Package: fontawesome (via r-universe)

August 30, 2024

Type Package

Version 0.5.2.9000

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, https://rstudio.github.io/fontawesome/

BugReports https://github.com/rstudio/fontawesome/issues

Encoding UTF-8

ByteCompile true

RoxygenNote 7.2.3

Depends R (>= 3.3.0)

Imports rlang (>= 1.0.6), htmltools (>= 0.5.1.1)

Suggests covr, dplyr (>= 1.0.8), gt (>= 0.9.0), knitr (>= 1.31), testthat (>= 3.0.0), rsvg

Roxygen list(markdown = TRUE)

Config/testthat/edition 3

Repository https://rstudio.r-universe.dev

RemoteUrl https://github.com/rstudio/fontawesome

RemoteRef HEAD

RemoteSha e0a38c2928156a7a7fae30085ce2d8c6d7c4f0a6

Contents

fa 2
fa_html_dependency 4
fa_i 5
fa_metadata 6
fa_png 7

Index **9**

`fa` *Generate Font Awesome icons as SVGs*

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_left = NULL,
  margin_right = NULL,
  vertical_align = NULL,
  position = NULL,
  title = NULL,
  prefer_type = c("regular", "solid"),
  a11y = c("deco", "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 6 Font Awesome names. A list of short and full names can be accessed through the `fa_metadata()` function with `fa_metadata()$icon_names` and `fa_metadata()$icon_names_full`. If supplying a previous name associated with the icon, it will be internally translated to the current name and a Version 6 icon will be returned.

`fill`, `fill_opacity`
: The fill color of the icon can be set with `fill`. If not provided then the default value of `"currentColor"` is applied so that the SVG fill matches the color of the parent HTML element's `color` 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 `"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 height and width style attributes of the rendered SVG. If nothing is provided for `height` then a default value of `"1em"` will be applied. If a `width` isn't given, then it will be calculated in units of `"em"` on the basis of the icon's SVG `"viewBox"` dimensions.

`margin_left`, `margin_right`
: The length value for the margin that's either left or right of the icon. By default, `"auto"` is used for both properties. If space is needed on either side then a length of `"0.2em"` is recommended as a starting point.

`vertical_align`
: The vertical alignment of the icon. By default, a length of `"-0.125em"` is used.

`position`
: The value for the `position` style attribute. By default, `"relative"` is used here.

`title`
: An option for populating the SVG `'title'` attribute, which provides on-hover text for the icon. By default, no title text is given to the icon. If `a11y == "semantic"` then title text will be automatically given to the rendered icon, however, providing text here will override that.

`prefer_type`
: Chooses the type of icon returned if: (1) providing a short name, and (2) that icon has both solid and regular types. For example, using `name = "address-book"` will result in two types of icons for an Address Book. By default, this preference is set to `"regular"` and the other option is `"solid"`.

`a11y`
: Cases that distinguish the role of the icon and inform which accessibility attributes to be used. Icons can either be `"deco"` (decorative, the default case) or `"sem"` (semantic). Using `"none"` 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,
  ...,
  prefer_type = c("regular", "solid"),
  html_dependency = fa_html_dependency()
)
```

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 Font Awesome names. A list of short and full names can be accessed through the fa_metadata() function with fa_metadata()$icon_names and fa_metadata()$icon_names_full. If supplying a known alias to a short icon name (e.g., "vcard", which is now "address-card"), it will be internally translated to the current icon name before returning the icon tag.
class	Additional classes to customize the style of the icon.
...	Arguments passed to the <i> tag of htmltools::tags.
prefer_type	Chooses the type of icon returned if: (1) providing a short name, and (2) that icon has both solid and regular types. For example, using name = "address-book" will result in two types of icons for an Address Book. By default, this preference is set to "regular" and the other option is "solid".
html_dependency	Provides an opportunity to use a custom html_dependency object (created via a call to htmltools::htmlDependency()) instead of one supplied by the function (which uses Font Awesome's free assets and are bundled in the package). A custom html_dependency 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., woff, woff2, eot, etc.). By default, this is NULL where the function internally generates an html_dependency.

Value

An icon element.

Examples

```
if (interactive()) {

# Create a Font Awesome icon object
fa_i(name = "r-project")

}
```

`fa_metadata` *Get metadata on the included Font Awesome assets*

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

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,
  prefer_type = c("regular", "solid")
)
```

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.
`prefer_type`	Chooses the type of icon returned if: (1) providing a short name, and (2) that icon has both solid and regular types. For example, using `name = "address-book"` will result in two types of icons for an Address Book. By default, this preference is set to `"regular"` and the other option is `"solid"`.

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(), *5*
fa_html_dependency, 4
fa_i, 5
fa_i(), *4*
fa_metadata, 6
fa_metadata(), *2*, *5*
fa_png, 7

htmltools::htmlDependency(), *5*
htmltools::tags, *5*