

# Package ‘favawesome’

May 8, 2026

**Title** 'Font Awesome' Icons as 'shiny' 'favicons'

**Version** 0.1.1

**Description** Easily use 'Font Awesome' icons as 'shiny' 'favicons' (the icons that appear on browser tabs). 'Font Awesome' (<<https://fontawesome.com/>>) is a popular set of icons that can be used in web pages. 'favawesome' provides a simple way to use these icons as 'favicons' in 'shiny' applications and other HTML pages.

**License** MIT + file LICENSE

**URL** <https://favawesome.shinyworks.org>,  
<https://github.com/shinyworks/favawesome>

**BugReports** <https://github.com/shinyworks/favawesome/issues>

**Imports** fontawesome, htmltools, jsonlite, rlang, rsvg

**Suggests** cli, testthat (>= 3.0.0), usethis

**Config/testthat/edition** 3

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Jon Harmon [aut, cre, cph] (ORCID:  
<<https://orcid.org/0000-0003-4781-4346>>)

**Maintainer** Jon Harmon <jonthegeek@gmail.com>

**Repository** CRAN

**Date/Publication** 2024-11-01 14:30:15 UTC

## Contents

fav . . . . .	2
use_favawesome_standalone . . . . .	3
<b>Index</b>	<b>5</b>

---

 fav

*Use Font Awesome icons as favicons*


---

## Description

Generate the html necessary to use a Font Awesome icon as the favicon (the icon that appears on browser tabs) for a shiny app or other HTML document.

## Usage

```
fav(name, ...)
```

## 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 <code>fa_metadata()</code> function with <code>fa_metadata()\$icon_names</code> and <code>fa_metadata()\$icon_names_full</code> . 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.
...	Arguments passed on to <code>fontawesome::fa</code>
fill, fill_opacity	The fill color of the icon can be set with <code>fill</code> . 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 `shiny.tag` (see `htmltools::tag()`) that can be used to embed a favicon in a shiny app or other HTML document.

## Examples

```
html_page <- htmltools::tags$html(  
  fav("earth-africa", fill = "blue"),  
  htmltools::tags$body(  
    htmltools::tags$h1("Hello world!"),  
    htmltools::tags$p("(on the browser tab)")  
  )  
)  
htmltools::html_print(html_page, viewer = utils::browseURL)
```

---

use\_favawesome\_standalone

*Add favawesome\_icon Function to a Project*

---

## Description

If you can't import favawesome in your project, you can still use a Font Awesome (<https://fontawesome.com>) icon as the favicon (the icon that appears on browser tabs). Calling this function will download the necessary code to add a `favawesome_icon()` function to your project, and will overwrite the existing file if it already exists, making it easy to update the code. The `favawesome_icon()` function takes the same arguments as `fav()`.

## Usage

```
use_favawesome_standalone()
```

**Value**

NULL invisibly (called for side effect of importing the `favawesome_icon()` definition). `favawesome_icon()` returns the same value as `fav()`.

# Index

`fa_metadata()`, [2](#)

`fav`, [2](#)

`fav()`, [3](#), [4](#)

`fontawesome::fa`, [2](#)

`htmltools::tag()`, [3](#)

`use_favawesome_standalone`, [3](#)