

Vitor Amadeu Souza

Introdução ao

Tailwind CSS

Aplicações práticas – Parte III

© 2024 by Cerne Tecnologia e Treinamento Ltda.

© 2024 by Vitor Amadeu Souza

Nenhuma parte desta publicação poderá ser reproduzida sem autorização prévia e escrita de **Cerne Tecnologia e Treinamento Ltda.**

Este livro publica nomes comerciais e marcas registradas de produtos pertencentes a diversas companhias. O editor utiliza as marcas somente para fins editoriais e em benefício dos proprietários das marcas, sem nenhuma intenção de atingir seus direitos.

Dezembro de 2024

Direitos reservados por:

Cerne Tecnologia e Treinamento Ltda

Produção: Cerne Tecnologia e Treinamento

E-mail da Empresa: cerne@cerne-tec.com.br

Home Page: www.cerne-tec.com.br.com.br

Atendimento ao Consumidor: sac@cerne-tec.com.br

Contato com o Autor: vitor@cerne-tec.com.br



FEITO NO BRASIL

*“A melhor maneira de se defender contra o mal é não se tornar como
ele.”*

Marco Aurélio

Cerne Tecnologia

A Cerne Tecnologia tem uma equipe preparada para desenvolvimento de projetos elétricos, eletrônicos, automação em diversas áreas: Médica, Entretenimento, Industrial, Robótica, Científica, Automobilística, Aeronáutica, etc. Trabalhamos com tecnologia microcontrolada usando o PIC, ARM, AVR, 8051, dsPIC, PIC24, PIC32 além do Arduino, Raspberry, Beaglebone etc. Desenvolvemos o projeto desde sua concepção até a entrega do produto final, passando pelas etapas de esquema elétrico, protótipo e desenvolvimento de circuito impresso.

Desenvolvemos aplicativos para smartphones/tablets Android e iOS e no desenvolvimento de softwares a nível PC para plataforma Windows, usando ferramentas como o Visual Basic, C# e C++.

Atuamos na parte de montagem de placas, onde podemos fornecer ambos os serviços de desenvolvimento de projetos e produção ou apenas um destes.

Desenvolvemos esquemas elétricos e layout de PCB, tanto em tecnologia convencional como SMD.

Temos a flexibilidade de customizar um de nossos produtos, de modo a atender a uma necessidade específica do cliente, tornando o custo de desenvolvimento menor se comparado a construção de um projeto desde a sua fase inicial.

Desenvolvemos e fornecemos kits didáticos para diversos microcontroladores além de apostilas, livros e e-books.

Na hora de desenvolver um projeto ou equipar seu laboratório não hesite em nos contatar. Entre em contato conosco através do endereço cerne-tec.com.br para obter mais informações.



Sumário

Introdução	6
II. Tailwind CSS	7
1. space-x-{size}	7
2. space-y-{size}	10
3. text-{size}	12
4. font-{style}	15
5. text-center	17
6. leading-{size}	19

Introdução

A proposta desta série de livros é explorar o uso do Tailwind CSS, começando com uma introdução ao HTML e aos comandos iniciais do framework. Nas partes seguintes, são abordados diversos comandos e técnicas avançadas, permitindo ao leitor entender e aplicar de maneira eficaz as funcionalidades do Tailwind CSS em seus projetos.

Consulte o sumário de cada parte para conferir os temas detalhados abordados em cada etapa da série.

Capítulo I

Tailwind CSS

1. `space-x-{size}`

No Tailwind CSS, a classe `space-x-{size}` é utilizada para adicionar um espaçamento horizontal uniforme entre os elementos adjacentes dentro de um contêiner. Essa classe é ideal para organizar elementos em linha, pois aplica margens automáticas apenas entre os itens, sem afetar o primeiro elemento (que não recebe margem à esquerda) nem o último elemento (que não recebe margem à direita). Os valores de espaçamento podem ser configurados a partir de uma escala predefinida, como `space-x-1` para 4px, `space-x-2` para 8px, ou valores maiores como `space-x-4` para 16px. Também é possível usar valores personalizados, como `space-x-[10px]`. Para que a classe funcione, o contêiner deve ter um layout flexível definido com `flex` ou `inline-flex`. Isso simplifica o código e garante que os elementos em um layout horizontal mantenham espaçamentos consistentes e visualmente equilibrados.

Acompanhe um exemplo que faz uso deste comando.

```

<!DOCTYPE html>
<html lang="pt-BR">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplo de space-x-{size} - Tailwind CSS</title>
  <script src="https://cdn.tailwindcss.com"></script>
</head>

<body class="bg-gray-100 flex items-center justify-center h-screen">

  <div class="flex space-x-4">
    <button class="bg-blue-500 text-white px-4 py-2 rounded">
      Botão 1
    </button>
    <button class="bg-green-500 text-white px-4 py-2 rounded">
      Botão 2
    </button>
    <button class="bg-red-500 text-white px-4 py-2 rounded">
      Botão 3
    </button>
  </div>

</body>

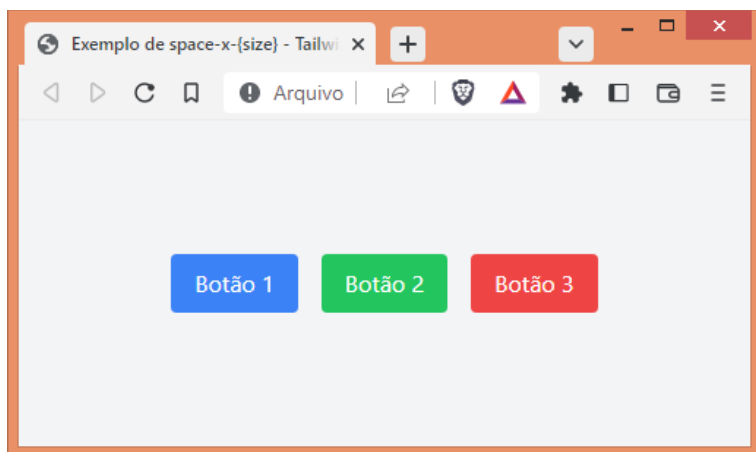
</html>

```

O código apresentado utiliza o Tailwind CSS para criar uma página com um layout centralizado na tela e um conjunto de botões organizados horizontalmente, com espaçamento uniforme entre eles. O contêiner principal é estilizado com a classe flex, que organiza os elementos em um layout flexível, e as classes items-center e justify-center centralizam o contêiner vertical e horizontalmente dentro da página, que possui um fundo cinza claro definido pela classe bg-gray-100.

Dentro do contêiner flexível, três botões são exibidos lado a lado. A classe `space-x-4` aplicada ao contêiner adiciona um espaçamento horizontal de 16px (1rem) entre cada botão, sem aplicar margens extras no primeiro ou no último elemento. Cada botão tem um estilo único, com uma cor de fundo específica (`bg-blue-500`, `bg-green-500`, `bg-red-500`), texto branco (`text-white`), bordas arredondadas (`rounded`) e padding interno (`px-4 py-2`) para ajustar o tamanho.

O resultado é um layout limpo e alinhado, onde os botões estão igualmente espaçados, garantindo uma aparência organizada e responsiva.



2. space-y-{size}

No Tailwind CSS, a classe `space-y-{size}` é utilizada para adicionar espaçamento vertical uniforme entre os elementos filhos de um contêiner. Essa classe é especialmente útil para organizar itens em uma coluna, eliminando a necessidade de aplicar margens individuais em cada elemento. Quando aplicada, ela insere margens verticais automáticas entre os elementos, sem afetar o primeiro, que não recebe margem na parte superior, nem o último, que não recebe margem na parte inferior. Os tamanhos de espaçamento seguem uma escala predefinida, como `space-y-1` para 4px, `space-y-2` para 8px, ou `space-y-4` para 16px, podendo também ser personalizados com valores arbitrários, como `space-y-[10px]`. Essa classe é geralmente usada em contêineres com layout flexível em direção vertical, definidos com `flex-col`. Por exemplo, em um contêiner contendo três elementos com a classe `space-y-4`, cada um teria um espaçamento vertical de 16px entre eles, garantindo um layout limpo e organizado.

```
<!DOCTYPE html>  
<html lang="pt-BR">
```