



Installation

Premiers pas avec Tailwind CSS

Tailwind CSS fonctionne en analysant tous vos fichiers HTML, composants JavaScript et tout autre modèle à la recherche de noms de classe, en générant les styles correspondants, puis en les écrivant dans un fichier CSS statique.

Il est rapide, flexible et fiable, avec une autonomie nulle.

Installation

Tailwind CLI

Utilisation de PostCSS

Guides de cadre

Lire le CDN

Le moyen le plus simple et le plus rapide d'être opérationnel avec Tailwind CSS à partir de zéro est d'utiliser l'outil CLI Tailwind. L'interface de ligne de commande est également disponible en tant [qu'exécutable autonome](#) si vous souhaitez l'utiliser sans l'installer Node.js.

1 Installer Tailwind CSS

Installez via npm et créez votre fichier. `tailwindcss`tailwind.config.js``

Terminal



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

2 Configurer les chemins d'accès à vos modèles

Ajoutez les chemins d'accès à tous vos fichiers de modèle dans votre fichier. `tailwind.config.js``

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

3 Ajouter les directives Tailwind à votre CSS

Ajoutez les directives de chacune des couches de Tailwind à votre fichier CSS principal. `@tailwind`

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

4 Démarrer le processus de génération de l'interface de ligne de commande Tailwind

Exécutez l'outil CLI pour analyser vos fichiers de modèle à la recherche de classes et créer votre CSS.

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

5 Commencez à utiliser Tailwind dans votre HTML

Ajoutez votre fichier CSS compilé à l'et commencez à utiliser les classes utilitaires de Tailwind pour styliser votre contenu. `<head>`

```
<!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

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.

