index

- Iconify for Tailwind CSS
  - Tailwind CSS version
  - Plugins
    - Installation
      - Tailwind 3 configuration
      - Tailwind 4 config
    - Icon sets
      - Custom icon sets
  - Usage


```
title: Iconify for Tailwind CSS version 3
replacements:
  - code: "60,000"
    value: "${counters.icons}"
  - code: "60k"
    value: "${counters.icons-short}"
functions:
  addDynamicIconSelectors: "../dynamic/index.md"
  addIconSelectors: "../iconify/index.md"
```

Iconify for Tailwind CSS

Iconify plugin for Tailwind CSS makes it easy to use icons in Tailwind CSS.

You can use over 60,000 open source icons and custom icons with minimal code.

Tailwind CSS version

This documentation covers plugins for Tailwind 3.

For a newer Tailwind 4 plugin, see Tailwind 4 plugin documentation.

However, Tailwind 3 plugins can be used with Tailwind 4 too, but you need to create a configuration file.

Plugins

There are several plugins available, they use different syntax, have different options.

There are 2 main plugins:

- `[func]addDynamicIconSelectors()`
- `[func]addIconSelectors()`

Plugins use different syntax, have different options.

For example, here is syntax to use icon `[icon]mdi-light:home` in HTML:

```
src: usage/tailwind/syntax-iconify.html
hint: "Usage with addDynamicIconSelectors"
```

```
src: usage/tailwind/syntax-iconify2.html
hint: "Usage with addIconSelectors"
```

Installation

To install plugins, add `[npm]@iconify/tailwind` as dev dependency:

```
npm i -D @iconify/tailwind
```

Then you need to configure it.

Tailwind 3 configuration

For Tailwind CSS 3, open `[file]tailwind.config.js`, import `[func]addIconSelectors` (main plugin) or `[func]addDynamicIconSelectors` (dynamic selectors) from `[npm]@iconify/tailwind` and add it to a list of plugins.

Example `[file]tailwind.config.js` with `[func]addIconSelectors()` plugin:

```
const { addIconSelectors } = require("@iconify/tailwind");

/** @type {import('tailwindcss').Config} */
module.exports = {
  content: ["./src/*.html"],
  plugins: [
    // Iconify plugin for clean selectors, requires writing a list of icon sets to load
    // Icons usage in HTML:
    //  <span class="iconify mdi-light--home"></span>
    //  <span class="iconify-color vscode-icons--file-type-tailwind"></span>
    addIconSelectors(["mdi-light", "vscode-icons"]),
  ],
};
```

Example `[file]tailwind.config.js` with `[func]addDynamicIconSelectors()` plugin:

```
const { addDynamicIconSelectors } = require("@iconify/tailwind");

/** @type {import('tailwindcss').Config} */
module.exports = {
  content: ["./src/*.html"],
  plugins: [
    // Iconify plugin for dynamic selectors, configuration is not required
    // Icons usage in HTML:
    //  <span class="i-[mdi-light--home]"></span>
    addDynamicIconSelectors(),
  ],
};
```

Tailwind 4 config

For Tailwind 4, you need to create a new file, for example, `[file]icons.mjs` with configuration:

```
import { addDynamicIconSelectors } from "@iconify/tailwind";

export default addDynamicIconSelectors();
```

Then in your CSS file import that plugin:

```
@plugin './icons.mjs';
```

All code samples in documentation are for Tailwind CSS 3.

Tailwind CSS 4 configuration is similar, except that instead of adding plugin to `[prop]plugins` property of Tailwind configuration, you need to export it as a default export.

This requires basic understanding of how modules and exports work.

Icon sets

Plugin does not include icons. You need to add icon sets you want to use.

To add all open source icon sets, add `[npm]@iconify/json` as dev dependency:

```
npm i -D @iconify/json
```

You can also install only icon sets that you want to use by installing `[npm]@iconify-json/{prefix}` dependencies (where `[str]{prefix}` is icon set prefix), such as `[npm]@iconify-json/mdi-light`.

See icon data documentation.

Custom icon sets

Plugins also work with custom icon sets, see detailed documentation below.

Usage

For more details, see documentation for each plugin:

- `[func]addIconSelectors()` plugin documentation.
- `[func]addDynamicIconSelectors()` plugin documentation.

□□□ #2
□ □□□ □□□ 2026-10-31 14:30:32 UTC
□ □□□ □□□ 2026-10-31 14:30:33 UTC