🔍 Search 6.4.0 docs

📖 **Docs Menu**   6.4.0

Rotate Icons

Sometimes you need to rotate, flip, or mirror an icon for it to work in your project or design. We've included some quick utilities to help with that.
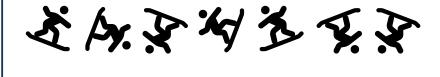
Do More with Font Awesome Pro!

Upgrade now and rev up your productivity with more icons, styles, and tools.

Before You Get Started

Before we start, make sure you're:

- ✅ **Set up with Font Awesome** in your project.
- ✅ Familiar with the basics of **adding Font Awesome icons**.

To arbitrarily rotate and flip icons, use the `fa-rotate-*` and `fa-flip-*` classes when you reference an icon.

```
<div class="fa-3x">
  <i class="fa-solid fa-snowboarding"></i>
  <i class="fa-solid fa-snowboarding fa-rotate-90"></i>
  <i class="fa-solid fa-snowboarding fa-rotate-180"></i>
  <i class="fa-solid fa-snowboarding fa-rotate-270"></i>
  <i class="fa-solid fa-snowboarding fa-flip-horizontal"></i>
  <i class="fa-solid fa-snowboarding fa-flip-vertical"></i>
  <i class="fa-solid fa-snowboarding fa-flip-both"></i>
</div>
```

Rotate Classes

Class	Details
fa-rotate-90	Rotates an icon 90°
fa-rotate-180	Rotates an icon 180°
fa-rotate-270	Rotates an icon 270°
fa-flip-horizontal	Mirrors an icon horizontally
fa-flip-vertical	Mirrors an icon vertically
fa-flip-both	Mirrors an icon both vertically and horizontally
fa-rotate-by	Rotates an icon by a specific degree - setting an accompanying **valid** ↗ inline value for --fa-rotate-angle is required

Custom Rotation

We've added a **CSS custom property** ☑ to make rotating icons more customizable and specific. Add the `.fa-rotate-by` class to an icon and define your own angle.

CSS Custom Property	Details	Accepted Values
`--fa-rotate-angle`	Set rotation angle of `.fa-rotate-by`	Any valid CSS `transform` rotate **value** ☑


```
<div class="fa-3x">
  <i class="fa-solid fa-snowboarding"></i>

  <i class="fa-solid fa-snowboarding fa-rotate-by" style="--fa-rotate-angl
  <i class="fa-solid fa-snowboarding fa-rotate-by" style="--fa-rotate-angl

  <i class="fa-solid fa-snowboarding fa-rotate-by" style="--fa-rotate-angl
  <i class="fa-solid fa-snowboarding fa-rotate-by" style="--fa-rotate-angl

  <i class="fa-solid fa-snowboarding fa-rotate-by" style="--fa-rotate-angl
  <i class="fa-solid fa-snowboarding fa-rotate-by" style="--fa-rotate-angl
</div>
```

Combining Rotating + Flipping

Our rotation utilities leverage CSS's **transform property** ☑ . You can both rotate and flip an icon, but you'll need to use some extra markup, such as a `<span>` element, to do so as placing multiple rotate/flip classes on one element will just overwrite one another. Apply one rotate utility class on the parent element and another on the nested icon.

```html
<div class="fa-3x">
  <!-- A icon that's rotated 90 degress and flipped horizontally -->
  <span class="fa-rotate-90" style="display: inline-block;">
    <i class="fa-solid fa-snowboarding fa-flip-horizontal"></i>
  </span>

  <!-- A icon that's flipped horizontally and rotated 90 degrees  -->
  <span class="fa-flip-horizontal" style="display: inline-block;">
    <i class="fa-solid fa-snowboarding fa-rotate-90"></i>
  </span>

  <!-- A icon that's flipped vertically and rotated 270 degress  -->
  <span class="fa-flip-vertical" style="display: inline-block;">
    <i class="fa-solid fa-snowboarding fa-rotate-270"></i>
  </span>

  <!-- A icon that's rotated 270 degress and flipped vertically -->
  <span class="fa-rotate-270" style="display: inline-block;">
    <i class="fa-solid fa-snowboarding fa-flip-vertical"></i>
  </span>

  <!-- A icon that's flipped on both axes and arbitrarily rotated 120 degr
  <span class="fa-flip-both" style="display: inline-block;">
    <i class="fa-solid fa-snowboarding fa-rotate-by" style="--fa-rotate-ar
  </span>

  <!-- A icon that's arbitrarily rotated 20 degres and flipped on both axe
  <span class="fa-rotate-by" style="display: inline-block; --fa-rotate-ang
    <i class="fa-solid fa-snowboarding fa-flip-both"></i>
  </span>
</div>
```

⚠️ **CSS Transforms don't work with inline elements!**

If you're using an HTML element that is native by default set to `display: inline;`, you'll need to add a custom rule to change that. In the examples above, we use and generally recommend `display: inline-block`.

Need more control when rotating?

Our **SVG + JS method** includes a **power transforms feature** that allows for rotating an icon on a more granular scale, even by a single degree! If you're using Font Awesome that way, check it out.