

Package ‘icons’

September 3, 2026

Type Package

Title Easily Insert SVG Icons into R Outputs

Version 1.0.0

Description Provides tools to download, style, and insert scalable vector graphics (SVG) icons into R output, such as 'R Markdown' and 'Quarto' documents, 'Shiny' apps, 'gt' tables, and 'ggplot2' visualisations via the 'ggicons' package. Bundles popular icon libraries, such as 'Font Awesome' (<<https://fontawesome.com>>), 'Academicons' (<<https://jpswalsh.github.io/academicons/>>), and 'Material Design Icons' (<<https://fonts.google.com/icons>>), which are downloaded on demand to keep the package small, or build a custom icon set from any folder of local SVG files. Icon appearance, such as size, colour, and position, can be customised.

Depends R (>= 3.1.3)

Imports base64enc, knitr, rlang, htmltools, glue, rappdirs, xml2, stringr, jsonlite, cli, utils, vctrs

Suggests rsvg, gh, rmarkdown, pillar, shiny, testthat (>= 3.0.0)

URL <https://pkg.mitchelloharawild.com/icons/>,
<https://github.com/mitchelloharawild/icons>

BugReports <https://github.com/mitchelloharawild/icons/issues>

VignetteBuilder knitr

License MIT + file LICENSE

Encoding UTF-8

Config/roxygen2/version 8.0.0

Config/testthat/edition 3

NeedsCompilation no

Author Mitchell O'Hara-Wild [aut, cre] (ORCID: <<https://orcid.org/0000-0001-6729-7695>>),
Earo Wang [ctb] (ORCID: <<https://orcid.org/0000-0001-6448-5260>>),
Timothy Hyndman [ctb] (ORCID: <<https://orcid.org/0000-0002-7047-5023>>),

Matthew Warkentin [ctb] (ORCID:
<<https://orcid.org/0000-0001-8730-3511>>),
Tom Jemmett [ctb] (ORCID: <<https://orcid.org/0000-0002-6943-2990>>)

Maintainer Mitchell O'Hara-Wild <mail@mitchelloharawild.com>

Repository CRAN

Date/Publication 2026-09-03 12:10:14 UTC

Contents

download_academicons	2
download_bioicons	3
download_feather_icons	4
download_fontawesome	5
download_google_material	5
download_health_icons	6
download_ionicons	7
download_octicons	8
download_simple_icons	8
download_super_tiny_icons	9
icon_find	10
icon_installed	11
icon_label	11
icon_path	12
icon_save	13
icon_set	14
icon_style	14
icon_uri	15
read_icon	16

Index	17
--------------	-----------

download_academicons	<i>Academicons icons</i>
----------------------	--------------------------

Description

Academicons is a specialist icon font for academics. It contains icons for websites and organisations related to academia that are often missing from mainstream font packages. It can be used by itself, but its primary purpose is to be used as a supplementary package alongside a larger icon set.

Usage

```
download_academicons(version = "dev")

academicons(name)
```

Arguments

version	Version of the library
name	Name of the icon

Value

download_academicons() invisibly returns the academicons icon set after downloading and installing its SVG files locally.

academicons(name) returns an icons vector (an SVG tag) for the requested icon.

License

Academicons is distributed under the [SIL Open Font License 1.1](#) (see the [project README](#) for details).

download_bioicons	<i>Bioicons icons</i>
-------------------	-----------------------

Description

Bioicons is a free library of open source icons for scientific illustrations using vector graphics software such as Inkscape or Adobe Illustrator.

Usage

```
download_bioicons()
```

```
bioicons(name, category = NULL)
```

Arguments

name	Name of the icon.
category	The icon's category, one of "Animals", "Blood_immunology", "Cell_lines", "Cell_membrane", "Chemistry", "Chemo-_and_Bioinformatics", "General_items", "Genetics", "Human_physiology", "Intracellular_components", "Lab_apparatus", "Machine_Learning", "Microbiology", "Nucleic_Acids", "Oncology", "Parasites", "Receptor_channels", "Safety_symbols", "Scientific_graphs", "Tissues", or "Viruses". If NULL, it will default to the first available icon with this name.

Value

download_bioicons() invisibly returns the bioicons icon set after downloading and installing its SVG files locally.

bioicons(name, category) returns an icons vector (an SVG tag) for the requested icon.

License

Each Bioicons icon is contributed under a license chosen by its author - commonly CC-0, MIT, CC-BY-SA, or BSD. The license is baked into the icon's identifier (use `icon_label()` to recover it, e.g. "bioicons\$CC-0\$General_items\$AuthorName\$name"), and is also listed alongside each icon at <https://bioicons.com/>. Check, and where required credit, the individual icon's license before reuse.

See Also

<https://bioicons.com/>

download_feather_icons

Feather Icons

Description

Feather Icons

Usage

```
download_feather_icons(version = "dev")
```

```
feather_icons(name)
```

Arguments

version	Version of the library
name	Name of the icon

Value

`download_feather_icons()` invisibly returns the `feather_icons` icon set after downloading and installing its SVG files locally.

`feather_icons(name)` returns an icons vector (an SVG tag) for the requested icon.

License

Feather Icons are distributed under the [MIT License](#).

download_fontawesome *Font Awesome icons*

Description

Font Awesome icons

Usage

```
download_fontawesome(version = "dev")
```

```
fontawesome(name, style = NULL)
```

Arguments

version	Version of the library
name	Name of the icon
style	Style of the icon. If NULL, a default style will be chosen for the specified icon.

Value

download_fontawesome() invisibly returns the fontawesome icon set after downloading and installing its SVG files locally.

fontawesome(name, style) returns an icons vector (an SVG tag) for the requested icon.

License

This package uses Font Awesome Free. Its icons are distributed under the [CC BY 4.0 License](#), which requires attribution when the icons are used.

download_google_material
Google's Material design icons

Description

Google's Material design icons

Usage

```
download_google_material(version = "dev")
```

```
google_material(name, category = NULL, theme = NULL)
```

Arguments

version	Version of the library
name	Name of the icon
category	The icon's category, either "action", "alert", "av", "communication", "content", "device", "editor", "file", "hardware", "home", "image", "maps", "navigation", "notification", "places", "social", or "toggle"
theme	The style variant for the icon, requires v4.0.0 or greater. Available themes vary by icon, but can be either "filled", "outlined", "round", "sharp", or "twotone". If NULL, it will default to the first available variant in this order.

Value

download_google_material() invisibly returns the google_material icon set after downloading and installing its SVG files locally.

google_material(name, category, theme) returns an icons vector (an SVG tag) for the requested icon.

License

Material Design Icons are distributed under the [Apache License 2.0](#).

See Also

<https://fonts.google.com/icons>

download_health_icons *Health icons*

Description

Health icons

Usage

```
download_health_icons()
```

```
health_icons(name, category = NULL, theme = NULL)
```

Arguments

name	Name of the icon
category	The icon's category, either "blood", "body", "conditions", "devices", "diagnostics", "emotions", "family", "medications", "objects", "people", "places", "shapes", "specialties", "symbols", "typography", or "vehicles"
theme	The style variant for the icon, requires v4.0.0 or greater. Available themes vary by icon, but can be either "filled", "negative", "outline". If NULL, it will default to the first available variant in this order.

Value

download_health_icons() invisibly returns the health_icons icon set after downloading and installing its SVG files locally.

health_icons(name, category, theme) returns an icons vector (an SVG tag) for the requested icon.

License

Health Icons are released into the public domain under the [CC0 1.0 Universal License](#); no attribution is required. The Health Icons codebase itself is licensed separately under the [MIT License](#).

download_ionicons	<i>Ionicons icons</i>
-------------------	-----------------------

Description

Ionicons icons

Usage

```
download_ionicons(version = "dev")
```

```
ionicons(name)
```

Arguments

version	Version of the library
name	Name of the icon

Value

download_ionicons() invisibly returns the ionicons icon set after downloading and installing its SVG files locally.

ionicons(name) returns an icons vector (an SVG tag) for the requested icon.

License

Ionicons are distributed under the [MIT License](#).

download_octicons	<i>GitHub's Octicons</i>
-------------------	--------------------------

Description

GitHub's Octicons

Usage

```
download_octicons(version = "dev")
```

```
octicons(name)
```

Arguments

version	Version of the library
name	Name of the icon

Value

`download_octicons()` invisibly returns the octicons icon set after downloading and installing its SVG files locally.

`octicons(name)` returns an icons vector (an SVG tag) for the requested icon.

License

Octicons are distributed under the **MIT License**.

See Also

<https://primer.style/octicons/>

download_simple_icons	<i>Simple Icons</i>
-----------------------	---------------------

Description

Simple Icons

Usage

```
download_simple_icons(version = "dev")
```

```
simple_icons(name)
```

Arguments

version	Version of the library
name	Name of the icon

Value

download_simple_icons() invisibly returns the simple_icons icon set after downloading and installing its SVG files locally.

simple_icons(name) returns an icons vector (an SVG tag) for the requested icon.

License

The Simple Icons artwork is distributed under the [CC0 1.0 Universal License](#). Many of the icons depict third-party brands and logos which may be protected as trademarks; read Simple Icons' [legal disclaimer](#) before using them.

download_super_tiny_icons
Super Tiny Icons

Description

Super Tiny Icons are miniscule SVG versions of popular website and app logos, each under 1KB in size.

Usage

```
download_super_tiny_icons(version = "dev")
```

```
super_tiny_icons(name)
```

Arguments

version	Version of the library
name	Name of the icon

Value

download_super_tiny_icons() invisibly returns the super_tiny_icons icon set after downloading and installing its SVG files locally.

super_tiny_icons(name) returns an icons vector (an SVG tag) for the requested icon.

License

The project's own icons are dual-licensed by the author under the [MIT License](#) and [CC0 1.0 Universal License](#). Many icons are adapted from third-party logos that carry their own licenses (and some depict trademarks), so check the source noted for an icon before reusing it - see the project's [licenses list](#) for details.

See Also

<https://github.com/edent/SuperTinyIcons>

icon_find

Find icons in installed sets by name

Description

Find icons in installed sets by name

Usage

```
icon_find(name, set = NULL)
```

Arguments

name	The name of the icon
set	Icon sets to search. If NULL, all available icons will be searched.

Value

An icons vector of matching icons (empty if none are found). Use `icon_label()` to recover each match's `library$sub$name` accessor expression (for example `"fontawesome$solid$rocket"`) - the result itself carries no `names()`, since an icons vector can't hold them.

Examples

```
# Empty if no installed icon set has a matching icon
icon_find("rocket")

if(icon_installed(fontawesome)){

  found <- icon_find("rocket")
  icon_label(found)

}
```

icon_installed	<i>Check if an icon set is installed.</i>
----------------	---

Description

This function will return TRUE if the icons for an icon set are installed. If they aren't, they can be installed using the appropriate `download_*`() function.

Usage

```
icon_installed(x)
```

Arguments

x An icon set (such as `fontawesome`).

Value

A single TRUE or FALSE.

Examples

```
icon_installed(fontawesome)
```

icon_label	<i>Identify an icon by its library accessor label</i>
------------	---

Description

Every icon sourced from an installed icon library (as opposed to a local `icon_set()` or a raw `read_icon()` file) has a file path of the form `<cache root>/<library>/<...name parts...>.svg` (see `00_registry.R`'s `icon_fn$get()`), which is exactly the `library$sub$name` expression used to access it. `icon_label()` recovers that label straight from `icon_path()` rather than requiring a name to be tracked separately — so, unlike `names()`, it survives `c()/rep()/subsetting` on an icons vector (a `vctrs` `rcrd`, which cannot carry element names()).

Usage

```
icon_label(x)
```

Arguments

x An icons vector.

Value

A character vector the same length as `x` giving each icon's `library$sub$name` label. Icons not sourced from an installed library (a local `icon_set()` or a `read_icon()` file outside the icon cache) return NA.

Examples

```
path <- tempfile(fileext = ".svg")
writeLines(
  '<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"><path d="M0 0"/></svg>',
  path
)
icon_label(read_icon(path)) # NA - not sourced from an installed library

if(icon_installed(fontawesome)){
  icon_label(fontawesome("rocket", style = "solid"))
}
```

icon_path

Get the file path of an icon

Description

Icons created from an SVG file (such as those from an `icon_set()`, or read directly with `read_icon()`) retain the path to their source file. This is useful for passing icons to other packages that work with SVG files directly, such as `ggplot2` (via `grid::rasterGrob()` or similar) or `gt`.

Usage

```
icon_path(x)
```

Arguments

`x` An icons vector.

Value

A character vector giving the path(s) to the icon's source SVG file(s) on disk.

Examples

```
path <- tempfile(fileext = ".svg")
writeLines(
  '<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"><path d="M0 0"/></svg>',
  path
)
```

```
icon_path(read_icon(path))
```

icon_save	<i>Save icons into a local path</i>
-----------	-------------------------------------

Description

Store icons in a local directory to simplify sharing of documents containing icons. Bundling icons with your document or app avoids the need for collaborators or servers to have all icons from a library installed.

Usage

```
icon_save-icons, path = ".")
```

Arguments

icons	A named list of icons, the names specify the icon's name in the new icon set.
path	A directory to save the icons into.

Value

Invisibly, the icon_set created from the newly saved SVG files at path.

Examples

```
if(icon_installed(fontawesome) && icon_installed(ionicons)){

  path <- tempfile()
  dir.create(path)
  icon_save(
    list(arrow = fontawesome$solid$`arrow-right`, alarm = ionicons$alarm),
    path = path
  )

  app_icons <- icon_set(path)
  app_icons$arrow

}
```

icon_set	Create a custom icon set
----------	--------------------------

Description

Create a custom icon set

Usage

```
icon_set(path, meta = list(name = "Custom", version = NULL, license = NULL))
```

Arguments

path	Path to the icons
meta	Meta information for the icons

Value

An icon_set object providing access to the SVG files found at path, for example via \$.

Examples

```
dir <- tempfile()
dir.create(dir)
writeLines(
  '<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"><path d="M0 0"/></svg>',
  file.path(dir, "example.svg")
)

custom <- icon_set(dir)
custom$example
```

icon_style	Customise the style of an icon
------------	--------------------------------

Description

Customise the style of an icon

Usage

```
icon_style(x, scale = NULL, fill = NULL, rotate = NULL, ...)
```

Arguments

x	The icon to style.
scale	Scaled size of the icon.
fill	The colour for the fill.
rotate	The angle to rotate the icon.
...	Other CSS rules for the icon style, for example float = "right".

Value

The input x (an icons vector), with its style updated to reflect the requested styling.

Examples

```
path <- tempfile(fileext = ".svg")
writeLines(
  '<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"><path d="M0 0"/></svg>',
  path
)
icon <- read_icon(path)
icon_style(icon, scale = 2, fill = "red", rotate = 90)
```

icon_uri

Encode a (styled) icon as a self-contained data: URI

Description

Renders an icon's SVG markup to a base64-encoded URI.

This enables self-contained and styled icon for consumers that use image paths/URLs, including for example: an `<img src>`, a `gt::html()` cell, and a leaflet marker `iconUrl`.

Usage

```
icon_uri(x)
```

Arguments

x	An icons vector.
---	------------------

Value

A character vector giving each icon's base64 encoded URI.

Examples

```
path <- tempfile(fileext = ".svg")
writeLines(
  '<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"><path d="M0 0"/></svg>',
  path
)
icon_uri(read_icon(path))
```

read_icon	<i>Read an individual icon</i>
-----------	--------------------------------

Description

Read an individual icon

Usage

```
read_icon(x)
```

Arguments

x	Path to the icon
---	------------------

Value

An icons vector: an SVG icon that can be printed, embedded in R Markdown, or styled with [icon_style\(\)](#). If x is a file path, it is retained and can be retrieved with [icon_path\(\)](#).

Examples

```
path <- tempfile(fileext = ".svg")
writeLines(
  '<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"><path d="M0 0"/></svg>',
  path
)
read_icon(path)
```

Index

academicons (download_academicons), 2

bioicons (download_bioicons), 3

download_academicons, 2

download_bioicons, 3

download_feather_icons, 4

download_fontawesome, 5

download_google_material, 5

download_health_icons, 6

download_ionicons, 7

download_octicons, 8

download_simple_icons, 8

download_super_tiny_icons, 9

feather_icons (download_feather_icons),
4

fontawesome, 11

fontawesome (download_fontawesome), 5

google_material
(download_google_material), 5

health_icons (download_health_icons), 6

icon_find, 10

icon_installed, 11

icon_label, 11

icon_label(), 4, 10

icon_path, 12

icon_path(), 11, 16

icon_save, 13

icon_set, 14

icon_set(), 11, 12

icon_style, 14

icon_style(), 16

icon_uri, 15

ionicons (download_ionicons), 7

octicons (download_octicons), 8

read_icon, 16

read_icon(), 11, 12

simple_icons (download_simple_icons), 8

super_tiny_icons
(download_super_tiny_icons), 9