size-color

- Changing icon size and color in Iconify for Tailwind CSS
  - Icon color
    - Monotone icons
    - Icons with palette
  - Icon size
    - Non-square icons
    - Scaling icon
    - Fixed size
    - Multiple sizes


```
title: Changing icon size and color in Iconify for Tailwind CSS
types:
  IconifyJSON: '/docs/types/iconify-json.md'
  IconifyInfo: '/docs/types/iconify-info.md'
functions:
  addIconSelectors: './index.md'
```

Changing icon size and color in Iconify for Tailwind CSS

This documentation explains how to change icon size and color for `[func]addIconSelectors` plugin.

Icon color

There are two types of icons:

- Icons with hardcoded palette.
- Monotone icons.

See how monotone icons work in CSS (html syntax in linked tutorial is different - tutorial is not specific to this plugin).

Monotone icons

To change the color of monotone icons, render them as mask images and change text color:

```
<span class="iconify mdi-light--home text-blue-600"></span>
```

Icons with palette

Icons with hardcoded palette aren't designed to be customisable. However, `[func]addIconSelectors` plugin gives you option to change color via `[prop]customise` option.

For example, this code will change yellow background in Twitter emojis with red:

```
addIconSelectors({
    prefixes: [
        // Render 'twemoji' icon set
        'twemoji',
        // Render 'twemoji' as 'twemoji-red' with custom color changes
        {
            // Change icon set prefix by setting 'prefix' property
            prefix: 'twemoji-red',

            // Resolve location of icon set from '@iconify-json/twemoji' package
            // Also below is the same code, using '@iconify/json' package
            source: require.resolve('@iconify-json/twemoji/icons.json'),
            // source: require.resolve('@iconify/json/json/twemoji.json'),

            // Filter only icons that contain 'face' to improve plugin performance
            icons: (name) => name.includes('face'),

            // Change color
            customise: (content, name) => content.replaceAll('#ffcc4d', '#ff9667'),
        },
```

```
    ],
})
```

Then you can use both original and customised icons:

```
<span class="iconify-color twemoji--winking-face"></span>
<span class="iconify-color twemoji-red--winking-face"></span>
```

See customise option documentation for details and usage examples.

Also see prefixes option documentation for explanation of how the configuration above works.

Icon size

By default, icons are rendered as `[num]1em` x `[num]1em` square.

To change icon size, you can change font size:

```
<span class="iconify mdi-light--home text-2xl"></span>
```

Because common selectors, which contain size, are treated as Tailwind components, you can also set width/height to resize icon:

```
<span class="iconify mdi-light--home w-12 h-12"></span>
```

Make sure width and height are identical (see `[prop]square` option below). Some browsers (Safari) fail to keep proportions when using SVG as a mask image, so using different width and height values might result in unexpected user experience for some users.

Additionally, you can change default size by using `[prop]scale` option.

Non-square icons

By default, all icons are rendered as square icons. This makes it easy to resize icons.

However, if you want to render non-square icons as is, you can disable this behavior by changing option `[prop]square` to `false`:

```
addIconSelectors({
    prefixes: ['fa6-regular'],
```

```
    square: false,
})
```

However, be aware that this might make resizing icon a bit more complex. If you are setting `[prop]width` and `[prop]height`, make sure width/height ratio matches icon's ratio.

Scaling icon

Option `[prop]scale` changes default icon size from `[num]1em`.

The following configuration

```
addIconSelectors({
    prefixes: ['mdi-light'],
    scale: 1.2,
})
```

changes default size to `[num]1.2em` x `[num]1.2em` square.

You can use font size to further resize icons, like in the example above, but icons will be 1.2 times bigger.

Fixed size

If you want to set default icon size to a fixed size in pixels, such as `[num]24` to render icons that are designed for `[num]24` x `[num]24`, you can set width and height in pixels using `[prop]extraMaskRules` or `[prop]extraBackgroundRules` options:

```
addIconSelectors({
    maskSelector: '.material-icon',
    extraMaskRules: {
        'width': '24px',
        'height': '24px',
    },
    backgroundSelector: '.material-icon-color',
    extraBackgroundRules: {
        'width': '24px',
        'height': '24px',
    },
})
```

Multiple sizes

What if you want to use `[num]1em` icons, icons without size and icons with fixed size in pixels in the same project?

You can use multiple instances of plugin with different settings and different selectors:

```
plugins: [
    // Default options, 1em size
    addIconSelectors(['mdi-light']),
    // Fixed size
    addIconSelectors({
        // Set fixed size
        maskSelector: '.material-icon',
        extraMaskRules: {
            'width': '24px',
            'height': '24px',
        },
        // Not using background selector for this demo - demo uses monotone icons only
        backgroundSelector: '',
    }),
]
```

```
<p>
  Icon with 1em size:
  <span class="iconify mdi-light--home"></span>
</p>
<p>
  Using custom width/height:
  <span class="iconify mdi-light--home h-12 w-12"></span>
</p>
<p>
  24px icon:
  <span class="material-icon mdi-light--home"></span>
</p>
```

Because icon data is identical in all instances of plugin, there is no need to change icon selector and the `[prop]prefixes` option needs to be used only once. This reduces duplication and makes parsing faster.

□□□ #2
□ □□□ □□□ 2026-10-31 14:30:27 UTC

□ □□□ □□□ 2026-10-31 14:30:28 UTC