Pelo display:block, esse trecho era pra estar abaixo do Título do Documento.

Installation

Get started with Tailwind CSS

Tailwind CSS works by scanning all of your HTML files, JavaScript components, and any other templates for class names, generating the corresponding styles and then writing them to a static CSS file.

It's fast, flexible, and reliable — with zero-runtime.

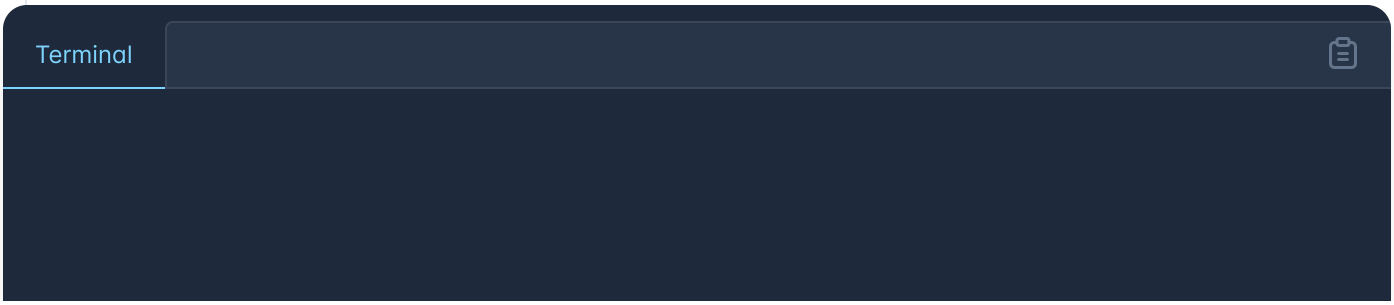
Installation

Tailwind CLI | Using PostCSS | Framework Guides | Play CDN

The simplest and fastest way to get up and running with Tailwind CSS from scratch is with the Tailwind CLI tool. The CLI is also available as a **standalone executable** if you want to use it without installing Node.js.

1 **Install Tailwind CSS**

Install `tailwindcss` via npm, and create your `tailwind.config.js` file.



Pelo display:block, esse trecho era pra estar abaixo do Título do Documento.

```
> npm install -D tailwindcss
> npx tailwindcss init
```

2 **Configure your template paths**

Add the paths to all of your template files in your `tailwind.config.js` file.

tailwind.config.js

```
/** @type {import('tailwindcss').Config} */
module.exports = {
  content: ["./src/**/*.{html,js}"],
  theme: {
    extend: {},
  },
  plugins: [],
}
```

3 **Add the Tailwind directives to your CSS**

Add the `@tailwind` directives for each of Tailwind's layers to your main CSS file.

src/input.css

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

4 **Start the Tailwind CLI build process**

Run the CLI tool to scan your template files for classes and build your CSS.

Pelo display:block, esse trecho era pra estar abaixo do Título do Documento.

Terminal

```
> npx tailwindcss -i ./src/input.css -o ./src/output.css --watch
```

5 **Start using Tailwind in your HTML**

Add your compiled CSS file to the `<head>` and start using Tailwind's utility classes to style your content.

src/index.html

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="./output.css" rel="stylesheet">
</head>
<body>
  <h1 class="text-3xl font-bold underline">
    Hello world!
  </h1>
</body>
</html>
```

What to read next

Get familiar with some of the core concepts that make Tailwind CSS different from writing traditional CSS.

Utility-First Fundamentals

Pelo display:block, esse trecho era pra estar abaixo do Título do Documento.

Using a utility-first workflow to build complex components from a constrained set of primitive utilities.

Responsive Design

Build fully responsive user interfaces that adapt to any screen size using responsive modifiers.

Hover, Focus & Other States

Style elements in interactive states like hover, focus, and more using conditional modifiers.

Dark Mode

Optimize your site for dark mode directly in your HTML using the dark mode modifier.

Reusing Styles

Manage duplication and keep your projects maintainable by creating reusable abstractions.

Customizing the Framework

Customize the framework to match your brand and extend it with your own custom styles.