




HELLO
WORLD



CLASSES UTILITÁRIAS COM TAILWIND



O QUE É TAILWIND CSS?

Tailwind CSS é um framework de CSS baseado em utilitários, ou seja, ele fornece uma série de classes prontas que permitem que você construa designs diretamente no HTML (ou JSX, no caso de React), sem precisar escrever CSS personalizado.

```
1 <div class="space-y-4">
2   <div class="w-96 bg-white shadow rounded">
3     w-96
4   </div>
5   <div class="w-80 bg-white shadow rounded">
6     w-80
7   </div>
8   <div class="w-72 bg-white shadow rounded">
9     w-72
10  </div>
11  <div class="w-64 bg-white shadow rounded">
12    w-64
13  </div>
14  <div class="w-60 bg-white shadow rounded">
15    w-60
16  </div>
17  <div class="w-56 bg-white shadow rounded">
18    w-56
19  </div>
```

BOOTSTRAP VS TAILWIND

Diferente de outros frameworks como **Bootstrap**, **Tailwind** não oferece componentes prontos, mas sim **classes utilitárias** que você pode combinar para criar seu próprio design.



```
1 <button class= "btn btn-success">
2   Success
3 </button>
```



```
1 <div class="bg-red-500 text-black rounded-lg">
2   Tailwind CSS utility classes
3 </div>
```



POR QUE USAR ESTILOS INLINE EM REACT?

01

Isolamento de Estilo: Estilos inline aplicam-se diretamente ao elemento, garantindo que não sejam afetados por regras CSS globais ou classes conflitantes.

02

Dinamismo: Estilos inline permitem modificar estilos de forma dinâmica com base no estado ou nas props do componente.

03

Manutenção Localizada: Para pequenos componentes, manter o estilo junto ao código facilita a leitura e compreensão.

04

Evita Dependências Externas: Não depende de arquivos CSS, simplificando a configuração inicial de projetos menores.

```
1 const style = {  
2   backgroundColor: 'red',  
3   fontSize: 16, // Assume '16px' automaticamente  
4   margin: '10px',  
5 };  
6  
7 return <div style={style}>Texto com estilo inline</div>;  
8
```



ESTILOS INLINE

Vantagens

01

Estilos Dinâmicos: Fáceis de integrar com lógica de estado ou props.

02

Previsibilidade: Sem surpresas causadas por herança ou cascata de estilos.

03

Isolamento: Evita o problema de vazamento ou sobrescrita de estilos globais.

04

Simples de Usar: Perfeito para prototipagem ou componentes pequenos e independentes.

Desvantagens

01

Manutenção Difícil: Para componentes complexos ou com muitos estilos, o uso de objetos de estilo inline pode se tornar confuso.

02

Sem Suporte a Pseudo-classes: Estilos inline não suportam pseudo-classes como **:hover** ou **:focus**. Esses casos exigem JavaScript adicional ou o uso de classes.

03

Sem Agrupamento de Estilos: Não é possível reutilizar regras de estilo em outros componentes sem duplicação de código.

04

Performance: Os navegadores não podem otimizar estilos inline tão eficientemente quanto os definidos em classes CSS.

INSTALAÇÃO DO TAILWIND CSS NO PROJETO REACT

01

Instalar o Tailwind CSS com npm: Para instalar o Tailwind em um projeto React, você pode usar o **npm** ou **yarn**. Aqui, vamos usar o npm. No terminal, dentro do diretório do seu projeto React, execute os seguintes comandos:



```
1 $ npm install -D tailwindcss postcss autoprefixer
```

02

Iniciar o Tailwind: Após a instalação, inicie o Tailwind para gerar os arquivos de configuração:



```
1 $ npx tailwindcss init -p
```



Getting Started

Installation

The simplest and fastest way to get up and running with Tailwind CSS from scratch is with the Tailwind CLI tool.

[SAIBA MAIS](#)



INSTALAÇÃO DO TAILWIND CSS NO PROJETO REACT


03

Configurar o Tailwind: Em seguida, abra o arquivo `tailwind.config.js` e configure o purge para remover o CSS não utilizado em produção, adicionando o seguinte no objeto de configuração:

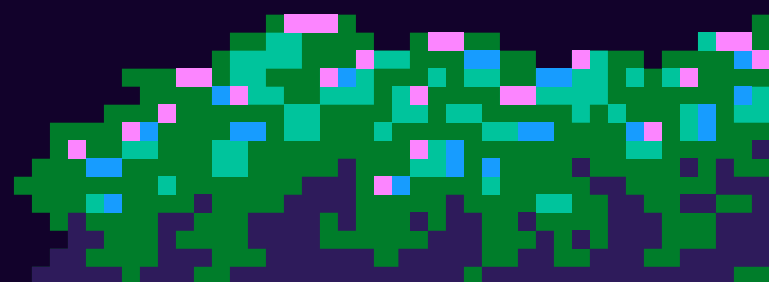
```
1  /** @type {import('tailwindcss').Config} */
2  export default {
3    content: ["../index.html", "./src/**/*.{js,ts,jsx,tsx}"],
4    theme: {
5      extend: {
6        colors: {
7          green: "#4caf50"
8        }
9      }
10   },
11   plugins: []
12 };
13
```

04

Adicionar Tailwind ao CSS: No arquivo `src/index.css` (ou outro arquivo CSS principal), adicione as diretrizes do Tailwind:



```
1  /* Arquivo css principal */
2  @tailwind base;
3  @tailwind components;
4  @tailwind utilities;
```



The background is a pixel art illustration of a park at night. At the top, there are several pixelated clouds in shades of blue, purple, and yellow. In the center, a wooden bench with brown slats sits on a path. The path is flanked by two tall, thin, purple light poles. Each pole has a yellow, pixelated, cone-shaped light fixture at the top. The ground is covered in green grass with small blue and pink flowers. The overall style is retro and digital.

PREFLIGHT

O **Preflight** no **Tailwind CSS** é um conjunto de estilos base incluídos automaticamente para garantir consistência e compatibilidade entre navegadores. Ele funciona como uma redefinição de estilo aprimorada, similar a bibliotecas como **Normalize.css**, mas com otimizações feitas sob medida para o funcionamento do Tailwind CSS.

[SAIBA MAIS](#)

PREFLIGHT

O **Preflight** no **Tailwind CSS** é um conjunto de estilos base que redefine e ajusta os estilos padrões dos navegadores para garantir consistência entre eles.

01

Remove margens, paddings e fontes padrões inconsistentes.

02

Harmoniza elementos como botões, inputs e selects.

03

Preserva comportamentos úteis, como foco em links e botões.

04

Prepara o ambiente para as classes utilitárias do Tailwind.



USANDO TAILWIND

O Tailwind CSS utiliza uma abordagem baseada em classes utilitárias para aplicar estilos diretamente nos elementos HTML. Em vez de escrever regras CSS personalizadas, você combina classes pré-definidas para criar o design desejado

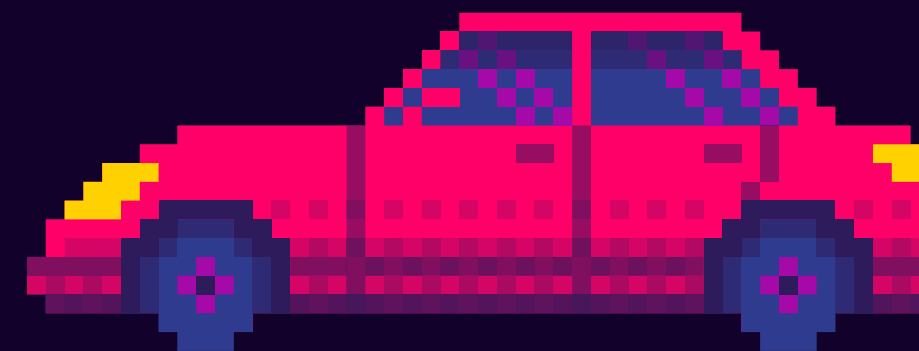
Como funciona na prática?

Você aplica classes diretamente nos elementos, onde cada classe tem um propósito específico, como definir margens, cores, tamanhos ou posicionamento.



```
1 <div class="bg-blue-500 text-white font-bold p-4 rounded-lg">
2   Botão estilizado com Tailwind
3 </div>
```

- bg-blue-500: Define o fundo como azul.
- text-white: Define a cor do texto como branco.
- font-bold: Deixa o texto em negrito.
- p-4: Adiciona padding (espaçamento interno).
- rounded-lg: Adiciona bordas arredondadas.





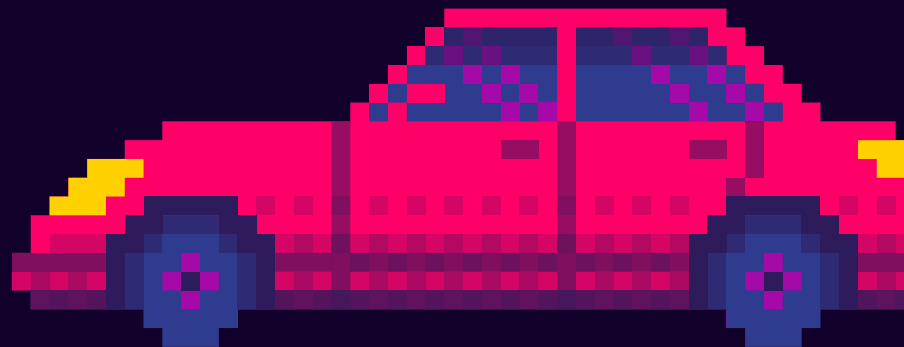
O QUE SÃO CLASSES UTILITÁRIAS?

Classes utilitárias são classes CSS que aplicam um único estilo ou propriedade específica a um elemento. Elas seguem a filosofia de "uma classe, um propósito", permitindo criar designs diretamente no HTML sem escrever CSS adicional.

Características das classes utilitárias

1. Granularidade: Cada classe controla uma propriedade CSS isolada.
2. Composição: Estilos complexos são criados combinando várias classes utilitárias.
3. Reutilização: Você reutiliza as mesmas classes em diferentes elementos, evitando a repetição de código.

Vantagens do uso de classes utilitárias

1. Rapidez no desenvolvimento: Com as classes já disponíveis, você não precisa escrever CSS personalizado para a maioria dos estilos comuns.
 2. Consistência no design: Ao usar classes predefinidas, o estilo é uniforme em todo o projeto.
 3. Redução de CSS personalizado: O código CSS manual é minimizado, reduzindo problemas com cascata ou herança.
 4. Facilidade de prototipagem: É possível criar designs rapidamente sem precisar configurar arquivos de estilo.
- 



POR QUE USAR CLASSES UTILITÁRIAS NO TAILWIND?

01

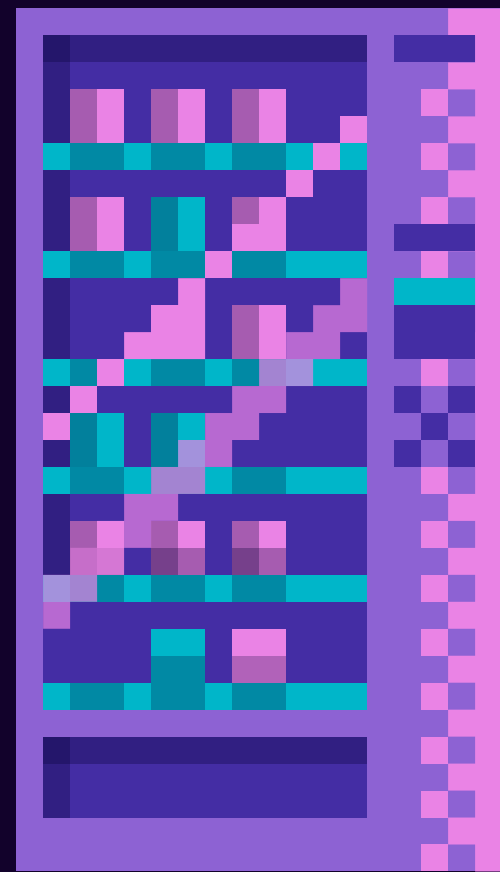
Controle total no HTML: Todas as regras de estilo estão diretamente visíveis no arquivo HTML, tornando mais fácil entender como o elemento está sendo estilizado.

02

Fácil manutenção: Evita problemas de cascata, já que cada classe tem uma função clara.

03

Altamente configurável: É possível personalizar as classes no arquivo *tailwind.config.js*, ajustando o design para atender às necessidades do projeto.



UTILITY-FIRST

Utility-first é uma abordagem de design que prioriza o uso de classes utilitárias simples e específicas para aplicar estilos diretamente aos elementos HTML, em vez de escrever CSS personalizado ou depender de classes específicas. Essa filosofia é central no Tailwind CSS.

Vantagens:

1. Consistência: Como as classes são predefinidas e reutilizáveis, o design mantém um padrão visual consistente em todo o projeto.
2. Desenvolvimento Rápido: Aplique estilos rapidamente sem escrever CSS do zero, facilitando a prototipagem.
3. Isolamento: Estilos são aplicados diretamente aos componentes, evitando conflitos de CSS global.
4. Facilidade de Manutenção: O código CSS é minimalista e fácil de ajustar conforme necessário, com pouca ou nenhuma sobrecarga.



ChitChat

You have a new message!

SAIBA MAIS

```
1 <div class="p-6 max-w-sm mx-auto bg-white rounded-xl shadow-lg flex items-center gap-x-4">
2   <div class="shrink-0">
3     
4   </div>
5   <div>
6     <div class="text-xl font-medium text-black">ChitChat</div>
7     <p class="text-slate-500">You have a new message!</p>
8   </div>
9 </div>
```



GET STARTED WITH TAILWIND CSS

A utility-first CSS framework packed with classes like `flex`, `pt-4`, `text-center` and `rotate-90` that can be composed to build any design, directly in your markup.



SAIBA MAIS



EODRA

COMPOA

NOSSOS CONTATOS:



27 99500-7495



<https://beacons.ai/prismatech>



producaoprismatech@gmail.com



Avenida Jerônimo Monteiro 145, Vitória





THANK
YOU