

Tutorial visual de TailwindCSS usando o Playground

Usei a versão 3.3.3 do TailwindCSS

Requisitos desejáveis

- Conhecimento de CSS
- Conhecimento de Bootstrap (os nomes das classes do Tailwind são parecidos ao CSS puro e ao Bootstrap)

Acessar

<https://play.tailwindcss.com/>

Site de documentação - <https://tailwindcss.com/docs/>

- Clique no painel da esquerda
- Tecle Ctrl+A
- Tecle Delete para apagar tudo

Todas as classes do TailwindCSS estão a nossa disposição para usar. Além disso temos um arquivo CSS e o config.

Exercício1

```
<h1>Tailwind</h1>
```

Veja que logo que digitamos no painel da esquerda, aparece o resultado à direita. Interessante é que mesmo sendo h1 o Tailwind mostra com tamanho de fonte como o de um parágrafo. Tanto faz h1, h2, h3, p, tudo vem com mesmo tamanho.



Exercício2 (centralizar o texto, <https://tailwindcss.com/docs/text-align>),

```
<h1 class="text-center">Tailwind</h1>
```



Exercício3 (aumentar o texto, <https://tailwindcss.com/docs/font-size>)

Adicionar para a classe existente: text-5xl



Exercício4 (Mudar a cor do texto e do fundo)

text-red-600 (<https://tailwindcss.com/docs/text-color>)

bg-orange-300 (<https://tailwindcss.com/docs/background-color>)

Tailwind

Exercício5 (Adicionar padding nos 4 lados do elemento)

Vamos fazer diferente. Digite no painel da esquerda dentro das classes "p-" e aguarde um pouco. Veja que ele mostra as opções existentes. Isso é o autocompletar do Playground, uma característica excelente. Aqui escolho p-8. (<https://tailwindcss.com/docs/padding>)

Tailwind

Exercício6 (Adicionar margens, <https://tailwindcss.com/docs/margin>)

m-32

Tailwind

Exercício7 (Criar um componente, <https://tailwindcss.com/docs/plugins#adding-components>)

Veja que a linha com as classes no nosso exercício está assim:

```
<h1 class="text-center text-5xl text-red-600 bg-orange-300 p-8 m-32">Tailwind</h1>
```

Vamos criar um componente chamado exercício, que contém

`text-center text-5xl text-red-600 bg-orange-300 p-8 m-32`

Clique na aba CSS acima e adicione abaixo:

```
@layer components {  
  .exercicio {  
    @apply text-center text-5xl text-red-600 bg-orange-300 p-8 m-32;  
  }  
}
```

```
HTML  CSS  Config  Tidy  
1  @tailwind base;  
2  @tailwind components;  
3  @tailwind utilities;  
4  @layer components {  
5    .exercicio {  
6      @apply text-center text-5xl text-red-600 bg-orange-300 p-8  
        m-32;  
7    }  
8  }
```

Caso esteja em seu desktop ou num servidor, precisa então executar o comando:

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

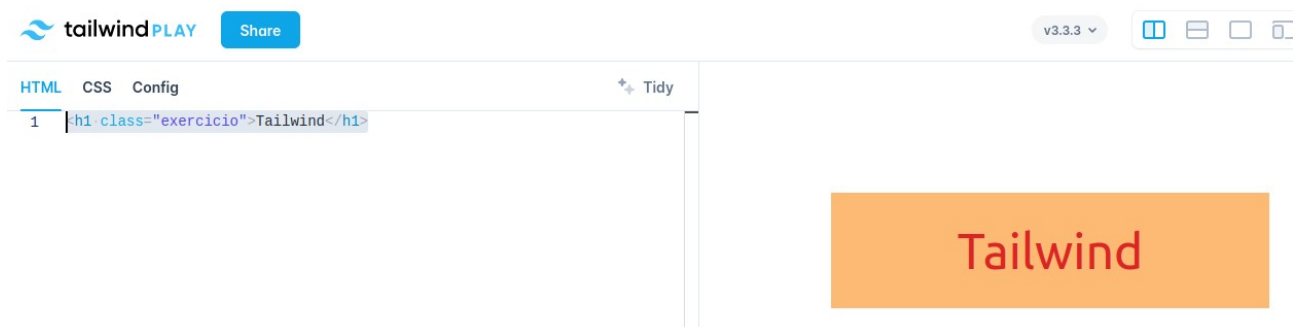
watch é responsável por ficar atualizando automaticamente cada vez que se altere. Devemos executar a linha acima sempre que formos trabalhar novamente com o código (caso tenhamos fechado o terminal).

Volte para a aba HTML e remova todo o conteúdo de class, deixando assim:

```
<h1 class="exercicio">Tailwind</h1>
```

Veja que funciona da mesma forma e deixa nosso HTML menos poluído.

Resultado final



Veja que à direita e acima existem botões para testar a responsividade e uma combo com as versões do TailwindCSS.

Observação – existe uma grande, uma enorme quantidade de classes do TailwindCSS prontas para uso e o melhor lugar para encontrá-las é no site da documentação <https://tailwindcss.com/docs/>

Customização

Caso queira poderá criar suas próprias classes ou componentes para estender o framework.

Autocompletar

Na documentação geralmente não mostra a faixa de valores aceitos para cada classe, mas podemos ver isso em um ambiente com o auto-complete ativado como o Playground ou um editor como o VSCode com a extensão Intellisense instalada: <https://tailwindcss.com/docs/editor-setup#intelli-sense-for-vs-code>

Dica

Se nos depararmos com uma linha de uma class cheia de classes do Tailwind, isso pode, a primeira vista, incomodar. Então, quando iniciar, comece estudando as classes uma a uma. Lembrando que cada classe do Tailwind trata de uma única função.

Ribamar FS