIframe(width = "600px", height = "600px",
               src = "https://dashr.plotly.com/")
  )
))

  app$run_server()
}
```

---

htmlImg

*Img component*


---

**Description**

Img is a wrapper for the <img> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/img>

**Usage**

```
htmlImg(children=NULL, id=NULL, n_clicks=NULL,
         n_clicks_timestamp=NULL, key=NULL, role=NULL, alt=NULL,
         crossOrigin=NULL, height=NULL, sizes=NULL, src=NULL,
         srcSet=NULL, useMap=NULL, width=NULL, accessKey=NULL,
         className=NULL, contentEditable=NULL, contextMenu=NULL,
         dir=NULL, draggable=NULL, hidden=NULL, lang=NULL,
         spellCheck=NULL, style=NULL, tabIndex=NULL, title=NULL,
         loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.

n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
alt	Character. Alternative text in case an image can't be displayed.
crossOrigin	Character. How the element handles cross-origin requests
height	Character   numeric. Specifies the height of elements listed here. For all other elements, use the CSS height property. Note: In some instances, such as <div>, this is a legacy attribute, in which case the CSS height property should be used instead.
sizes	Character.
src	Character. The URL of the embeddable content.
srcSet	Character. One or more responsive image candidates.
useMap	Character.
width	Character   numeric. For the elements listed here, this establishes the element's width. Note: For all other instances, such as <div>, this is a legacy attribute, in which case the CSS width property should be used instead.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(htmlDiv(list(
    htmlImg(src = 'https://brand.plotly.com/static/images/plotly-logo-01-stripe@2x.png',
            height='200', width='400')
  )
  )
  )

  app$run_server()
}

```

---

htmlIns

*Ins component*


---

**Description**

Ins is a wrapper for the <ins> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ins>

**Usage**

```

htmlIns(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, cite=NULL,
dateTime=NULL, accessKey=NULL, className=NULL,
contentEditable=NULL, contextMenu=NULL, dir=NULL,
draggable=NULL, hidden=NULL, lang=NULL, spellCheck=NULL,
style=NULL, tabIndex=NULL, title=NULL, loading_state=NULL,
...)

```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.

n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
cite	Character. Contains a URI which points to the source of the quote or change.
dateTime	Character. Indicates the date and time associated with the element.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)
```

```

app <- Dash$new()

app$layout(
  htmlDiv(list(
    htmlIns('This text has been inserted')
  )
)
)

app$run_server()
}

```

---

htmlIsindex

*Isindex component*


---

### Description

Isindex is a wrapper for the `<isindex>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/isindex>

### Usage

```

htmlIsindex(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.



className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
# This component is deprecated and its use is no longer recommended.
```

---

htmlKbd

*Kbd component*


---

**Description**

Kbd is a wrapper for the <kbd> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/kbd>

**Usage**

```
htmlKbd(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.

loading\_state Lists containing elements 'is\_loading', 'prop\_name', 'component\_name'. those elements have the following types: - is\_loading (logical; optional): determines if the component is loading or not - prop\_name (character; optional): holds which property is loading - component\_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer

... wildcards allowed have the form: 'data-\*', 'aria-\*'

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlP('Please Press: '),
      htmlKbd(list(
        'Ctl + ',
        'Alt + ',
        'Delete'))
    ))
  )
)

app$run_server()
}
```

---

htmlKeygen

*Keygen component*

---

### Description

Keygen is a wrapper for the <keygen> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/keygen>

### Usage

```
htmlKeygen(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
autoFocus=NULL, challenge=NULL, disabled=NULL, form=NULL,
keyType=NULL, name=NULL, accessKey=NULL, className=NULL,
contentEditable=NULL, contextMenu=NULL, dir=NULL,
```

draggable=NULL, hidden=NULL, lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL, title=NULL, loading\_state=NULL, ...)

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
autoFocus	A value equal to: 'autofocus', 'autofocus', 'autofocus'   logical. The element should be automatically focused after the page loaded.
challenge	Character. A challenge string that is submitted along with the public key.
disabled	A value equal to: 'disabled', 'disabled'   logical. Indicates whether the user can interact with the element.
form	Character. Indicates the form that is the owner of the element.
keyType	Character. Specifies the type of key generated.
name	Character. Name of the element. For example used by the server to identify the fields in form submits.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.

tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
# This feature is obsolete. It may still work in some
# browsers, but could stop working at any time. Try to
# avoid using this component.
```

---

htmlLabel

*Label component*


---

**Description**

Label is a wrapper for the <label> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/label>

**Usage**

```
htmlLabel(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, htmlFor=NULL,
form=NULL, accessKey=NULL, className=NULL,
contentEditable=NULL, contextMenu=NULL, dir=NULL,
draggable=NULL, hidden=NULL, lang=NULL, spellCheck=NULL,
style=NULL, tabIndex=NULL, title=NULL, loading_state=NULL,
...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.

n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
htmlFor	Character. Describes elements which belongs to this one.
form	Character. Indicates the form that is the owner of the element.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)
```

```

library(dashCoreComponents)

app <- Dash$new()

app$layout(
  htmlDiv(
    htmlLabel(
      list(htmlDiv(list("Time points")),
        dccInput(
          id = "times-input",
          placeholder = "Enter a value...",
          type = "number",
          value = 1,
          min = 3,
          max = 999)
        )
      )
    )
  )
  app$run_server()
}

```

---

htmlLegend

*Legend component*


---

## Description

Legend is a wrapper for the `<legend>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/legend>

## Usage

```

htmlLegend(children=NULL, id=NULL, n_clicks=NULL,
  n_clicks_timestamp=NULL, key=NULL, role=NULL,
  accessKey=NULL, className=NULL, contentEditable=NULL,
  contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
  lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
  title=NULL, loading_state=NULL, ...)

```

## Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.

n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)
  library(dashCoreComponents)

  app <- Dash$new()
```





n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
value	Character. Defines a default value which will be displayed in the element on page load.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)
```

```
app <- Dash$new()

app$layout(
  htmlDiv(list(
    htmlOl(list(
      htmlLi("Montreal"),
      htmlLi("Toronto"),
      htmlLi("Halifax")
    )),
    htmlUl(list(
      htmlLi("Montreal"),
      htmlLi("Toronto"),
      htmlLi("Halifax")
    ))
  ))
)

app$run_server()
}
```

---

htmlLink

*Link component*

---

### Description

Link is a wrapper for the <link> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/link>

### Usage

```
htmlLink(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
crossOrigin=NULL, href=NULL, hrefLang=NULL, integrity=NULL,
media=NULL, rel=NULL, sizes=NULL, accessKey=NULL,
className=NULL, contentEditable=NULL, contextMenu=NULL,
dir=NULL, draggable=NULL, hidden=NULL, lang=NULL,
spellCheck=NULL, style=NULL, tabIndex=NULL, title=NULL,
loading_state=NULL, ...)
```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.

n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
crossOrigin	Character. How the element handles cross-origin requests
href	Character. The URL of a linked resource.
hrefLang	Character. Specifies the language of the linked resource.
integrity	Character. Specifies a Subresource Integrity value that allows browsers to verify what they fetch.
media	Character. Specifies a hint of the media for which the linked resource was designed.
rel	Character. Specifies the relationship of the target object to the link object.
sizes	Character.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlLink(rel = "stylesheet",
               type = "text/css",
               href = "https://codepen.io/chriddyp/pen/bWLwgP.css")
    ))
  )

  app$run_server()
}

```

---

htmlListing

*Listing component*


---

**Description**

Listing is a wrapper for the <listing> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/listing>

**Usage**

```

htmlListing(children=NULL, id=NULL, n_clicks=NULL,
            n_clicks_timestamp=NULL, key=NULL, role=NULL,
            accessKey=NULL, className=NULL, contentEditable=NULL,
            contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
            lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
            title=NULL, loading_state=NULL, ...)

```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.

n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  # Warning: The <listing> element was intended as a way to render HTML code on a page.
  # It was never properly supported, and is now deprecated. Using <listing> will almost
  # certainly result in unexpected results. Instead, use <code>, or place the content in
  # a <div> with the appropriate CSS styling.
```

```

library(dash)
library(dashHtmlComponents)

app <- Dash$new()

app$layout(
  htmlDiv(list(
    htmlListing(list(
      htmlUI("A"),
      htmlUI("B"),
      htmlUI("C")
    ))
  ))
)

app$run_server()
}

```

---

htmlMain

*Main component*


---

## Description

Main is a wrapper for the <main> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/main>

## Usage

```

htmlMain(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

## Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.

key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlMain(
        list(htmlH1("Benjamin Franklin")),
```



```

    htmlP("Benjamin Franklin was an American polymath
          and one of the Founding Fathers of the United States.
          Franklin was a leading author, printer, political theorist,
          politician, Freemason, postmaster, scientist, inventor,
          humorist, civic activist, statesman, and diplomat.")
  )
))
)

app$run_server()
}

```

---

htmlMapEl

*MapEl component*


---

### Description

MapEl is a wrapper for the `<map>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/map>

### Usage

```

htmlMapEl(children=NULL, id=NULL, n_clicks=NULL,
           n_clicks_timestamp=NULL, key=NULL, role=NULL, name=NULL,
           accessKey=NULL, className=NULL, contentEditable=NULL,
           contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
           lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
           title=NULL, loading_state=NULL, ...)

```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
name	Character. Name of the element. For example used by the server to identify the fields in form submits.

accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
# The URL below has been chunked to comply with CRAN
# requirements; the use of file.path is optional and not required
# for this component.
if (interactive() && require(dash)) {
  app$layout(
    htmlDiv(list(
      htmlImg(src = file.path('https://upload.wikimedia.org',
        'wikipedia/commons/0/0c',
        'PIA17351-ApparentSizes-MarsDeimosPhobos-EarthMoon.jpg',
        fsep = '/'),
        useMap = '#image-map'),
      htmlMapEl(list(
        htmlArea(target='_blank',
          alt='Deimos',
          title='Deimos',
```

```

        href='https://en.wikipedia.org/wiki/Deimos_(moon)',
        coords='5,114,32,147',
        shape='rect'),
    htmlArea(target='_blank',
        alt='Phobos',
        title='Phobos',
        href='https://en.wikipedia.org/wiki/Phobos_(moon)',
        coords='113,196,32,103',
        shape='rect'),
    htmlArea(target='_blank',
        alt='Moon',
        title='Moon',
        href='https://en.wikipedia.org/wiki/Moon',
        coords='127,285,294,1',
        shape='rect')
    ),
    name = 'image-map'
),
htmlDiv(children = 'Click on the image to visit a Wikipedia article',
    id = 'object-name')
)
)
)
app$run_server()
}

```

---

htmlMark

*Mark component*


---

## Description

Mark is a wrapper for the `<mark>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/mark>

## Usage

```

htmlMark(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

## Arguments

**children** A list of or a singular dash component, string or number. The children of this component

<code>id</code>	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
<code>n_clicks</code>	Numeric. An integer that represents the number of times that this element has been clicked on.
<code>n_clicks_timestamp</code>	Numeric. An integer that represents the time (in ms since 1970) at which <code>n_clicks</code> changed. This can be used to tell which button was changed most recently.
<code>key</code>	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
<code>role</code>	Character. The ARIA role attribute
<code>accessKey</code>	Character. Keyboard shortcut to activate or add focus to the element.
<code>className</code>	Character. Often used with CSS to style elements with common properties.
<code>contentEditable</code>	Character. Indicates whether the element's content is editable.
<code>contextMenu</code>	Character. Defines the ID of a <code>&lt;menu&gt;</code> element which will serve as the element's context menu.
<code>dir</code>	Character. Defines the text direction. Allowed values are <code>ltr</code> (Left-To-Right) or <code>rtl</code> (Right-To-Left)
<code>draggable</code>	Character. Defines whether the element can be dragged.
<code>hidden</code>	A value equal to: <code>'hidden'</code> , <code>'hidden'   logical</code> . Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
<code>lang</code>	Character. Defines the language used in the element.
<code>spellCheck</code>	Character. Indicates whether spell checking is allowed for the element.
<code>style</code>	Named list. Defines CSS styles which will override styles previously set.
<code>tabIndex</code>	Character. Overrides the browser's default tab order and follows the one specified instead.
<code>title</code>	Character. Text to be displayed in a tooltip when hovering over the element.
<code>loading_state</code>	Lists containing elements <code>'is_loading'</code> , <code>'prop_name'</code> , <code>'component_name'</code> . those elements have the following types: - <code>is_loading</code> (logical; optional): determines if the component is loading or not - <code>prop_name</code> (character; optional): holds which property is loading - <code>component_name</code> (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
<code>...</code>	wildcards allowed have the form: <code>'data-*'</code> , <code>'aria-*'</code>

### Value

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlP(list(
        htmlMark("Plotly"),
        " develops online data analytics and visualization tools."
      ))
    ))
  )

  app$run_server()
}

```

---

htmlMarquee

*Marquee component*


---

**Description**

Marquee is a wrapper for the <marquee> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/marquee>

**Usage**

```

htmlMarquee(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, loop=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.

key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
loop	A value equal to: 'loop', 'loop'   logical. Indicates whether the media should start playing from the start when it's finished.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  # This feature is obsolete. It may still work in some
  # browsers, but could stop working at any time. Try to
  # avoid using this component.

  library(dash)
  library(dashHtmlComponents)
```

```

app <- Dash$new()

app$layout(
  htmlDiv(list(
    htmlMarquee("Here is some sliding text that uses htmlMarquee")
  ))
)

app$run_server()
}

```

---

htmlMeta

*Meta component*


---

### Description

Meta is a wrapper for the <meta> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/meta>

### Usage

```

htmlMeta(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, charSet=NULL,
content=NULL, httpEquiv=NULL, name=NULL, accessKey=NULL,
className=NULL, contentEditable=NULL, contextMenu=NULL,
dir=NULL, draggable=NULL, hidden=NULL, lang=NULL,
spellCheck=NULL, style=NULL, tabIndex=NULL, title=NULL,
loading_state=NULL, ...)

```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
charSet	Character. Declares the character encoding of the page or script.

content	Character. A value associated with http-equiv or name depending on the context.
httpEquiv	Character. Defines a pragma directive.
name	Character. Name of the element. For example used by the server to identify the fields in form submits.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlP("The <meta> tag provides metadata about the HTML document.
        Metadata will not be displayed on the page, but will be machine parsable.
```



```

        To view meta tag the content of this page can be inspected."),
    htmlMeta(name = "author", content = "Edward Tufte")
  ))
)

app$run_server()
}

```

---

htmlMeter

*Meter component*


---

### Description

Meter is a wrapper for the `<meter>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/meter>

### Usage

```

htmlMeter(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, form=NULL,
high=NULL, low=NULL, max=NULL, min=NULL, optimum=NULL,
value=NULL, accessKey=NULL, className=NULL,
contentEditable=NULL, contextMenu=NULL, dir=NULL,
draggable=NULL, hidden=NULL, lang=NULL, spellCheck=NULL,
style=NULL, tabIndex=NULL, title=NULL, loading_state=NULL,
...)

```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
form	Character. Indicates the form that is the owner of the element.
high	Character. Indicates the lower bound of the upper range.

low	Character. Indicates the upper bound of the lower range.
max	Character   numeric. Indicates the maximum value allowed.
min	Character   numeric. Indicates the minimum value allowed.
optimum	Character. Indicates the optimal numeric value.
value	Character. Defines a default value which will be displayed in the element on page load.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
```

```

    htmlDiv(list(
      htmlLabel("Sample Level Meter: "),
      htmlMeter(id = "sample-meter",
        min = 0,
        max = 100,
        low = 33,
        high = 66,
        optimum = 80,
        value = 80
      )
    ))
  )

  app$run_server()
}

```

---

htmlMulticol

*Multicol component*


---

## Description

Multicol is a wrapper for the `<multicol>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/multicol>

## Usage

```

htmlMulticol(children=NULL, id=NULL, n_clicks=NULL,
  n_clicks_timestamp=NULL, key=NULL, role=NULL,
  accessKey=NULL, className=NULL, contentEditable=NULL,
  contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
  lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
  title=NULL, loading_state=NULL, ...)

```

## Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info

role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
# Warning: The <multicol> tag is obsolete, it might not work as intended.
# Try to avoid using it.
```

---

htmlNav

*Nav component*


---

**Description**

Nav is a wrapper for the <nav> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/nav>

**Usage**

```
htmlNav(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.

loading\_state Lists containing elements 'is\_loading', 'prop\_name', 'component\_name'. those elements have the following types: - is\_loading (logical; optional): determines if the component is loading or not - prop\_name (character; optional): holds which property is loading - component\_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer

... wildcards allowed have the form: "data-\*", "aria-\*"

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlNav(
        list(
          htmlA("Plotly", href = "https://plotly.com/"),
          "> ",
          htmlA("Dash", href = "https://plotly.com/dash"),
          "> ",
          htmlA("Request Trial", href = "https://go.plotly.com/dash-enterprise-trial")
        )
      )
    ))
  )

  app$run_server()
}
```

---

htmlNextid

*Nextid component*

---

### Description

Nextid is a wrapper for the <nextid> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/nextid>

**Usage**

```
htmlNextid(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.

loading\_state Lists containing elements 'is\_loading', 'prop\_name', 'component\_name'. those elements have the following types: - is\_loading (logical; optional): determines if the component is loading or not - prop\_name (character; optional): holds which property is loading - component\_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer

... wildcards allowed have the form: 'data-\*', 'aria-\*'

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
# This component is deprecated and its use is no longer recommended.
# The <nextid> tag has been obsolete since HTML Version 3.2.
```

---

htmlNobr	<i>Nobr component</i>
----------	-----------------------

---

### Description

Nobr is a wrapper for the <nobr> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/nobr>

### Usage

```
htmlNobr(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.



key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlNobr("Lorem ipsum dolor sit amet,
                consectetur adipiscing elit, sed do eiusmod
```

```

        tempor incididunt ut labore et dolore magna aliqua.
        Ut enim ad minim veniam, quis nostrud exercitation
        ullamco laboris nisi ut aliquip ex ea commodo consequat.
        Duis aute irure dolor in reprehenderit in voluptate
        velit esse cillum dolore eu fugiat nulla pariatur.
        Excepteur sint occaecat cupidatat non proident,
        sunt in culpa qui officia deserunt mollit anim id est laborum."
    )
  ))
)

app$run_server()
}

```

---

`htmlNoscript`*Noscript component*

---

### Description

Noscript is a wrapper for the `<noscript>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/noscript>

### Usage

```

htmlNoscript(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

### Arguments

<code>children</code>	A list of or a singular dash component, string or number. The children of this component
<code>id</code>	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
<code>n_clicks</code>	Numeric. An integer that represents the number of times that this element has been clicked on.
<code>n_clicks_timestamp</code>	Numeric. An integer that represents the time (in ms since 1970) at which <code>n_clicks</code> changed. This can be used to tell which button was changed most recently.
<code>key</code>	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
<code>role</code>	Character. The ARIA role attribute

accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
# This component may be used in the index template to define
# alternate content in browsers which have disabled scripts,
# or in which scripts are not supported.
```

---

htmlObjectEl

*ObjectEl component*


---

**Description**

ObjectEl is a wrapper for the <object> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/object>

**Usage**

```
htmlObjectEl(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, form=NULL,
height=NULL, name=NULL, type=NULL, useMap=NULL, width=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
form	Character. Indicates the form that is the owner of the element.
height	Character   numeric. Specifies the height of elements listed here. For all other elements, use the CSS height property. Note: In some instances, such as <div>, this is a legacy attribute, in which case the CSS height property should be used instead.
name	Character. Name of the element. For example used by the server to identify the fields in form submits.
type	Character. Defines the type of the element.
useMap	Character.
width	Character   numeric. For the elements listed here, this establishes the element's width. Note: For all other instances, such as <div>, this is a legacy attribute, in which case the CSS width property should be used instead.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.

<code>dir</code>	Character. Defines the text direction. Allowed values are <code>ltr</code> (Left-To-Right) or <code>rtl</code> (Right-To-Left)
<code>draggable</code>	Character. Defines whether the element can be dragged.
<code>hidden</code>	A value equal to: <code>'hidden'</code> , <code>'hidden'   logical</code> . Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
<code>lang</code>	Character. Defines the language used in the element.
<code>spellCheck</code>	Character. Indicates whether spell checking is allowed for the element.
<code>style</code>	Named list. Defines CSS styles which will override styles previously set.
<code>tabIndex</code>	Character. Overrides the browser's default tab order and follows the one specified instead.
<code>title</code>	Character. Text to be displayed in a tooltip when hovering over the element.
<code>loading_state</code>	Lists containing elements <code>'is_loading'</code> , <code>'prop_name'</code> , <code>'component_name'</code> . those elements have the following types: - <code>is_loading</code> (logical; optional): determines if the component is loading or not - <code>prop_name</code> (character; optional): holds which property is loading - <code>component_name</code> (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
<code>...</code>	wildcards allowed have the form: <code>'data-*'</code> , <code>'aria-*'</code>

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  # Note that htmlObjectEl generates the <object> HTML element;
  # for more information, please visit the link in this
  # component's description.
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlObjectEl(
        width = 100,
        height = 97
        #data = "https://i.postimg.cc/tJd8PSVf/Plotly-logo-01-square.png"
      )
    ))
  )

  app$run_server()
}
```

htmlOl

*Ol component***Description**

Ol is a wrapper for the `<ol>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ol>

**Usage**

```
htmlOl(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, reversed=NULL,
start=NULL, accessKey=NULL, className=NULL,
contentEditable=NULL, contextMenu=NULL, dir=NULL,
draggable=NULL, hidden=NULL, lang=NULL, spellCheck=NULL,
style=NULL, tabIndex=NULL, title=NULL, loading_state=NULL,
...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
reversed	A value equal to: 'reversed', 'reversed'   logical. Indicates whether the list should be displayed in a descending order instead of a ascending.
start	Character. Defines the first number if other than 1.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <code>&lt;menu&gt;</code> element which will serve as the element's context menu.

<code>dir</code>	Character. Defines the text direction. Allowed values are <code>ltr</code> (Left-To-Right) or <code>rtl</code> (Right-To-Left)
<code>draggable</code>	Character. Defines whether the element can be dragged.
<code>hidden</code>	A value equal to: <code>'hidden'</code> , <code>'hidden'   logical</code> . Prevents rendering of given element, while keeping child elements, e.g. <code>script</code> elements, active.
<code>lang</code>	Character. Defines the language used in the element.
<code>spellCheck</code>	Character. Indicates whether spell checking is allowed for the element.
<code>style</code>	Named list. Defines CSS styles which will override styles previously set.
<code>tabIndex</code>	Character. Overrides the browser's default tab order and follows the one specified instead.
<code>title</code>	Character. Text to be displayed in a tooltip when hovering over the element.
<code>loading_state</code>	Lists containing elements <code>'is_loading'</code> , <code>'prop_name'</code> , <code>'component_name'</code> . those elements have the following types: - <code>is_loading</code> (logical; optional): determines if the component is loading or not - <code>prop_name</code> (character; optional): holds which property is loading - <code>component_name</code> (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from <code>dash-renderer</code>
<code>...</code>	wildcards allowed have the form: <code>'data-*'</code> , <code>'aria-*'</code>

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlOl(list(
        htmlLi("Un"),
        htmlLi("Deux"),
        htmlLi("Trois")
      ))
    ))
  )

  app$run_server()
}
```

---

 htmlOptgroup

*Optgroup component*


---

### Description

Optgroup is a wrapper for the <optgroup> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/optgroup>

### Usage

```
htmlOptgroup(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, disabled=NULL,
label=NULL, accessKey=NULL, className=NULL,
contentEditable=NULL, contextMenu=NULL, dir=NULL,
draggable=NULL, hidden=NULL, lang=NULL, spellCheck=NULL,
style=NULL, tabIndex=NULL, title=NULL, loading_state=NULL,
...)
```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
disabled	A value equal to: 'disabled', 'disabled'   logical. Indicates whether the user can interact with the element.
label	Character. Specifies a user-readable title of the element.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.



<code>dir</code>	Character. Defines the text direction. Allowed values are <code>ltr</code> (Left-To-Right) or <code>rtl</code> (Right-To-Left)
<code>draggable</code>	Character. Defines whether the element can be dragged.
<code>hidden</code>	A value equal to: <code>'hidden'</code> , <code>'hidden'   logical</code> . Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
<code>lang</code>	Character. Defines the language used in the element.
<code>spellCheck</code>	Character. Indicates whether spell checking is allowed for the element.
<code>style</code>	Named list. Defines CSS styles which will override styles previously set.
<code>tabIndex</code>	Character. Overrides the browser's default tab order and follows the one specified instead.
<code>title</code>	Character. Text to be displayed in a tooltip when hovering over the element.
<code>loading_state</code>	Lists containing elements <code>'is_loading'</code> , <code>'prop_name'</code> , <code>'component_name'</code> . those elements have the following types: - <code>is_loading</code> (logical; optional): determines if the component is loading or not - <code>prop_name</code> (character; optional): holds which property is loading - <code>component_name</code> (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
<code>...</code>	wildcards allowed have the form: <code>'data-*'</code> , <code>'aria-*'</code>

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlLabel(
        htmlFor = "option-select", "Please select car brand/model: "
      ),
      htmlSelect(id = "option-select", list(
        htmlOptgroup("Audi"), #label = "Audi"
        htmlOption("TT"),
        htmlOption("A4"),
        htmlOptgroup("BMW"), #label = "BMW"
        htmlOption("3 Series"),
        htmlOption("5 Series")
      ))
    ))
  )

  app$run_server()
}

```

htmlOption

*Option component***Description**

Option is a wrapper for the <option> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/option>

**Usage**

```
htmlOption(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, disabled=NULL,
label=NULL, selected=NULL, value=NULL, accessKey=NULL,
className=NULL, contentEditable=NULL, contextMenu=NULL,
dir=NULL, draggable=NULL, hidden=NULL, lang=NULL,
spellCheck=NULL, style=NULL, tabIndex=NULL, title=NULL,
loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
disabled	A value equal to: 'disabled', 'disabled'   logical. Indicates whether the user can interact with the element.
label	Character. Specifies a user-readable title of the element.
selected	A value equal to: 'selected', 'selected'   logical. Defines a value which will be selected on page load.
value	Character. Defines a default value which will be displayed in the element on page load.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.

contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlSelect(list(
        htmlOption("d'Artagnan"),
        htmlOption("Athos"),
        htmlOption("Porthos"),
        htmlOption("Aramis")
      ))
    ))
  )

  app$run_server()
}

```

---

htmlOutput	<i>Output component</i>
------------	-------------------------

---

### Description

Output is a wrapper for the `<output>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/output>

### Usage

```
htmlOutput(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, htmlFor=NULL,
form=NULL, name=NULL, accessKey=NULL, className=NULL,
contentEditable=NULL, contextMenu=NULL, dir=NULL,
draggable=NULL, hidden=NULL, lang=NULL, spellCheck=NULL,
style=NULL, tabIndex=NULL, title=NULL, loading_state=NULL,
...)
```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
htmlFor	Character. Describes elements which belongs to this one.
form	Character. Indicates the form that is the owner of the element.
name	Character. Name of the element. For example used by the server to identify the fields in form submits.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <code>&lt;menu&gt;</code> element which will serve as the element's context menu.

<code>dir</code>	Character. Defines the text direction. Allowed values are <code>ltr</code> (Left-To-Right) or <code>rtl</code> (Right-To-Left)
<code>draggable</code>	Character. Defines whether the element can be dragged.
<code>hidden</code>	A value equal to: <code>'hidden'</code> , <code>'hidden'   logical</code> . Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
<code>lang</code>	Character. Defines the language used in the element.
<code>spellCheck</code>	Character. Indicates whether spell checking is allowed for the element.
<code>style</code>	Named list. Defines CSS styles which will override styles previously set.
<code>tabIndex</code>	Character. Overrides the browser's default tab order and follows the one specified instead.
<code>title</code>	Character. Text to be displayed in a tooltip when hovering over the element.
<code>loading_state</code>	Lists containing elements <code>'is_loading'</code> , <code>'prop_name'</code> , <code>'component_name'</code> . those elements have the following types: - <code>is_loading</code> (logical; optional): determines if the component is loading or not - <code>prop_name</code> (character; optional): holds which property is loading - <code>component_name</code> (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
<code>...</code>	wildcards allowed have the form: <code>'data-*'</code> , <code>'aria-*'</code>

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
# This component works with htmlForm and htmlInput
# and may be used to present the result of an
# executed script.
```

---

htmlP

*P component*

---

**Description**

P is a wrapper for the `<p>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/p>

**Usage**

```
htmlP(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlP("The <p> tag defines a paragraph.")
    ))
  )

  app$run_server()
}

```

---

htmlParam

*Param component*


---

**Description**

Param is a wrapper for the `<param>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/param>

**Usage**

```

htmlParam(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, name=NULL,
value=NULL, accessKey=NULL, className=NULL,
contentEditable=NULL, contextMenu=NULL, dir=NULL,
draggable=NULL, hidden=NULL, lang=NULL, spellCheck=NULL,
style=NULL, tabIndex=NULL, title=NULL, loading_state=NULL,
...)

```

**Arguments**

<code>children</code>	A list of or a singular dash component, string or number. The children of this component
<code>id</code>	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
<code>n_clicks</code>	Numeric. An integer that represents the number of times that this element has been clicked on.
<code>n_clicks_timestamp</code>	Numeric. An integer that represents the time (in ms since 1970) at which <code>n_clicks</code> changed. This can be used to tell which button was changed most recently.

key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
name	Character. Name of the element. For example used by the server to identify the fields in form submits.
value	Character. Defines a default value which will be displayed in the element on page load.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: ``data-*`, `aria-*``

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()
```



```

app$layout(
  htmlDiv(list(
    htmlP("The <param> element is used to specify the parameters that apply to
    plugin-powered content embedded with an <object> element.
    Read more: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/param"),
    htmlObjectEl(
      #data = "link-to-data-file"
      htmlParam(name = "controller", value = TRUE)
    )
  ))
)

app$run_server()
}

```

---

htmlPicture

*Picture component*


---

## Description

Picture is a wrapper for the <picture> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/picture>

## Usage

```

htmlPicture(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

## Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info

role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
# The URL below has been chunked to comply with CRAN
# requirements; the use of file.path is optional and not required
# for this component.
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlPicture(list(
        htmlSource(srcSet = file.path("https://upload.wikimedia.org",
          "wikipedia/commons/a/a7",
```

```

        "Winter_and_the_City.jpg",
        fsep = "/" ),
        media = "(min-width: 800px)",
        htmlImg(src = file.path("https://upload.wikimedia.org",
            "wikipedia/commons/5/56",
            "Summer_and_the_City.jpg",
            fsep = "/")),
        htmlP("Resize screen to see image changing...")
    ))
  ))
)

app$run_server()
}

```

---

htmlPlaintext

*Plaintext component*


---

### Description

Plaintext is a wrapper for the `<plaintext>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/plaintext>

### Usage

```

htmlPlaintext(children=NULL, id=NULL, n_clicks=NULL,
  n_clicks_timestamp=NULL, key=NULL, role=NULL,
  accessKey=NULL, className=NULL, contentEditable=NULL,
  contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
  lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
  title=NULL, loading_state=NULL, ...)

```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info

role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

## Value

named list of JSON elements corresponding to React.js properties and their values

## Examples

```
if (interactive() && require(dash)) {
  # Warning: The <plaintext> tag is obsolete,
  # it might not work as intended.
  # Use the <pre> tag instead.

  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlPlaintext(),
      htmlBr(),
```

```

    htmlH4("The HTML Plaintext Element (<plaintext>) renders everything following
    the start tag as raw text, ignoring any following HTML. There is no closing tag,
    since everything after it is considered raw text.")
  ))
)

app$run_server()
}

```

---

htmlPre

*Pre component*


---

### Description

Pre is a wrapper for the `<pre>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/pre>

### Usage

```

htmlPre(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.

contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: "data-*", "aria-*"

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlPre(
        "
        Text in a <pre> element is displayed
        in a fixed-width font (usually Courier),
        and it preserves both spaces and line breaks.
        "
      )
    ))
  )

  app$run_server()
}

```

---

htmlProgress	<i>Progress component</i>
--------------	---------------------------

---

### Description

Progress is a wrapper for the `<progress>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/progress>

### Usage

```
htmlProgress(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, form=NULL,
max=NULL, value=NULL, accessKey=NULL, className=NULL,
contentEditable=NULL, contextMenu=NULL, dir=NULL,
draggable=NULL, hidden=NULL, lang=NULL, spellCheck=NULL,
style=NULL, tabIndex=NULL, title=NULL, loading_state=NULL,
...)
```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
form	Character. Indicates the form that is the owner of the element.
max	Character   numeric. Indicates the maximum value allowed.
value	Character. Defines a default value which will be displayed in the element on page load.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <code>&lt;menu&gt;</code> element which will serve as the element's context menu.

<code>dir</code>	Character. Defines the text direction. Allowed values are <code>ltr</code> (Left-To-Right) or <code>rtl</code> (Right-To-Left)
<code>draggable</code>	Character. Defines whether the element can be dragged.
<code>hidden</code>	A value equal to: <code>'hidden'</code> , <code>'hidden'   logical</code> . Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
<code>lang</code>	Character. Defines the language used in the element.
<code>spellCheck</code>	Character. Indicates whether spell checking is allowed for the element.
<code>style</code>	Named list. Defines CSS styles which will override styles previously set.
<code>tabIndex</code>	Character. Overrides the browser's default tab order and follows the one specified instead.
<code>title</code>	Character. Text to be displayed in a tooltip when hovering over the element.
<code>loading_state</code>	Lists containing elements <code>'is_loading'</code> , <code>'prop_name'</code> , <code>'component_name'</code> . those elements have the following types: - <code>is_loading</code> (logical; optional): determines if the component is loading or not - <code>prop_name</code> (character; optional): holds which property is loading - <code>component_name</code> (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
<code>...</code>	wildcards allowed have the form: <code>'data-*'</code> , <code>'aria-*'</code>

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlP("Below is an example of htmlProgress"),
      htmlProgress(value = 80, max = 100)
    ))
  )

  app$run_server()
}
```



---

htmlQ	<i>Q component</i>
-------	--------------------

---

### Description

Q is a wrapper for the <q> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/q>

### Usage

```
htmlQ(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, cite=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
cite	Character. Contains a URI which points to the source of the quote or change.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.

hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: "data-*", "aria-*"

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlP(list(
        "The <q> tag defines a short quotation: ",
        htmlQ("This example text is wrapped in htmlQ")
      ))
    ))
  )

  app$run_server()
}
```

---

htmlRb

*Rb component*

---

### Description

Rb is a wrapper for the <rb> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/rb>

**Usage**

```
htmlRb(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.

loading\_state Lists containing elements 'is\_loading', 'prop\_name', 'component\_name'. those elements have the following types: - is\_loading (logical; optional): determines if the component is loading or not - prop\_name (character; optional): holds which property is loading - component\_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer

... wildcards allowed have the form: 'data-\*', 'aria-\*'

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlRuby(list(
        "\U{6f22}",
        htmlRp("("),
        htmlRt("kan"),
        htmlRp(")")
      )),
      htmlRuby(list(
        "\U{5b57}",
        htmlRp("("),
        htmlRt("ji"),
        htmlRp(")")
      ))
    ))
  )

  app$run_server()
}
```

---

htmlRp

*Rp component*

---

### Description

Rp is a wrapper for the <rp> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/rp>

**Usage**

```
htmlRp(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.

loading\_state Lists containing elements 'is\_loading', 'prop\_name', 'component\_name'. those elements have the following types: - is\_loading (logical; optional): determines if the component is loading or not - prop\_name (character; optional): holds which property is loading - component\_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer

... wildcards allowed have the form: 'data-\*', 'aria-\*'

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlRuby(list(
        "\U{6f22}",
        htmlRp("("),
        htmlRt("kan"),
        htmlRp(")")
      )),
      htmlRuby(list(
        "\U{5b57}",
        htmlRp("("),
        htmlRt("ji"),
        htmlRp(")")
      ))
    ))
  )

  app$run_server()
}
```

---

htmlRt

*Rt component*

---

### Description

Rt is a wrapper for the <rt> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/rt>

**Usage**

```
htmlRt(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.

loading\_state Lists containing elements 'is\_loading', 'prop\_name', 'component\_name'. those elements have the following types: - is\_loading (logical; optional): determines if the component is loading or not - prop\_name (character; optional): holds which property is loading - component\_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer

... wildcards allowed have the form: 'data-\*', 'aria-\*'

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlRuby(list(
        "\U{6f22}",
        htmlRp("("),
        htmlRt("kan"),
        htmlRp(")")
      )),
      htmlRuby(list(
        "\U{5b57}",
        htmlRp("("),
        htmlRt("ji"),
        htmlRp(")")
      ))
    ))
  )

  app$run_server()
}
```

---

htmlRtc

*Rtc component*

---

### Description

Rtc is a wrapper for the <rtc> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/rtc>



**Usage**

```
htmlRtc(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.

loading\_state Lists containing elements 'is\_loading', 'prop\_name', 'component\_name'. those elements have the following types: - is\_loading (logical; optional): determines if the component is loading or not - prop\_name (character; optional): holds which property is loading - component\_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer

... wildcards allowed have the form: 'data-\*', 'aria-\*'

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlRuby(list(
        "\u2661",
        htmlRtc(htmlRt("Heart"))
      ))
    ))
  )

  app$run_server()
}
```

---

htmlRuby

*Ruby component*

---

### Description

Ruby is a wrapper for the <ruby> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ruby>

### Usage

```
htmlRuby(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlRuby("\U{54d0}")
    ))
  )

  app$run_server()
}

```

htmlS

*S component***Description**

S is a wrapper for the <s> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/s>

**Usage**

```

htmlS(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info

<code>role</code>	Character. The ARIA role attribute
<code>accessKey</code>	Character. Keyboard shortcut to activate or add focus to the element.
<code>className</code>	Character. Often used with CSS to style elements with common properties.
<code>contentEditable</code>	Character. Indicates whether the element's content is editable.
<code>contextMenu</code>	Character. Defines the ID of a <code>&lt;menu&gt;</code> element which will serve as the element's context menu.
<code>dir</code>	Character. Defines the text direction. Allowed values are <code>ltr</code> (Left-To-Right) or <code>rtl</code> (Right-To-Left)
<code>draggable</code>	Character. Defines whether the element can be dragged.
<code>hidden</code>	A value equal to: <code>'hidden'</code> , <code>'hidden'   logical</code> . Prevents rendering of given element, while keeping child elements, e.g. <code>script</code> elements, active.
<code>lang</code>	Character. Defines the language used in the element.
<code>spellCheck</code>	Character. Indicates whether spell checking is allowed for the element.
<code>style</code>	Named list. Defines CSS styles which will override styles previously set.
<code>tabIndex</code>	Character. Overrides the browser's default tab order and follows the one specified instead.
<code>title</code>	Character. Text to be displayed in a tooltip when hovering over the element.
<code>loading_state</code>	Lists containing elements <code>'is_loading'</code> , <code>'prop_name'</code> , <code>'component_name'</code> . those elements have the following types: <code>- is_loading</code> (logical; optional): determines if the component is loading or not - <code>prop_name</code> (character; optional): holds which property is loading - <code>component_name</code> (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from <code>dash-renderer</code>
<code>...</code>	wildcards allowed have the form: <code>'data-*'</code> , <code>'aria-*'</code>

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlS("htmlS generates strikethrough text"),
      htmlP(),
      htmlB("htmlB generates bold text")
    ))
  )

  app$run_server()
}

```

htmlSamp

*Samp component***Description**

Samp is a wrapper for the <samp> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/samp>

**Usage**

```
htmlSamp(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.

lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlSamp("htmlSamp formats text to computer program output.")
    ))
  )

  app$run_server()
}
```

---

 htmlScript

*Script component*


---

**Description**

Script is a wrapper for the <script> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/script>

**Usage**

```
htmlScript(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, async=NULL,
charSet=NULL, crossOrigin=NULL, defer=NULL, integrity=NULL,
src=NULL, type=NULL, accessKey=NULL, className=NULL,
contentEditable=NULL, contextMenu=NULL, dir=NULL,
draggable=NULL, hidden=NULL, lang=NULL, spellCheck=NULL,
style=NULL, tabIndex=NULL, title=NULL, loading_state=NULL,
...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
async	A value equal to: 'async', 'async'   logical. Executes the script asynchronously.
charSet	Character. Declares the character encoding of the page or script.
crossOrigin	Character. How the element handles cross-origin requests
defer	A value equal to: 'defer', 'defer'   logical. Indicates that the script should be executed after the page has been parsed.
integrity	Character. Specifies a Subresource Integrity value that allows browsers to verify what they fetch.
src	Character. The URL of the embeddable content.
type	Character. Defines the type of the element.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)



draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
# This component is retained for compatibility reasons, but we suggest
# using Dash's capability for embedding scripts within the assets folder
# instead.
```

---

htmlSection

*Section component*

---

**Description**

Section is a wrapper for the <section> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/section>

**Usage**

```
htmlSection(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlSection(
        children = list(
          htmlH1("This is a section title"),
          htmlDiv("This is some text within a section")
        )
      )
    )
  )
)
app$run_server()
}

```

---

`htmlSelect`*Select component*

---

**Description**

Select is a wrapper for the `<select>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/select>

**Usage**

```

htmlSelect(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
autoComplete=NULL, autoFocus=NULL, disabled=NULL, form=NULL,
multiple=NULL, name=NULL, required=NULL, size=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

**Arguments**

<code>children</code>	A list of or a singular dash component, string or number. The children of this component
<code>id</code>	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.

n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
autoComplete	Character. Indicates whether controls in this form can by default have their values automatically completed by the browser.
autoFocus	A value equal to: 'autofocus', 'autofocus', 'autofocus'   logical. The element should be automatically focused after the page loaded.
disabled	A value equal to: 'disabled', 'disabled'   logical. Indicates whether the user can interact with the element.
form	Character. Indicates the form that is the owner of the element.
multiple	A value equal to: 'multiple', 'multiple'   logical. Indicates whether multiple values can be entered in an input of the type email or file.
name	Character. Name of the element. For example used by the server to identify the fields in form submits.
required	A value equal to: 'required', 'required'   logical. Indicates whether this element is required to fill out or not.
size	Character   numeric. Defines the width of the element (in pixels). If the element's type attribute is text or password then it's the number of characters.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.

loading\_state Lists containing elements 'is\_loading', 'prop\_name', 'component\_name'. those elements have the following types: - is\_loading (logical; optional): determines if the component is loading or not - prop\_name (character; optional): holds which property is loading - component\_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer

... wildcards allowed have the form: "data-\*", "aria-\*"

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlSelect(
        children = list(
          htmlOption("This is an option in htmlSelect"),
          htmlOption("But you might want to check out dccDropdown as well"),
          htmlOption("dccDropdown is part of the dashCoreComponents library")
        )
      )
    )
  )
  app$run_server()
}
```

---

htmlShadow

*Shadow component*

---

### Description

Shadow is a wrapper for the <shadow> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/shadow>

### Usage

```
htmlShadow(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
```

contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL, lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL, title=NULL, loading\_state=NULL, ...)

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
# The Shadow element requires a browser that supports
# Web Components. It is experimental and should be used
# with caution; it is now considered an obsolete element
# within the Web Components suite. It is retained here
# solely for backwards compatibility reasons.
#
# For more information, please see the MDN link above.
```

---

htmlSlot

*Slot component*


---

**Description**

Slot is a wrapper for the <slot> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/slot>

**Usage**

```
htmlSlot(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in call-backs. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute

accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
# Please see https://mdn.github.io/web-components-examples/element-details/
# and https://github.com/mdn/web-components-examples/tree/master/element-details
# for a useful example of this element (with accompanying JavaScript) in action.
```

---

htmlSmall

*Small component*


---

**Description**

Small is a wrapper for the <small> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/small>



**Usage**

```
htmlSmall(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.

loading\_state Lists containing elements 'is\_loading', 'prop\_name', 'component\_name'. those elements have the following types: - is\_loading (logical; optional): determines if the component is loading or not - prop\_name (character; optional): holds which property is loading - component\_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer

... wildcards allowed have the form: 'data-\*', 'aria-\*'

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      "This is normal text",
      htmlBr(),
      htmlSmall("And this is text in an htmlSmall component")
    )
  )
)

app$run_server()
}
```

---

htmlSource

*Source component*

---

### Description

Source is a wrapper for the <source> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/source>

### Usage

```
htmlSource(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, media=NULL,
sizes=NULL, src=NULL, srcSet=NULL, type=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
media	Character. Specifies a hint of the media for which the linked resource was designed.
sizes	Character.
src	Character. The URL of the embeddable content.
srcSet	Character. One or more responsive image candidates.
type	Character. Defines the type of the element.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer

... wildcards allowed have the form: `'data-*', 'aria-*'`

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
# The URL below has been chunked to comply with CRAN
# requirements; the use of file.path is optional and not required
# for this component.
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      "Resize your browser window to see the image source change based on the browser width",
      htmlBr(),
      htmlPicture(
        list(
          htmlSource(
            media = "(min-width: 1000px)",
            srcSet = "https://apod.nasa.gov/apod/image/1907/FishheadNebula_Pham_2401.jpg"
          ),
          htmlImg(
            src = "https://apod.nasa.gov/apod/image/1907/ngc3576_campbell_1824.jpg"
          )
        )
      )
    )
  )
  app$run_server()
}
```

---

htmlSpacer

*Spacer component*

---

### Description

Spacer is a wrapper for the `<spacer>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/spacer>

**Usage**

```
htmlSpacer(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.

loading\_state Lists containing elements 'is\_loading', 'prop\_name', 'component\_name'. those elements have the following types: - is\_loading (logical; optional): determines if the component is loading or not - prop\_name (character; optional): holds which property is loading - component\_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer

... wildcards allowed have the form: 'data-\*', 'aria-\*'

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
# This component provides an HTML element that is now obsolete
# and not supported by modern web browsers; it is retained for
# backwards compatibility.
```

---

htmlSpan

*Span component*

---

### Description

Span is a wrapper for the `<span>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/span>

### Usage

```
htmlSpan(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

### Arguments

children A list of or a singular dash component, string or number. The children of this component

id Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.

n\_clicks Numeric. An integer that represents the number of times that this element has been clicked on.

n\_clicks\_timestamp Numeric. An integer that represents the time (in ms since 1970) at which n\_clicks changed. This can be used to tell which button was changed most recently.

key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      "This is some text",
      htmlBr(),
```

```

    htmlSpan(
      children = "And some text within an italicized span",
      style = list(fontStyle = "italic")
    )
  )
)
)
app$run_server()
}

```

---

htmlStrike

*Strike component*


---

### Description

Strike is a wrapper for the `<strike>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/strike>

### Usage

```

htmlStrike(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.



contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      "This is normal text",
      htmlStrike("Text within an htmlStrike element will be stricken out")
    )
  )
)

app$run_server()
}

```

---

htmlStrong	<i>Strong component</i>
------------	-------------------------

---

### Description

Strong is a wrapper for the `<strong>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/strong>

### Usage

```
htmlStrong(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <code>&lt;menu&gt;</code> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are <code>ltr</code> (Left-To-Right) or <code>rtl</code> (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: <code>'hidden'</code> , <code>'hidden'   logical</code> . Prevents rendering of given element, while keeping child elements, e.g. script elements, active.

lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      "This is normal text",
      htmlBr(),
      htmlStrong("Text within an htmlStrong element will be Bold")
    )
  )
)

app$run_server()
}

```

---

htmlSub

*Sub component*


---

**Description**

Sub is a wrapper for the <sub> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/sub>

**Usage**

```
htmlSub(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.

loading\_state Lists containing elements 'is\_loading', 'prop\_name', 'component\_name'. those elements have the following types: - is\_loading (logical; optional): determines if the component is loading or not - prop\_name (character; optional): holds which property is loading - component\_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer

... wildcards allowed have the form: 'data-\*', 'aria-\*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      "This is normal text ",
      htmlSub("And this is subscript text within an htmlSub")
    )
  )
)

app$run_server()
}
```

---

htmlSummary

*Summary component*


---

**Description**

Summary is a wrapper for the <summary> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/summary>

**Usage**

```
htmlSummary(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlDetails(
        children = list(
          htmlSummary(
            children = "Within a details element, the summary can act as a clickable description"
          ),
          "And the rest is hidden until the summary is clicked"
        )
      )
    )
  )
  app$run_server()
}

```

---

`htmlSup`*Sup component*

---

**Description**

Sup is a wrapper for the `<sup>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/sup>

**Usage**

```

htmlSup(children=NULL, id=NULL, n_clicks=NULL,
         n_clicks_timestamp=NULL, key=NULL, role=NULL,
         accessKey=NULL, className=NULL, contentEditable=NULL,
         contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
         lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
         title=NULL, loading_state=NULL, ...)

```

**Arguments**

<code>children</code>	A list of or a singular dash component, string or number. The children of this component
<code>id</code>	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.

<code>n_clicks</code>	Numeric. An integer that represents the number of times that this element has been clicked on.
<code>n_clicks_timestamp</code>	Numeric. An integer that represents the time (in ms since 1970) at which <code>n_clicks</code> changed. This can be used to tell which button was changed most recently.
<code>key</code>	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
<code>role</code>	Character. The ARIA role attribute
<code>accessKey</code>	Character. Keyboard shortcut to activate or add focus to the element.
<code>className</code>	Character. Often used with CSS to style elements with common properties.
<code>contentEditable</code>	Character. Indicates whether the element's content is editable.
<code>contextMenu</code>	Character. Defines the ID of a <code>&lt;menu&gt;</code> element which will serve as the element's context menu.
<code>dir</code>	Character. Defines the text direction. Allowed values are <code>ltr</code> (Left-To-Right) or <code>rtl</code> (Right-To-Left)
<code>draggable</code>	Character. Defines whether the element can be dragged.
<code>hidden</code>	A value equal to: <code>'hidden'</code> , <code>'hidden'   logical</code> . Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
<code>lang</code>	Character. Defines the language used in the element.
<code>spellCheck</code>	Character. Indicates whether spell checking is allowed for the element.
<code>style</code>	Named list. Defines CSS styles which will override styles previously set.
<code>tabIndex</code>	Character. Overrides the browser's default tab order and follows the one specified instead.
<code>title</code>	Character. Text to be displayed in a tooltip when hovering over the element.
<code>loading_state</code>	Lists containing elements <code>'is_loading'</code> , <code>'prop_name'</code> , <code>'component_name'</code> . those elements have the following types: <code>- is_loading</code> (logical; optional): determines if the component is loading or not - <code>prop_name</code> (character; optional): holds which property is loading - <code>component_name</code> (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
<code>...</code>	wildcards allowed have the form: <code>'data-*'</code> , <code>'aria-*'</code>

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)
```



```

app <- Dash$new()

app$layout(
  htmlDiv(list(
    "This is normal text",
    htmlSup("And this is superscript text within an htmlSup")
  )
)
)

app$run_server()
}

```

---

htmlTable

*Table component*


---

## Description

Table is a wrapper for the <table> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/table>

## Usage

```

htmlTable(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, summary=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

## Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute

summary	Character.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      "You can create a table with htmlTable:",
      htmlBr(),
      htmlTable(
        list(
          htmlTr(
            list(
```



key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      "Within an htmlTable, htmlTbody specifies rows for the table body",
      htmlBr(),
```



id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
colSpan	Character   numeric. The colspan attribute defines the number of columns a cell should span.
headers	Character. IDs of the <th> elements which applies to this element.
rowSpan	Character   numeric. Defines the number of rows a table cell should span over.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

## Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      "Within an htmlTable, individual cells can be made with htmlTd",
      htmlBr(),
      htmlTable(
        list(
          htmlTr(
            list(
              htmlTh("Header 1"),
              htmlTh("Header 2")
            )
          ),
          htmlTr(
            list(
              htmlTd("this is a cell"),
              htmlTd("this is another cell")
            )
          )
        )
      )
    )
  )
  app$run_server()
}
```

---

htmlTemplate	<i>Template component</i>
--------------	---------------------------

---

## Description

Template is a wrapper for the <template> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/template>

## Usage

```
htmlTemplate(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values



## Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      "You can create an HTML template to be populated later via js",
      htmlBr(),
      htmlTable(
        id = "myTable",
        htmlTr(
          list(
            htmlTh("Header 1"),
            htmlTh("Header 2")
          )
        )
      ),
    ),
    htmlTemplate(
      id = "myRowTemplate",
      htmlTr(
        list(
          htmlTd(className = "someRowValue"),
          htmlTd()
        )
      )
    )
  )
)
app$run_server()
}
```

---

htmlTextarea

*Textarea component*

---

## Description

Textarea is a wrapper for the `<textarea>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/textarea>

## Usage

```
htmlTextarea(children=NULL, id=NULL, n_clicks=NULL,
             n_clicks_timestamp=NULL, key=NULL, role=NULL,
             autoComplete=NULL, autoFocus=NULL, cols=NULL, disabled=NULL,
```

```

form=NULL, inputMode=NULL, maxLength=NULL, minLength=NULL,
name=NULL, placeholder=NULL, readOnly=NULL, required=NULL,
rows=NULL, wrap=NULL, accessKey=NULL, className=NULL,
contentEditable=NULL, contextMenu=NULL, dir=NULL,
draggable=NULL, hidden=NULL, lang=NULL, spellCheck=NULL,
style=NULL, tabIndex=NULL, title=NULL, loading_state=NULL,
...)
```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
autoComplete	Character. Indicates whether controls in this form can by default have their values automatically completed by the browser.
autoFocus	A value equal to: 'autofocus', 'autofocus', 'autofocus'   logical. The element should be automatically focused after the page loaded.
cols	Character   numeric. Defines the number of columns in a textarea.
disabled	A value equal to: 'disabled', 'disabled'   logical. Indicates whether the user can interact with the element.
form	Character. Indicates the form that is the owner of the element.
inputMode	Character. Provides a hint as to the type of data that might be entered by the user while editing the element or its contents. The attribute can be used with form controls (such as the value of textarea elements), or in elements in an editing host (e.g., using contenteditable attribute).
maxLength	Character   numeric. Defines the maximum number of characters allowed in the element.
minLength	Character   numeric. Defines the minimum number of characters allowed in the element.
name	Character. Name of the element. For example used by the server to identify the fields in form submits.
placeholder	Character. Provides a hint to the user of what can be entered in the field.
readOnly	Character. Indicates whether the element can be edited.

required	A value equal to: 'required', 'required'   logical. Indicates whether this element is required to fill out or not.
rows	Character   numeric. Defines the number of rows in a text area.
wrap	Character. Indicates whether the text should be wrapped.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlTextarea(
        rows = 4, cols = 50,
```

```

        children = "A text area allows users to input text"
      )
    )
  )
}

app$run_server()
}

```

---

htmlTfoot

*Tfoot component*


---

### Description

Tfoot is a wrapper for the <tfoot> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/tfoot>

### Usage

```

htmlTfoot(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.

contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: "data-*", "aria-*"

### Value

named list of JSON elements corresponding to React.js properties and their values

### Examples

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      "Within an htmlTable, you can create footer rows with htmlTfoot",
      htmlBr(),
      htmlTable(
        list(
          htmlThead(
            htmlTr(
              htmlTh("This is in the header of the table")
            )
          ),
          htmlTbody(
            htmlTr(
              htmlTd("This is in the body of the table")
            )
          )
        )
      )
    )
  )
}
```



role	Character. The ARIA role attribute
colSpan	Character   numeric. The colspan attribute defines the number of columns a cell should span.
headers	Character. IDs of the <th> elements which applies to this element.
rowSpan	Character   numeric. Defines the number of rows a table cell should span over.
scope	Character. Defines the cells that the header test (defined in the th element) relates to.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()
```

```

app$layout(
  htmlDiv(list(
    htmlTable(
      list(
        # the following row contains headers
        htmlTr(
          list(
            htmlTh("Header 1"),
            htmlTh("Header 2")
          )
        )
      )
    )
  )
)
app$run_server()
}

```

---

htmlThead

*Thead component*


---

### Description

Thead is a wrapper for the <thead> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/thead>

### Usage

```

htmlThead(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.



key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      "Within an htmlTable, you can create a header with htmlThead",
      htmlBr(),
```



n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
dateTime	Character. Indicates the date and time associated with the element.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlP(
        list(
          "It might be useful to wrap dates like ",
          htmlTime(dateTime = "2019-07-29", children = "July 29th"),
          " in an htmlTime to make your datetime strings machine-readable."
        )
      )
    )
  )
  app$run_server()
}

```

---

`htmlTitle`*Title component*

---

**Description**

Title is a wrapper for the <title> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/title>

**Usage**

```

htmlTitle(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

**Arguments**

<code>children</code>	A list of or a singular dash component, string or number. The children of this component
<code>id</code>	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
<code>n_clicks</code>	Numeric. An integer that represents the number of times that this element has been clicked on.

n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: <code>'data-*'</code> , <code>'aria-*'</code>

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
# This component's effects will be overridden by the index
# template in Dash for R. We suggest using Dash's API to
# set the page title instead:
#
# app$title('My page title')
```

---

htmlTr	<i>Tr component</i>
--------	---------------------

---

### Description

Tr is a wrapper for the `<tr>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/tr>

### Usage

```
htmlTr(children=NULL, id=NULL, n_clicks=NULL,
        n_clicks_timestamp=NULL, key=NULL, role=NULL,
        accessKey=NULL, className=NULL, contentEditable=NULL,
        contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
        lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
        title=NULL, loading_state=NULL, ...)
```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <code>&lt;menu&gt;</code> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are <code>ltr</code> (Left-To-Right) or <code>rtl</code> (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: <code>'hidden'</code> , <code>'hidden'   logical</code> . Prevents rendering of given element, while keeping child elements, e.g. script elements, active.

lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      "Within an htmlTable, individual rows can be made with htmlTr",
      htmlBr(),
      htmlTable(
        list(
          # the following row contains headers
          htmlTr(
            list(
              htmlTh("Header 1"),
              htmlTh("Header 2")
            )
          ),
          # the following row contains cells
          htmlTr(
            list(
              htmlTd("this is a cell"),
              htmlTd("this is another cell")
            )
          )
        )
      )
    )
  )
}

```

```

    )
    app$run_server()
}

```

---

htmlTrack

*Track component*


---

### Description

Track is a wrapper for the `<track>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/track>

### Usage

```

htmlTrack(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, default=NULL,
kind=NULL, label=NULL, src=NULL, srcLang=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
default	A value equal to: 'default', 'default'   logical. Indicates that the track should be enabled unless the user's preferences indicate something different.
kind	Character. Specifies the kind of text track.
label	Character. Specifies a user-readable title of the element.
src	Character. The URL of the embeddable content.
srcLang	Character.



accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

## Value

named list of JSON elements corresponding to React.js properties and their values

## Examples

```
# The URL below has been chunked to comply with CRAN
# requirements; the use of file.path is optional and not required
# for this component.
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlVideo(children = list(
        htmlSource(src = file.path("https://interactive-examples.mdn.mozilla.net",
          "media/examples",
          "friday.mp4",
```

```

        fsep = "/"),
        type = 'video/mp4'),
    htmlTrack(kind = 'captions',
        srcLang = 'en',
        src = file.path("https://interactive-examples.mdn.mozilla.net",
            "media/examples",
            "friday.vtt",
            fsep = "/"),
        default = 'default',
        label = 'English')
    ),
    controls = TRUE
)
)
)
)
app$run_server()
}

```

---

htmlU

*U component*


---

## Description

U is a wrapper for the <u> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/u>

## Usage

```

htmlU(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

## Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.

key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlU("Wrap your text in htmlU to have it underlined")
    )
  )
}
```

```

    )
  )
  app$run_server()
}

```

---

htmlUI

*UI component*


---

### Description

UI is a wrapper for the `<ul>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/ul>

### Usage

```

htmlUI(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <code>&lt;menu&gt;</code> element which will serve as the element's context menu.

<code>dir</code>	Character. Defines the text direction. Allowed values are <code>ltr</code> (Left-To-Right) or <code>rtl</code> (Right-To-Left)
<code>draggable</code>	Character. Defines whether the element can be dragged.
<code>hidden</code>	A value equal to: <code>'hidden'</code> , <code>'hidden'   logical</code> . Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
<code>lang</code>	Character. Defines the language used in the element.
<code>spellCheck</code>	Character. Indicates whether spell checking is allowed for the element.
<code>style</code>	Named list. Defines CSS styles which will override styles previously set.
<code>tabIndex</code>	Character. Overrides the browser's default tab order and follows the one specified instead.
<code>title</code>	Character. Text to be displayed in a tooltip when hovering over the element.
<code>loading_state</code>	Lists containing elements <code>'is_loading'</code> , <code>'prop_name'</code> , <code>'component_name'</code> . those elements have the following types: - <code>is_loading</code> (logical; optional): determines if the component is loading or not - <code>prop_name</code> (character; optional): holds which property is loading - <code>component_name</code> (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
<code>...</code>	wildcards allowed have the form: <code>'data-*'</code> , <code>'aria-*'</code>

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      "You can make an unordered list with htmlUI",
      htmlBr(),
      htmlUL(
        children = list(
          htmlLi("Some item"),
          htmlLi("Some other item")
        )
      )
    )
  )
)

app$run_server()
}
```

---

htmlVar	<i>Var component</i>
---------	----------------------

---

### Description

Var is a wrapper for the <var> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/var>

### Usage

```
htmlVar(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

### Arguments

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.

lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      "You can use htmlVar to represent the name of a variable",
      htmlBr(),
      htmlVar("myVariable")
    ))
  )
}

app$run_server()
}

```

---

htmlVideo

*Video component*


---

**Description**

Video is a wrapper for the <video> HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/video>

**Usage**

```
htmlVideo(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL, autoPlay=NULL,
controls=NULL, crossOrigin=NULL, height=NULL, loop=NULL,
muted=NULL, poster=NULL, preload=NULL, src=NULL, width=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in call-backs. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
autoPlay	A value equal to: 'autoplay', 'autoplay', 'autoplay'   logical. The audio or video should play as soon as possible.
controls	A value equal to: 'controls', 'controls'   logical. Indicates whether the browser should show playback controls to the user.
crossOrigin	Character. How the element handles cross-origin requests
height	Character   numeric. Specifies the height of elements listed here. For all other elements, use the CSS height property. Note: In some instances, such as <div>, this is a legacy attribute, in which case the CSS height property should be used instead.
loop	A value equal to: 'loop', 'loop'   logical. Indicates whether the media should start playing from the start when it's finished.
muted	A value equal to: 'muted', 'muted'   logical. Indicates whether the audio will be initially silenced on page load.
poster	Character. A URL indicating a poster frame to show until the user plays or seeks.
preload	Character. Indicates whether the whole resource, parts of it or nothing should be preloaded.
src	Character. The URL of the embeddable content.



width	Character   numeric. For the elements listed here, this establishes the element's width. Note: For all other instances, such as <div>, this is a legacy attribute, in which case the CSS width property should be used instead.
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```
if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlVideo(
        src = file.path('https://ia800303.us.archive.org',
          '18/items/bacteria_friend_and_foe',
          'bacteria_friend_and_foe_512kb.mp4',
```

```

        fsep = '/'),
        controls = TRUE,
        title = "Bacteria: Friend and Foe"
    )
  )
)
app$run_server()
}

```

htmlWbr

*Wbr component***Description**

Wbr is a wrapper for the `<wbr>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/wbr>

**Usage**

```

htmlWbr(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)

```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.

contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <menu> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are ltr (Left-To-Right) or rtl (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: 'hidden', 'hidden'   logical. Prevents rendering of given element, while keeping child elements, e.g. script elements, active.
lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      "In a long string, it might be a good idea to add an htmlWbr to specify word breaks",
      htmlP("Thisverylongstringwithnowhitespaceswon'tlookverygood"),
      htmlWbr(),
      htmlP("butatleastyoucanspecifya'natural'placeforthestringtobebrokenup")
    )
  )
)

app$run_server()
}

```

htmlXmp

*Xmp component***Description**

Xmp is a wrapper for the `<xmp>` HTML5 element. For detailed attribute info see: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/xmp>

**Usage**

```
htmlXmp(children=NULL, id=NULL, n_clicks=NULL,
n_clicks_timestamp=NULL, key=NULL, role=NULL,
accessKey=NULL, className=NULL, contentEditable=NULL,
contextMenu=NULL, dir=NULL, draggable=NULL, hidden=NULL,
lang=NULL, spellCheck=NULL, style=NULL, tabIndex=NULL,
title=NULL, loading_state=NULL, ...)
```

**Arguments**

children	A list of or a singular dash component, string or number. The children of this component
id	Character. The ID of this component, used to identify dash components in callbacks. The ID needs to be unique across all of the components in an app.
n_clicks	Numeric. An integer that represents the number of times that this element has been clicked on.
n_clicks_timestamp	Numeric. An integer that represents the time (in ms since 1970) at which n_clicks changed. This can be used to tell which button was changed most recently.
key	Character. A unique identifier for the component, used to improve performance by React.js while rendering components See <a href="https://reactjs.org/docs/lists-and-keys.html">https://reactjs.org/docs/lists-and-keys.html</a> for more info
role	Character. The ARIA role attribute
accessKey	Character. Keyboard shortcut to activate or add focus to the element.
className	Character. Often used with CSS to style elements with common properties.
contentEditable	Character. Indicates whether the element's content is editable.
contextMenu	Character. Defines the ID of a <code>&lt;menu&gt;</code> element which will serve as the element's context menu.
dir	Character. Defines the text direction. Allowed values are <code>ltr</code> (Left-To-Right) or <code>rtl</code> (Right-To-Left)
draggable	Character. Defines whether the element can be dragged.
hidden	A value equal to: <code>'hidden'</code> , <code>'hidden'   logical</code> . Prevents rendering of given element, while keeping child elements, e.g. script elements, active.

lang	Character. Defines the language used in the element.
spellCheck	Character. Indicates whether spell checking is allowed for the element.
style	Named list. Defines CSS styles which will override styles previously set.
tabIndex	Character. Overrides the browser's default tab order and follows the one specified instead.
title	Character. Text to be displayed in a tooltip when hovering over the element.
loading_state	Lists containing elements 'is_loading', 'prop_name', 'component_name'. those elements have the following types: - is_loading (logical; optional): determines if the component is loading or not - prop_name (character; optional): holds which property is loading - component_name (character; optional): holds the name of the component that is loading. Object that holds the loading state object coming from dash-renderer
...	wildcards allowed have the form: 'data-*', 'aria-*'

**Value**

named list of JSON elements corresponding to React.js properties and their values

**Examples**

```

if (interactive() && require(dash)) {
  library(dash)
  library(dashHtmlComponents)

  app <- Dash$new()

  app$layout(
    htmlDiv(list(
      htmlXmp("xmp elements will be rendered in monospace font"),
      htmlXmp("Note that this element is obsolete in HTML5"),
      htmlA(
        "See this for more details",
        href = "https://developer.mozilla.org/en-US/docs/Web/HTML/Element/xmp"
      )
    )
  )
)

app$run_server()
}

```

# Index

dashHtmlComponents  
    (dashHtmlComponents-package), 5  
dashHtmlComponents-package, 5

htmlA, 5  
htmlAbbr, 7  
htmlAcronym, 9  
htmlAddress, 11  
htmlArea, 12  
htmlArticle, 15  
htmlAside, 17  
htmlAudio, 19  
htmlB, 21  
htmlBase, 23  
htmlBasefont, 25  
htmlBdi, 27  
htmlBdo, 28  
htmlBig, 30  
htmlBlink, 32  
htmlBlockquote, 34  
htmlBr, 36  
htmlButton, 38  
htmlCanvas, 40  
htmlCaption, 42  
htmlCenter, 44  
htmlCite, 46  
htmlCode, 48  
htmlCol, 50  
htmlColgroup, 52  
htmlCommand, 54  
htmlContent, 56  
htmlData, 57  
htmlDatalist, 59  
htmlDd, 61  
htmlDel, 63  
htmlDetails, 65  
htmlDfn, 67  
htmlDialog, 69  
htmlDiv, 70  
htmlDl, 72  
htmlDt, 74  
htmlElement, 76  
htmlEm, 77  
htmlEmbed, 79  
htmlFieldset, 81  
htmlFigcaption, 83  
htmlFigure, 85  
htmlFont, 87  
htmlFooter, 88  
htmlForm, 90  
htmlFrame, 93  
htmlFrameset, 94  
htmlH1, 96  
htmlH2, 98  
htmlH3, 99  
htmlH4, 101  
htmlH5, 103  
htmlH6, 105  
htmlHeader, 107  
htmlHgroup, 109  
htmlHr, 110  
htmlI, 112  
htmlIframe, 114  
htmlImg, 116  
htmlIns, 118  
htmlIsindex, 120  
htmlKbd, 121  
htmlKeygen, 123  
htmlLabel, 125  
htmlLegend, 127  
htmlLi, 129  
htmlLink, 131  
htmlListing, 133  
htmlMain, 135  
htmlMapEl, 137  
htmlMark, 139  
htmlMarquee, 141  
htmlMeta, 143  
htmlMeter, 145

htmlMulticol, 147  
htmlNav, 148  
htmlNextid, 150  
htmlNobr, 152  
htmlNoscript, 154  
htmlObjectEl, 155  
htmlOL, 158  
htmlOptgroup, 160  
htmlOption, 162  
htmlOutput, 164  
htmlP, 165  
htmlParam, 167  
htmlPicture, 169  
htmlPlaintext, 171  
htmlPre, 173  
htmlProgress, 175  
htmlQ, 177  
htmlRb, 178  
htmlRp, 180  
htmlRt, 182  
htmlRtc, 184  
htmlRuby, 186  
htmlS, 188  
htmlSamp, 190  
htmlScript, 191  
htmlSection, 193  
htmlSelect, 195  
htmlShadow, 197  
htmlSlot, 199  
htmlSmall, 200  
htmlSource, 202  
htmlSpacer, 204  
htmlSpan, 206  
htmlStrike, 208  
htmlStrong, 210  
htmlSub, 211  
htmlSummary, 213  
htmlSup, 215  
htmlTable, 217  
htmlTbody, 219  
htmlTd, 221  
htmlTemplate, 223  
htmlTextarea, 225  
htmlTfoot, 228  
htmlTh, 230  
htmlThead, 232  
htmlTime, 234  
htmlTitle, 236  
htmlTr, 238  
htmlTrack, 240  
htmlU, 242  
htmlUL, 244  
htmlVar, 246  
htmlVideo, 247  
htmlWbr, 250  
htmlXmp, 252