

# Package ‘fontawesome’

March 28, 2021

**Type** Package

**Version** 0.2.0

**Title** Easily Work with 'Font Awesome' Icons

**Description** Easily and flexibly insert 'Font Awesome' icons into 'R Markdown' documents and 'Shiny' apps. These icons can be inserted into HTML content through inline 'SVG' tags or 'i' tags. There is also a utility function for exporting 'Font Awesome' icons as 'PNG' images for those situations where raster graphics are needed.

**License** MIT + file LICENSE

**URL** <https://github.com/rstudio/fontawesome>

**BugReports** <https://github.com/rstudio/fontawesome/issues>

**Encoding** UTF-8

**ByteCompile** true

**RoxygenNote** 7.1.1

**Depends** R (>= 3.3.0)

**Imports** rlang (>= 0.4.10), htmltools (>= 0.5.1.1)

**Suggests** covr, knitr, testthat, rsvg

**NeedsCompilation** no

**Author** Richard Iannone [aut, cre] (<<https://orcid.org/0000-0003-3925-190X>>),  
Christophe Dervieux [ctb] (<<https://orcid.org/0000-0003-4474-2498>>),  
Winston Chang [ctb],  
Dave Gandy [ctb, cph] (Font-Awesome font),  
RStudio [cph, fnd]

**Maintainer** Richard Iannone <[rich@rstudio.com](mailto:rich@rstudio.com)>

**Repository** CRAN

**Date/Publication** 2021-03-28 06:10:03 UTC

## R topics documented:

fa	2
fa_html_dependency	3
fa_i	4
fa_metadata	5
fa_png	6

<b>Index</b>	<b>8</b>
--------------	----------

---

fa	<i>Generate Font Awesome icons as SVGs</i>
----	--

---

### Description

Add one or more Font Awesome icons as SVGs contained within `<svg>...</svg>`. We can optionally set certain style attributes. The `fa()` function can be used directly within inline evaluations of R code in R Markdown documents.

### Usage

```
fa(
  name,
  fill = NULL,
  fill_opacity = NULL,
  stroke = NULL,
  stroke_width = NULL,
  stroke_opacity = NULL,
  height = NULL,
  width = NULL,
  margin_right = NULL,
  position = NULL,
  title = NULL,
  ally = c("desc", "sem", "none")
)
```

### Arguments

name	The name of the Font Awesome icon. This could be as a short name (e.g., "npm", "drum", etc.), or, a full name (e.g., "fab fa-npm", "fas fa-drum", etc.). The names should correspond to current Version 5 Font Awesome names. A list of short and full names can be accessed through the <code>fa_metadata()</code> function with <code>fa_metadata()\$icon_names</code> and <code>fa_metadata()\$icon_names_full</code> . If supplying a Version 4 icon name, it will be internally translated to the Version 5 icon name and a Version 5 icon will be returned. A data frame containing the short names that changed from version 4 ( <code>v4_name</code> ) to version 5 ( <code>v5_name</code> ) can be obtained by using <code>fa_metadata()\$v4_v5_name_tbl</code> . If multiple names are provided as a character vector, then multiple icons will be produced by concatenating together the SVG tags.
------	---

fill, fill_opacity	The fill color of the icon can be set with <code>fill</code> . If not provided then the default value of <code>"currentColor"</code> is applied so that the SVG fill matches the color of the parent HTML element's <code>color</code> attribute. The opacity level of the SVG fill can be controlled with a decimal value between 0 and 1.
stroke, stroke_width, stroke_opacity	The stroke options allow for setting the color, width, and opacity of the SVG outline stroke. By default, the stroke width is very small at <code>"1px"</code> so a size adjustment with <code>"stroke_width"</code> can be useful. The <code>"stroke_opacity"</code> value can be any decimal values between 0 and 1 (bounds included).
height, width	The height and width style attributes of the rendered SVG. If nothing is provided for height then a default value of <code>"1em"</code> will be applied. If a width isn't given, then it will be calculated in units of <code>"em"</code> on the basis of the icon's SVG <code>"viewBox"</code> dimensions.
margin_right	The length value for the margin that's right of the icon. By default, <code>"0.2rem"</code> is used.
position	The value for the <code>position</code> style attribute. By default, <code>"relative"</code> is used here.
title	An option for populating the SVG <code>'title'</code> attribute, which provides on-hover text for the icon. By default, no title text is given to the icon. If <code>ally == "semantic"</code> then title text will be automatically given to the rendered icon, however, providing text here will override that.
ally	Cases that distinguish the role of the icon and inform which accessibility attributes to be used. Icons can either be <code>"desc"</code> (decorative, the default case) or <code>"sem"</code> (semantic). Using <code>"none"</code> will result in no accessibility features for the icon.

**Value**

A fontawesome object.

**Examples**

```
if (interactive()) {
  # Create a Font Awesome SVG icon
  fa(name = "r-project")
}
```

---

fa\_html\_dependency      *Use a Font Awesome html\_dependency*

---

**Description**

The `fa_html_dependency()` function adds a `html_dependency` object into a Shiny or R Markdown context. This allows for the direct use of `<i>` tags that refer to Font Awesome icons without having to use the `fa_i()` to create these tags and also add the `html_dependency` to the document.

**Usage**

```
fa_html_dependency()
```

**Details**

The `html_dependency` object is created internally with the following invocation:

```
htmltools::htmlDependency(
  name = "font-awesome",
  version = fa_version,
  src = "fontawesome",
  package = "fontawesome",
  stylesheet = c("css/all.min.css", "css/v4-shims.min.css")
)
```

The `fa_version` object is an internal object that provides the released version number for the Font Awesome icons. This can be inspected by using `fa_metadata()$version`.

**Value**

An `html_dependency` object.

**Examples**

```
if (interactive()) {
  # Create a Font Awesome `html_dependency`
  fa_html_dependency()
}
```

---

fa\_i

*Generate a Font Awesome <i> tag*


---

**Description**

The `fa_i()` function creates a Font Awesome `<i>` tag and not an SVG as with `fa()`. The primary use case for `fa_i()` is for legacy Shiny applications that use the `shiny::icon()` function. This function is called within a `shiny::icon()` call and all HTML dependencies to support icon generation are hosted in the **fontawesome** package.

**Usage**

```
fa_i(name, class = NULL, ..., html_dependency = NULL, verify_fa = TRUE)
```

**Arguments**

name	The name of the Font Awesome icon. This could be as a short name (e.g., "npm", "drum", etc.), or, a full name (e.g., "fab fa-npm", "fas fa-drum", etc.). The names should correspond to current Version 5 Font Awesome names. A list of short and full names can be accessed through the <code>fa_metadata()</code> function with <code>fa_metadata()\$icon_names</code> and <code>fa_metadata()\$icon_names_full</code> . If supplying a Version 4 icon name, it will be internally translated to the Version 5 icon name and a Version 5 icon will be returned. A data frame containing the short names that changed from version 4 ( <code>v4_name</code> ) to version 5 ( <code>v5_name</code> ) can be obtained by using <code>fa_metadata()\$v4_v5_name_tbl</code> .
class	Additional classes to customize the style of the icon (see the usage examples for details on supported styles).
...	Arguments passed to the <code>&lt;i&gt;</code> tag of <code>htmltools::tags</code> .
html_dependency	Provides an opportunity to use a custom <code>html_dependency</code> object (created via a call to <code>htmltools::htmlDependency()</code> ) instead of one supplied by the function (which uses Font Awesome's free assets and are bundled in the package). A custom <code>html_dependency</code> object is useful when you have paid icons from Font Awesome or would otherwise like to customize exactly which icon assets are used (e.g., <code>woff</code> , <code>woff2</code> , <code>eot</code> , etc.). By default, this is <code>NULL</code> where the function internally generates an <code>html_dependency</code> .
verify_fa	An option to verify the provided icon name. If <code>TRUE</code> (the default), internal checks will take place and issue messages should the name is a Font Awesome 4 icon name (the message will provide the Version 5 name), or, if the icon name cannot be found in either Font Awesome 4 or 5.

**Value**

An icon element.

**Examples**

```
if (interactive()) {
  # Create a Font Awesome icon object
  fa_i(name = "r-project")
}
```

## Description

This function provide some metadata about the included Font Awesome assets in the **fontawesome** package. The list that is returned has the following components:

- `version`: The released version number for the Font Awesome icons
- `icon_count`: The total count of unique Font Awesome icons
- `icon_names`: A vector of short names (e.g., "npm", "drum", etc.) for all included icons
- `icon_names_full`: A vector containing the full names (e.g., "fab fa-npm", "fas fa-drum", etc.) for all included icons
- `icon_names_fa(r|s|b)`: Vectors of short names within the regular ("r"), solid ("s"), and brand ("b") groups
- `icon_names_full_fa(r|s|b)`: Vectors with the full names of icons within the regular ("r"), solid ("s"), and brand ("b") groups
- `v4_v5_name_tbl`: A data frame containing the short names that changed from version 4 (`v4_name`) to version 5 (`v5_name`) of Font Awesome

## Usage

```
fa_metadata()
```

## Value

A list with metadata for the included Font Awesome assets.

## Examples

```
if (interactive()) {

  # Get information on the Font Awesome
  # assets included in this package
  fa_metadata()

}
```

---

fa\_png

*Create a PNG version of a Font Awesome icon*

---

## Description

Get a Font Awesome icon as a PNG file. We can optionally set the fill attribute before writing the PNG. Additionally, there is control over the output width and height (usually, icons are 512 by 512 pixels). Please note that this function requires that the **rsvg** is installed on the system. Attempting to use `fa_png()` without **rsvg** available will result in an error message.

**Usage**

```
fa_png(  
  name,  
  file = NULL,  
  fill = NULL,  
  fill_opacity = NULL,  
  stroke = NULL,  
  stroke_width = NULL,  
  stroke_opacity = NULL,  
  height = NULL,  
  width = NULL  
)
```

**Arguments**

name	The name of the Font Awesome icon.
file	the path to the output file. If NULL, then filename will take the short name of the icon and a .png extension will be applied.
fill, fill_opacity	The fill color of the icon can be set with fill. If not provided then the default fill color will be black. The opacity level of the fill color can be controlled with a decimal value between 0 and 1.
stroke, stroke_width, stroke_opacity	The stroke options allow for setting the color, width, and opacity of the outline stroke. By default, the stroke width is very small at "1px" so a size adjustment with "stroke_width" can be useful. The "stroke_opacity" value can be any decimal values between 0 and 1 (bounds included).
height, width	The output height and width of the rendered PNG. If nothing is provided then the output dimensions will match that of the input SVG viewBox.

**Value**

A PNG file written to disk.

**Examples**

```
if (interactive()) {  
  
  # Create a Font Awesome SVG icon as a  
  # PNG file on disk  
  fa_png(name = "r-project")  
  
}
```

# Index

fa, [2](#)  
fa(), [4](#)  
fa\_html\_dependency, [3](#)  
fa\_i, [4](#)  
fa\_i(), [3](#)  
fa\_metadata, [5](#)  
fa\_metadata(), [2](#), [5](#)  
fa\_png, [6](#)  
  
htmltools::htmlDependency(), [5](#)  
htmltools::tags, [5](#)