



# Plataformas de aplica  es Web

## Aula 03 - Bibliotecas de componentes UI - Parte 2



Material Did tico do Instituto Metr pole Digital - IMD

### Termo de uso

Os materiais did ticos aqui disponibilizados est o licenciados atrav s de Creative Commons **Atribui  o-SemDeriva  es-SemDerivados CC BY-NC-ND**. Voc  possui a permiss o para realizar o download e compartilhar, desde que atribua os cr ditos do autor. N o poder  alter -los e nem utiliza-los para fins comerciais.

Atribui  o-SemDeriva  es-SemDerivados

CC BY-NC-ND



<https://creativecommons.org/licenses/by-nc-nd/4.0/>

# Apresentação

Dando continuidade ao assunto de Biblioteca de Componentes UI, vamos ver em mais detalhes mais duas dessas bibliotecas: o Materialize e o Tailwind CSS.

Nas páginas seguintes, veremos suas características, exemplos, comparativos e com isso você terá uma melhor noção do que as diferenciam das outras e saber como aprender mais sobre elas.

## Materialize

Website: <https://materializecss.com/>

O Materialize é um framework CSS baseado no guia de estilo visual Material Design do Google.

Também é baseado em um sistema de grid de 12 colunas com classes CSS que determinam como seu layout será aplicado.

O Materialize tem um visual baseado nos aplicativos para dispositivos móveis, com ampla área de entrada de texto, botões, listas, etc. Ele faz grande uso de Javascript também para obter os efeitos visuais desejados.



# Materialize

A modern responsive front-end  
framework based on Material Design

GET STARTED

UPGRADE FROM 0.100.2

Release: 1.0.0

Latest Commit on Github: [824e78248b3de81e383445e76ffb04cc3264fe7d](https://github.com/materializecss/materialize/commit/824e78248b3de81e383445e76ffb04cc3264fe7d) about a year ago

GITHUB



## Speeds up development

We did most of the heavy  
lifting for you to provide a



## User Experience Focused

By utilizing elements and  
principles of Material



## Easy to work with

We have provided detailed  
documentation as well as  
specific code examples to  
help new users get started.

Página web do Materialize. Fonte: <https://materializecss.com/>

O Materialize emprega a filosofia de design moderno de acordo com os guias desenvolvidos pelo Google no seu Material Design e disponível em <https://material.io/design> em uma biblioteca com HTML, CSS (escrito em SASS) e javascripts que combinam as melhores práticas no para o design, portabilidade

entre navegadores e responsividade. É bastante leve e vem com recursos avançados por padrão como flow texts (textos responsivos), efeitos parallax, itens com efeito de hover (quando se passa o mouse por cima), dentre outros.

O Materialize pode ser instalado de diversas formas, incluindo através de CDN. Abaixo um documento HTML com o mínimo necessário para se usar o Materialize:

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Import Google Icon Font -->
    <link
href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">
    <!-- Import materialize.css -->
    <!-- Compiled and minified CSS -->
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/m
aterialize.min.css">

    <!-- Compiled and minified JavaScript -->
    <script defer
src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/mat
erialize.min.js"></script>

    <!-- Let browser know website is optimized for mobile -->
    <meta name="viewport" content="width=device-width, initial-
scale=1.0"/>
  </head>

  <body>

  </body>
</html>
```

## Materialize - Componentes

O sistema de grid do Materialize também tem 12 colunas e as classes base para o seu uso são: row, col, s1, s2, s3..., s11, s12. As classes que iniciam com "s" e são seguidas de um número representam quantas unidades em largura aquela coluna específica utilizará, como pode ser visto na imagem abaixo.

This div is 12-columns wide on all screen sizes

6-columns (one-half)

6-columns (one-half)

language-markup

```
<div class="row">
  <div class="col s12">This div is 12-columns wide on all screen sizes</div>
  <div class="col s6">6-columns (one-half)</div>
  <div class="col s6">6-columns (one-half)</div>
</div>
```

Sistema de grid do Materialize. Fonte: <https://materializecss.com/grid.html>

Assim como o Bulma e o Bootstrap, o sistema de grid permite que você utilize offsets para deslocar colunas, consegue adequar larguras de colunas automaticamente, permite aninhamento, etc.

Para mais informações de como usar o grid system do Materialize visite: <https://materializecss.com/grid.html>

O Materialize vem com alguns efeitos de animação básicos por padrão. Você pode informar que um botão é "pulsante" para chamar atenção, simplesmente adicionando a class "pulse" nesse elemento.

Sombras e transições também são possíveis utilizando classes como "z-depth-2" e "scale-transition".

Alguns exemplos de componentes do Materialize nas figuras abaixo:

# Collections

Alan	1
Alan	4 new
Alan	
Alan	14

language-markup

```
<div class="collection">
  <a href="#" class="collection-item"><span class="badge">1</span>Alan</a>
  <a href="#" class="collection-item"><span class="new badge">4</span>Alan</a>
  <a href="#" class="collection-item">Alan</a>
  <a href="#" class="collection-item"><span class="badge">14</span>Alan</a>
</div>
```

Materialize - Collections. Fonte: <https://materializecss.com>

# Badges in Navbar

Logo

sass

sass

4 new

sass

language-markup

```
<nav>
  <div class="nav-wrapper">
    <a href="#" class="brand-logo">Logo</a>
    <ul id="nav-mobile" class="right hide-on-med-and-down">
      <li><a href="#">sass</a></li>
      <li><a href="#">sass <span class="new badge">4</span></a></li>
      <li><a href="#">sass</a></li>
    </ul>
  </div>
</nav>
```

Materialize - Badges in Navbar. Fonte: <https://materializecss.com>

# Submit Button

When you use a button to submit a form, instead of using a input tag, use a button tag with a type submit



language-markup

```
<button class="btn waves-effect waves-light" type="submit" name="action">Submit  
  <i class="material-icons right">send</i>  
</button>
```

Materialize - Botão com imagem. Fonte: <https://materializecss.com>

First > Second > Third

language-markup

```
<nav>  
  <div class="nav-wrapper">  
    <div class="col s12">  
      <a href="#" class="breadcrumb">First</a>  
      <a href="#" class="breadcrumb">Second</a>  
      <a href="#" class="breadcrumb">Third</a>  
    </div>  
  </div>  
</nav>
```

Materialize - Breadcrumbs. Fonte: <https://materializecss.com>

# Image Card



Card Title

I am a very simple card. I am good at containing small bits of information. I am convenient because I require little markup to use effectively.

THIS IS A LINK

Here is the standard card with an image thumbnail.

language-markup

```
<div class="row">
  <div class="col s12 m7">
    <div class="card">
      <div class="card-image">
        
        <span class="card-title">Card Title</span>
      </div>
      <div class="card-content">
        <p>I am a very simple card. I am good at containing small bits of informatio
        I am convenient because I require little markup to use effectively.</p>
      </div>
      <div class="card-action">
        <a href="#">This is a link</a>
      </div>
    </div>
  </div>
</div>
```

Materialize - Cards. Fonte: <https://materializecss.com>



O Materialize vem com uma coleção de Material Design Icons com mais de 930 ícones prontos para uso, seguindo as especificações do Material Design. Utilizar os ícones é muito simples, basta criar um elemento `<i />` com a classe "material-icons" e no conteúdo do elemento colocar o nome do ícone. Segue um exemplo abaixo:

```
<i class="material-icons">insert_chart</i>
<i class="material-icons">alarm_on</i>
<i class="material-icons">add_to_queue</i>
<i class="material-icons">arrow_back</i>
<i class="material-icons">bug_report</i>
```



Ícones - Materialize. Fonte:

<https://materializecss.com>

A lista completa de ícones pode ser vista em: <https://materializecss.com/icons.html>

Comercialmente o Materialize oferece alguns temas personalizados, mas não de forma gratuita. Trata-se de uma forma de apoiar o projeto financeiramente. Entretanto, não é necessário adquirir nenhum desses temas para ser produtivo utilizando o Materialize.

Se desejar se aprofundar, explore o Materialize com o documento HTML básico adicionando os exemplos da documentação do site dentro da tag `<body>`, faça alterações e observe o resultado.

## Materialize - Vantagens

- **Velocidade de desenvolvimento:** a biblioteca oferece em classes CSS e Javascript, bastantes funcionalidades com pouco código, mesmo com recursos mais avançados.
- **Mobile-first e responsividade:** assim como outras bibliotecas, o Materialize tem foco em dispositivos móveis de forma primária, com elementos de fácil visualização e manipulação em telas de todos os tamanhos
- **Módulos Javascript inclusos:** funcionalidades somente possíveis com Javascript são incluídas no Materialize, diminuindo a necessidade de se utilizar outras bibliotecas.
- **Plugins:** existe um grande número de plugins gratuitos para diálogos, collapse parallax etc.

# Materialize - Desvantagens

- **Comunidade menor:** por ser relativamente novo o Materialize não tem uma comunidade tão grande como o Bootstrap, por exemplo. Entretanto, a quantidade de recursos disponíveis é mais que suficiente para o uso confortável dessa biblioteca.
- **Limitações e restrições:** o Materialize é muito focado no guia do Material Design do Google, tanto em aparência como em comportamento. Sair do padrão não é tão simples com essa biblioteca.
- **Tamanho:** arquivos CSS são grandes, comparativamente aos concorrentes.

Se desejar se aprofundar, explore o Materialize com o documento HTML básico adicionando os exemplos da documentação do site dentro da tag `<body>`, faça alterações e observe o resultado.

## Tailwind CSS

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

Diferente de outras bibliotecas apresentadas aqui, o Tailwind CSS não é uma biblioteca de componentes. Ele se define como um utility-first CSS framework. Isso significa que ao invés de tentar definir como um botão deve ser através de uma simples classe, como no Bootstrap, por exemplo, o Tailwind simplesmente nem tem o conceito de botão. No lugar de definir componentes, como um Button ou um Card, por exemplo, o Tailwind oferece uma coleção de classes que te ajudam a criar o seu próprio botão de uma maneira mais simples que em CSS puro, porém sem criar o botão para você.



# Rapidly build modern websites without ever leaving your HTML.

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.

[Get started](#)`$ npm install tailwindcss`

Página do Tailwind CSS. Fonte: <https://tailwindcss.com/>

Em termos gerais o Tailwind CSS é uma grande coleção de classes que permitem você criar componentes complexos com menos CSS porém com personalização total. Para se ter uma ideia, assim que se adiciona o TailwindCSS em uma página ele remove por padrão até os estilos básicos do HTML como os tamanhos de fonte e margens do H1, H2, H3..., remove a cor azul dos links `<a>` remove o estilo das listas `<ul>` e itens `<li>`, etc. Ele por padrão não adiciona estilo a sua página e sim o remove.

## Tailwind CSS - Características

Abaixo temos uma página pura HTML, sem o TailwindCSS, mas com alguns elementos padrão do HTML como H1, H2, H3, UL, LI e A:

```
<!doctype html>
<html>
<head>
```

```
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-
scale=1.0" />
</head>
<body>
  <h1>Minha página com o Tailwind CSS</h1>

  <h2>H2 personalizado</h2>
  <h3>H3 personalizado</h3>

  <ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
  </ul>

  <a href="https://tailwindcss.com/">https://tailwindcss.com/</a>
</body>
</html>
```

Veja o resultado dessa página HTML pura (Sem o Tailwind CSS):

# Minha página com o Tailwind CSS

## H2 personalizado

### H3 personalizado

- Item 1
- Item 2
- Item 3

<https://tailwindcss.com/>

*Página em HTML*

*puro, sem nenhuma biblioteca. Fonte: Autor*

Agora vamos somente adicionar o Tailwind CSS acrescentando a referência ao CDN em que ele está hospedado:

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
  <!-- Adicionando o Tailwind CSS-->
```

```
<link href="https://unpkg.com/tailwindcss@^2/dist/tailwind.min.css"
rel="stylesheet">
</head>
<body>
  <h1>Minha página com o Tailwind CSS</h1>

  <h2>H2 personalizado</h2>
  <h3>H3 personalizado</h3>

  <ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
  </ul>

  <a href="https://tailwindcss.com/">https://tailwindcss.com/</a>
</body>
</html>
```

Adicionamos somente a linha do CDN do Tailwind e veja agora como a página ficou:

Minha página com o Tailwind CSS

H2 personalizado

H3 personalizado

Item 1

Item 2

Item 3

<https://tailwindcss.com/>

*Página HTML com o Tailwind CSS*

*adicionado. Fonte: Autor*

Repare que todos os elementos perderam o estilo padrão que o HTML oferece.

Esse comportamento do Tailwind é algo proposital. Isso faz com que você não tenha problemas com diferentes navegadores tentando exibir elementos HTML de maneira distinta, o que pode quebrar a coerência visual do seu site. Além disso, dificilmente um sistema vai usar os estilos padrão do HTML e precisa de toda forma os substituir por outros mais agradáveis e modernos, então no fim das contas está se ganhando mais que perdendo quando se começa do zero.

Como uma biblioteca que remove meus estilos vai me ajudar? você pode se perguntar? Calma. A seguir vamos ver um exemplo da própria documentação do Tailwind para que você entenda melhor como podemos personalizar com bastante liberdade nossas páginas com essa ferramenta.

# Tailwind CSS - Exemplo

Suponha que deseja criar o seguinte componente, como na imagem abaixo, na sua página:



*Componente exemplo.*

Fonte: Autor

Sim, parece um elemento simples, mas repare que ele contém bordas arredondadas, uma imagem à esquerda e dois textos posicionados de maneira específica, além de uma sombra. Talvez você consiga criar algo similar com outros frameworks CSS, mas vamos por enquanto utilizar HTML e CSS puro pra isso. O resultado seria algo assim:

```
<div class="chat-notification">
  <div class="chat-notification-logo-wrapper">
    
  </div>
  <div class="chat-notification-content">
    <h4 class="chat-notification-title">Bate Papo</h4>
    <p class="chat-notification-message">Você tem uma nova
mensagem!</p>
  </div>
</div>
<style>
.chat-notification {
  display: flex;
  max-width: 24rem;
  margin: 0 auto;
  padding: 1.5rem;
  border-radius: 0.5rem;
  background-color: #fff;
  box-shadow: 0 20px 25px -5px rgba(0, 0, 0, 0.1), 0 10px 10px
-5px rgba(0, 0, 0, 0.04);
}
.chat-notification-logo-wrapper {
  flex-shrink: 0;
}
```

```
}  
.chat-notification-logo {  
  height: 3rem;  
  width: 3rem;  
}  
.chat-notification-content {  
  margin-left: 1.5rem;  
  padding-top: 0.25rem;  
}  
.chat-notification-title {  
  color: #1a202c;  
  font-size: 1.25rem;  
  line-height: 1.25;  
}  
.chat-notification-message {  
  color: #718096;  
  font-size: 1rem;  
  line-height: 1.5;  
}  
</style>
```

Repare que foram criadas classes próprias para cada um dos elementos do componente e logo depois foram adicionados estilos para essas classes, para se obter o visual desejado.

Como o Tailwind oferece várias classes de estilo, vamos ver como se criaria esse mesmo componente utilizando ele:

```
<div class="p-6 max-w-sm mx-auto bg-white rounded-xl shadow-md flex  
items-center space-x-4">  
  <div class="flex-shrink-0">  
      
  </div>  
  <div>  
    <div class="text-xl font-medium text-black">Bate Papo</div>  
    <p class="text-gray-500">Você tem uma nova mensagem!</p>  
  </div>  
</div>
```

E o resultado:

**Bate Papo**

Você tem uma nova mensagem!

*Componente criado com o**TailwindCSS. Fonte: Autor*

Bem mais simples. Repare que não precisamos utilizar nenhum CSS extra, fora as classes do Tailwind para criar esse componente totalmente personalizado. Entretanto usamos algumas classes utilitárias disponíveis para chegar ao resultado. Vamos explorar quais foram elas:

- Utilitários de flexbox and padding (flex, flex-shrink-0 e p-6) para controlar o layout geral do card (componente geral).
- Utilitários de max-width e margin (max-w-sm e mx-auto) para restringir a largura do card e centralizá-lo horizontalmente
- Utilitários de background color, border radius, e box-shadow (bg-white, rounded-xl, e shadow-md) para estilizar a aparência do card
- Utilitários de width(largura) e height(altura) (w-12 e h-12) para dimensionar a imagem do logo
- O utilitário space-between(espaco entre): (space-x-4) para ajustar o espaçamento entre a logo e o texto.
- Os utilitários de font size(tamanho da fonte), text color(cor do texto), and font-weight(peso da fonte) (text-xl, text-black, font-medium etc.) para estilizar o texto do card.

Veja que o Tailwind é uma ferramenta que ajuda muito quem deseja criar interfaces totalmente personalizadas com utilitários que diminuem a necessidade de se usar CSS puro.

Se você conhece bem estilos CSS, pode se perguntar: "Por que então não se usar estilos inline com a atributo style?".

O que acontece é que o Tailwind é uma abordagem superior a estilos inline pois ele oferece suporte a estados com, por exemplo, o "hover", "focus", etc, fornece responsividade e permite uma abordagem de design baseada em restrições.

Vamos mudar o elemento que tem "Você tem uma nova mensagem" com as classes Tailwind abaixo e ver o que acontece quando passo o mouse sobre a mensagem:

```
<p class="text-gray-500 rounded p-1 text-blue-400 cursor-pointer  
hover:text-white hover:bg-purple-600 ">Você tem uma nova  
mensagem!</p>
```



Sem o mouse por cima



Bate Papo

Você tem uma nova mensagem!

Com o mouse por cima



Bate Papo

Você tem uma nova mensagem!

Componente Tailwind com novas classes. Fonte: Autor

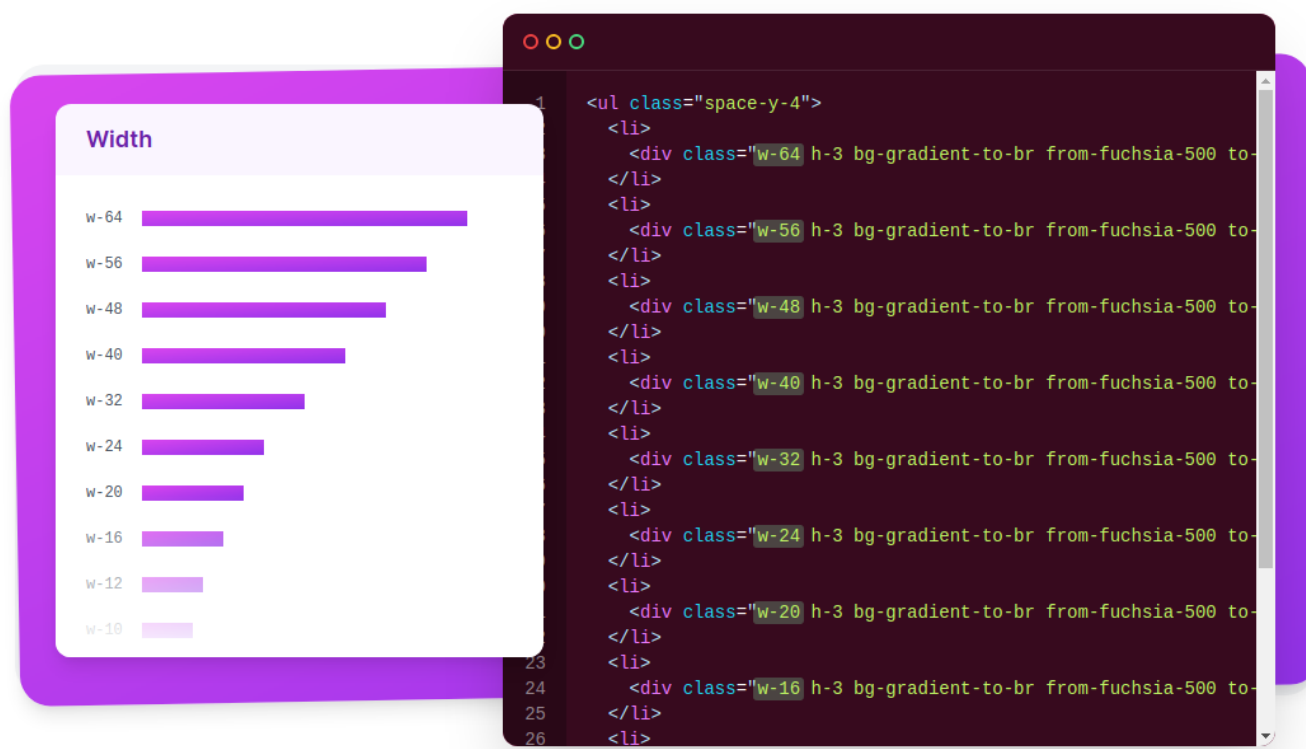
Veja o que modificamos no texto da mensagem com cada classe extra:

- **rounded**: informamos que o elemento tem cantos arredondados
- **p-1**: adicionamos um padding simples no elemento
- **text-blue-400**: a cor padrão do elemento é um azul 400 (o Tailwind tem uma tabela de cores nesse padrão. Quando maior o número mais forte e escura é a cor)
- **cursor-pointer**: informamos que o cursor deve indicar que o elemento é clicável
- **hover:text-white**: quando o mouse passa por cima o texto muda para branco
- **hover:bg-purple-600**: quando o mouse passa por cima o fundo muda para roxo 600

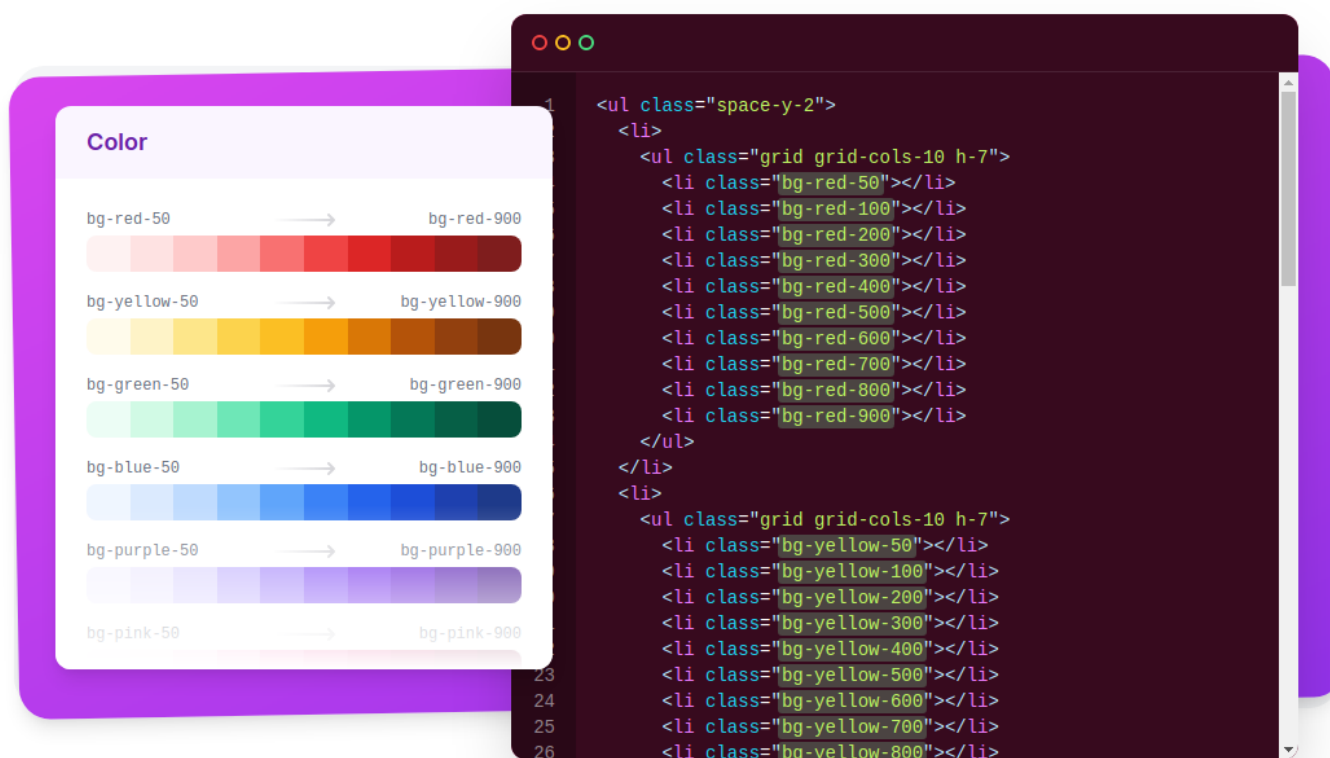
## Tailwind CSS - Possibilidades

Como pode ser observado o poder do TailwindCSS é muito grande. Ele permite que você crie páginas com um grau de personalização extremo com classes que personalizam cada um de seus elementos com as características que você desejar.

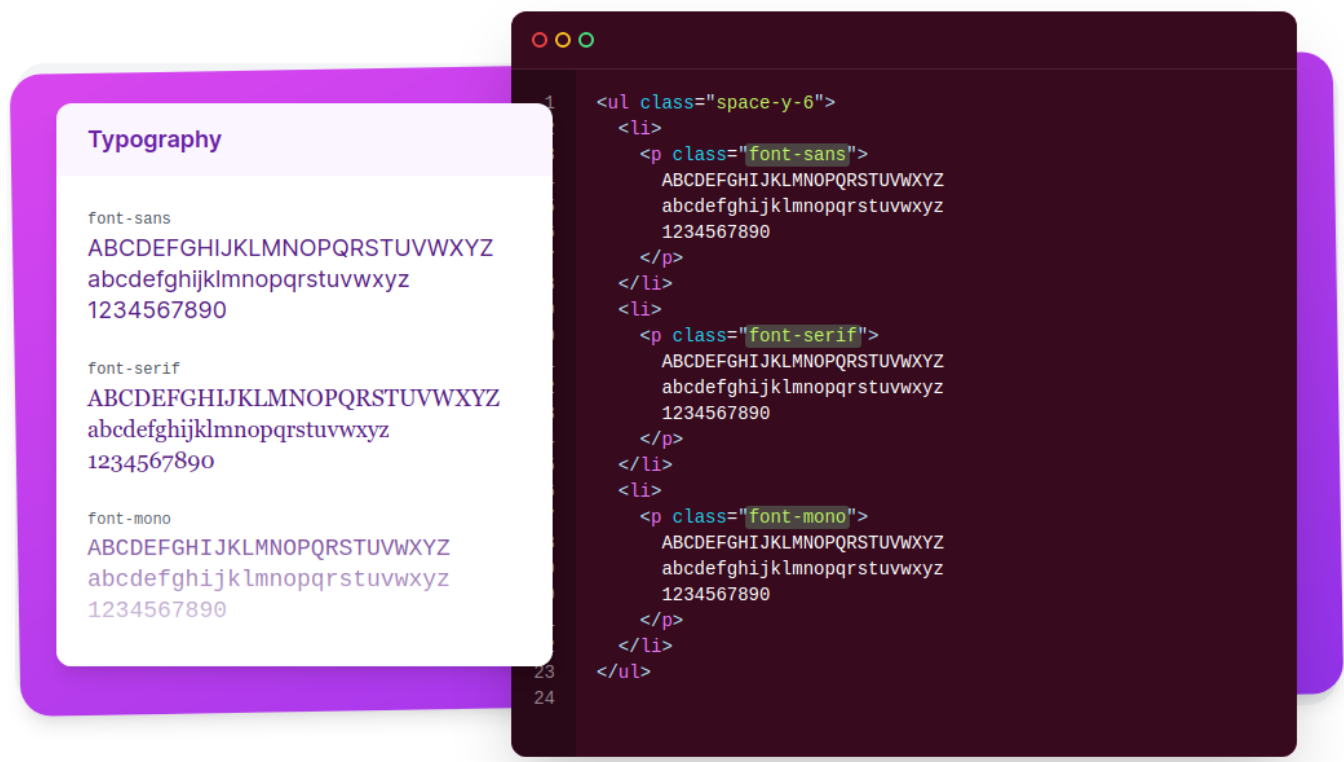
Seguem algumas imagens das possibilidades do Tailwind CSS:



Tailwind CSS - Dimensões. Fonte: <https://tailwindcss.com>



Tailwind CSS - Cores. Fonte: <https://tailwindcss.com>



Tailwind CSS - Tipografia. Fonte: <https://tailwindcss.com>



Tailwind CSS - Sistema de grid (você quem define quantas colunas terá). Fonte: <https://tailwindcss.com>



```
<img class="transform rotate-45 ...">  
<img class="transform skew-y-12 ...">  
<img class="transform scale-50 ...">  
<img class="transform translate-x-4 translate-y-4 ...">
```

Tailwind CSS - Transformações de rotação, torção, escala e translação. Fonte: <https://tailwindcss.com>

## Tailwind CSS - Quando utilizar

Talvez a aplicação que você esteja trabalhando não necessite de componentes com esse grau de personalização e que bibliotecas como o Bootstrap ou o Bulma sejam uma melhor opção por trazer componentes mais utilizados já prontos e com um visual agradável. Nesse caso o Tailwind não é a melhor opção, mas se você deseja realmente personalizar com liberdade seus componentes sem precisar escrever muito CSS puro ou tentar alterar drasticamente o visual padrão do Bulma ou Bootstrap ou Materialize, o Tailwind é sem dúvida uma das melhores opções disponíveis.

Se desejar se aprofundar, explore o Tailwind CSS com o documento HTML básico adicionando os exemplos da documentação do site dentro da tag `<body>`, faça alterações e observe o resultado.