Package ‘aos’

April 29, 2020

Title  Animate on Scroll Library for ‘shiny’
Version 0.1.0
Description Trigger animation effects on scroll on any HTML element of ‘shiny’ and ‘rmarkdown’, such as any text or plot, thanks to the ‘AOS’ Animate On Scroll jQuery library.
License MIT + file LICENSE
Encoding UTF-8
LazyData true
RoxygenNote 7.1.0
Imports jsonlite, htmltools
Suggests shiny
BugReports https://github.com/lgnbhl/aos/issues
NeedsCompilation no
Author Félix Luginbuhl [aut, cre]
Maintainer Félix Luginbuhl <felix.luginbuhl@protonmail.ch>
Repository CRAN
Date/Publication 2020-04-29 14:30:05 UTC

R topics documented:

aos .................................................. 2
use_aos ........................................ 3
use_aos_refresh .............................. 4
use_aos_refresh_hard ....................... 5

Index 6
Animate element on scroll

**Description**

Animate on scroll a Shiny or R Markdown element.

**Usage**

```r
aos(
  element,
  animation,
  offset = "120",
  duration = "400",
  easing = "ease",
  delay = "0",
  anchor = "",
  anchor_placement = "top-bottom",
  once = FALSE,
  ...
)
```

**Arguments**

- `element`: An shiny or rmarkdown element.
- `animation`: An animation from AOS
- `offset`: string. Change offset to trigger animations sooner or later, px
- `duration`: string. Duration of animation in ms
- `easing`: string. Choose timing function to ease elements in different ways
- `delay`: string. Delay animation in ms
- `anchor`: string. Anchor placement
- `anchor_placement`: string. Anchor placement - which one position of element on the screen should trigger animation
- `once`: boolean.
- `...`: Additional class element

**Value**

Javascript code animating the element.
Examples

```r
if (interactive()) {
library(shiny)
shinyApp(
  ui = fluidPage(
    use_aos(), # add use_aos() in the UI
    aos(h1("Shiny with AOS - Animate On Scroll"), animation = "fade-zoom-in"),
    aos(textOutput("text"), animation = "fade-up")
  ),
  server <- function(input, output, session){
    output$text <- renderText({
      print("An animated text.")
    })
  }
)
}
```

---

### `use_aos`

**Use aos**

#### Description

Use aos

#### Usage

```r
use_aos(
  disable = FALSE,
  startEvent = "DOMContentLoaded",
  initClassName = "aos-init",
  animatedClassName = "aos-animate",
  useClassNames = FALSE,
  disableMutationObserver = FALSE,
  debounceDelay = "50",
  throttleDelay = "99",
  offset = "120",
  duration = "400",
  easing = "ease",
  delay = "0"
)
```

#### Arguments

- **disable** string. Condition when AOS should be disabled, for example 'mobile'
- **startEvent** string. Name of the event dispatched on the document, that AOS should initialize on
- **initClassName** string. Class applied after initialization
animatedClassName
  string. Class applied on animation
useClassNames
  boolean. If true, will add content of data-aos as classes on scroll
disableMutationObserver
  boolean. Disables automatic mutations detections (advanced)
debounceDelay
  string. The delay on debounce used while resizing window (advanced)
throttleDelay
  string. The delay on throttle used while scrolling the page (advanced)
offset
  string. Change offset to trigger animations sooner or later, in px
duration
  string. Duration of animation in ms.
easing
  string. Timing function to animate elements nicely
delay
  string. Delay animation (ms)

Value
  Javascript code initializing aos with dependencies.

Examples
  if (interactive()) {
    library(shiny)
    shinyApp(
      ui = fluidPage(
        use_aos(), # add use_aos() in the UI
        aos(h1("Shiny with AOS - Animate On Scroll"), animation = "fade-zoom-in"),
        aos(textOutput("text"), animation = "fade-up")
      ),
      server <- function(input, output, session){
        output$text <- renderText({ print("An animated text." ) })
      }
    )
  }

---

use_aos_refresh
  Refresh aos

Description
  Recalculate all offsets and positions of elements (called on window resize).

Usage
  use_aos_refresh()
**use_aos_refresh_hard**

**Details**
By default AOS is watching for DOM changes and if there are any new elements loaded asynchronously or when something is removed from DOM it calls refreshHard automatically. In browsers that don’t support MutationObserver like IE you might need to call AOS.refreshHard() by yourself.

**Value**
Javascript code

**Examples**
```javascript
use_aos_refresh()
```

---

**use_aos_refresh_hard**  *Refresh hard aos*

**Description**
Reinit array with AOS elements and trigger refresh (called on DOM changes that are related to aos elements)

**Usage**
```javascript
use_aos_refresh_hard()
```

**Value**
Javascript code

**Examples**
```javascript
use_aos_refresh_hard()
```
Index

aos, 2
use_aos, 3
use_aos_refresh, 4
use_aos_refresh_hard, 5