functions

- SVG framework functions
  - Usage
  - Functions
  - General functions {#general}
  - Getting icons {#getting-icons}
  - Adding icons {#adding-icons}
  - Rendering icons {#render}
  - Scanning and observing DOM {#scanner}
  - Helper functions {#helper}
  - API functions {#api}
  - Internal API functions {#internal}


```
title: Iconify SVG Framework Functions
replacements:
  - code: '"2.0.0"'
    value: '"${svg-framework.version.full}"'
  - code: "@iconify/iconify@2"
    value: "${svg-framework.import}"
types:
  IconifyJSON: "../../types/iconify-json.md"
  IconifyIcon: "../../types/iconify-icon.md"
  IconifyInfo: "../../types/iconify-json-metadata.md"
functions:
  addCollection: "./add-collection.md"
  addIcon: "./add-icon.md"
  iconExists: "./icon-exists.md"
  iconLoaded: "./icon-exists.md"
  listIcons: "./list-icons.md"
  getIcon: "./get-icon.md"
  renderSVG: "./render-svg.md"
```

```
  renderHTML: "./render-html.md"
  renderIcon: "./render-icon.md"
  replaceIDs: "./replace-ids.md"
  scan: "./scan.md"
  observe: "./observe.md"
  stopObserving: "./stop-observing.md"
  pauseObserver: "./pause-observer.md"
  resumeObserver: "./resume-observer.md"
  loadIcons: "./load-icons.md"
  loadIcon: "./load-icon.md"
  enableCache: "./enable-cache.md"
  disableCache: "./disable-cache.md"
  addAPIProvider: "./add-api-provider.md"
  buildIcon: "./build-icon.md"
  fetFetch: "./set-fetch.md"
```

SVG framework functions

This tutorial is part of Iconify SVG Framework tutorial.

Iconify SVG framework has many functions that you can use to use the SVG framework in your scripts.

Usage

There are two ways of using SVG framework functions:

- By using `[var]Iconify` global.
- By importing `[var]Iconify` (or named functions you want to import) from `[npm]@iconify/iconify@2` if you are bundling SVG framework with your scripts.

Examples of using `[func]loadIcon` function:

```
src: icon-components/iconify/load-icon-global.html
title: "Browser:"
replacements:
  - code: "/2/2.0.0/"
    value: "/${svg-framework.version.major}/${svg-framework.version.full}/"
```

```
extra:
  - src: icon-components/iconify/load-icon-promise.js
    title: "Node.js:"
```

Even if you use bundle (method shown in second example), `[var]Iconify` global is also available because SVG framework exports functions and creates a global variable regardless of how you use it. That means you can use method shown in first example regardless of how you import SVG framework.

Functions

Functions are split in several groups:

- General functions.
- Getting icons.
- Adding icons.
- Rendering icons.
- Scanning and observing DOM.
- API functions.
- Internal API functions.

Click function name to see more details and examples.

General functions {#general}

In this section there is only one function:

- `[func]getVersion()`. This function returns SVG framework version string. `[str]"2.0.0"`

Getting icons {#getting-icons}

```
include: icon-components/components/functions-list/getting-icons
```

Adding icons {#adding-icons}

```
include: icon-components/components/functions-list/adding-icons
replacements:
  - search: "component"
    replace: "SVG framework"
```

Rendering icons {#render}

Functions that generate SVG or data:

- `[func]renderSVG(name, customisations?)`. Creates `[tag]svg` element.
- `[func]renderHTML(name, customisations?)`. Returns `[tag]svg` string.
- `[func]renderIcon(name, customisations?)`. Generates data used by functions above. This can be used if you prefer to generate `[tag]svg` yourself. Data includes attributes for `[tag]svg` and inner HTML.
- `[func]buildIcon(data, customisations)` is identical to `[func]renderIcon()`, but uses icon data as first parameter

Scanning and observing DOM {#scanner}

SVG framework automatically scans DOM whenever something changes. However, there are some limitations:

- Observer can observe only child elements of `[prop]document.body`.
- SVG framework scans DOM after every change (though scans are throttled to avoid scanning too often).

In some instances you might want to temporarily disable observer or scan an element that is not part of DOM, such as Shadow DOM. There are functions that you can use:

- `[func]scan(root?)`. Scans DOM or custom element for placeholder elements.
- `[func]observe(root)`. Observes custom root element.
- `[func]stopObserving(root)`. Stops observing custom root element. You can call it with `[prop]document.body` as parameter to stop observing `[prop]document.body`.
- `[func]pauseObserver(root?)`. Pauses observer.
- `[func]resumeObserver(root?)`. Resumes observer.

Helper functions {#helper}

Few helper functions that are exposed because they might be useful when creating things such as icon picker:

- `[func]calculateSize()`. Helper function to calculates icon size. It is used to calculate `[attr]width` if only `[attr]height` is set and vice versa.
- `[func]replaceIDs(html)`. Randomizes IDs in generated string. This should be used when rendering icon based on data returned by `[func]renderIcon()` or `[func]getIcon` to make sure elements inside each icon have unique IDs. This function is not needed for icons generated by `[func]renderSVG()` and `[func]renderHTML()`.

API functions {#api}

```
include: icon-components/components/functions-list/api
```

Internal API functions {#internal}

`include icon-components/components/functions-list/internal-intro`

All internal API functions are exposed as properties of `[var]Iconify._api` object and are available only when API is included:

- `[func]getAPI()`. Returns internal API module.
- `[func]getAPIConfig()`. Returns API configuration.
- `[func]setAPIModule(provider)`. Sets API module for provider. This is experimental function intended for custom API providers. API provider functionality is in development.

□□□ #2
□ □□□ □□□ 2026-10-31 14:20:15 UTC
□ □□□ □□□ 2026-10-31 14:20:16 UTC