

Getting started

Install

`npm` `pnpm` `yarn` `bun`

```
npm install --save-dev mistcss
```

shell

Usage

MistCSS is better understood with an example.

Let's create a classic `Button` component accepting two props:

- `size: 'lg' | 'sm'`
- `danger: boolean`

`src/Button.mist.css`

`CSS` `CSS + Tailwind`

```
@scope (.button) {  
  button:scope {  
    /* Default style */  
    font-size: 1rem;  
    border-radius: 0.25rem;  
  
    &[data-size='lg'] {  
      font-size: 1.5rem;  
    }  
  }  
}
```

CSS

```
&[data-size='sm'] {
  font-size: 0.75rem;
}

&[data-danger] {
  background-color: red;
  color: white;
}
}
}
```

WARNING

The class in `@scope` needs to be unique across your project. Plans are in place to automate this check.

INFO

100 of the CSS code above is standard and valid. MistCSS doesn't introduce any proprietary syntax. As a consequence, your code editor will support it out of the box.

Run `mistcss` command:

```
npx mistcss ./src
# Button.mist.tsx will be created
```

shell

You can now import your `Button` component like this:

```
import { Button } from './components/Button.mist'

export default const App = () => (
  <main>
    /* Use it like a normal React component */
    <Button size="lg">Submit</Button>
    <Button size="lg" danger>Delete</Button>
    <Button onClick={handleClick}>Cancel</Button>

    /* 🚀 TypeScript will catch the wrong size here */
    <Button size="foo">Oops</Button>
  </main>
)
```

tsx

```
</main>
```

```
)
```

And just like that a visual and type safe React component was created without writing a single line of JS ✨

The power of CSS

Since MistCSS uses pure CSS, you're not limited and can use **ALL** CSS features directly:

- Container queries `@container` to adapt style based on your component size
- CSS variables, for example `--primary-color` , to have a consistent style
- Media queries `@media (prefers-color-scheme: dark)` , to handle dark mode
- And more...

Of course, you can also use TailwindCSS utility classes in your MistCSS components (click on the [CSS + Tailwind](#) in the Button example above).

Next step

Visit the [workflow](#) section to see how to automatically run `mistcss` as part your dev workflow.

Support

MistCSS is a new project, expect **breaking changes** until `v1.0` .

If you like this project and want to help, please consider having your company [sponsor](#) it. Leaving a ✨ on GitHub is always helpful too.

Thank you 🙏

Why the name?

C in CSS stands for cascade 🌊 → atomized water forms mist 🌫️ → MistCSS creates pure CSS atomic components 🧩

Previous page
[Introduction](#)

Next page
[Integrate to your Workflow](#)