

Vitor Amadeu Souza

Introdução ao

Tailwind CSS

Aplicações práticas – Parte V

© 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

*“Seja como a rocha que as ondas continuam a bater, mas que
permanece inabalável.”*

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 PCI, 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. justify-center	7
2. grid	10
3. grid-cols-{n}	14
4. gap-{size}	17

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. justify-center

A classe `justify-center` do Tailwind CSS é utilizada para alinhar os itens de um contêiner flexível ao longo do eixo principal, centralizando-os horizontalmente. Quando aplicada a um contêiner com a classe `flex`, a classe `justify-center` garante que todos os itens filhos sejam posicionados no centro do contêiner, proporcionando um layout equilibrado e esteticamente agradável. Por exemplo, ao usar `<div class="flex justify-center">`, todos os elementos dentro desse contêiner serão centralizados. Essa classe é uma alternativa ao CSS padrão `justify-content: center`; e é especialmente útil em layouts onde espera-se que os itens apareçam uniformemente distribuídos no centro da área disponível.

Além de `justify-center`, o Tailwind CSS oferece outras classes para controlar o alinhamento dos itens, como `justify-start`, que alinha os itens ao início do contêiner; `justify-end`, que alinha ao final; `justify-between`, que distribui os itens com espaço igual entre eles; e `justify-`

around, que cria espaço igual ao redor de cada item. Essas classes permitem uma flexibilidade significativa na criação de layouts responsivos e organizados. 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 justify-center - Tailwind CSS</title>
  <script src="https://cdn.tailwindcss.com"></script>
</head>
<body class="bg-gray-100 h-screen flex items-center justify-center">

  <!-- Contêiner principal com flex e alinhamento central -->
  <div class="flex justify-center bg-blue-200 p-4 rounded shadow">
    <!-- Item 1 -->
    <div class="bg-blue-500 text-white p-4">Item 1</div>
    <!-- Item 2 -->
    <div class="bg-green-500 text-white p-4 ml-4">Item 2</div>
    <!-- Item 3 -->
    <div class="bg-red-500 text-white p-4 ml-4">Item 3</div>
  </div>

</body>
</html>
```

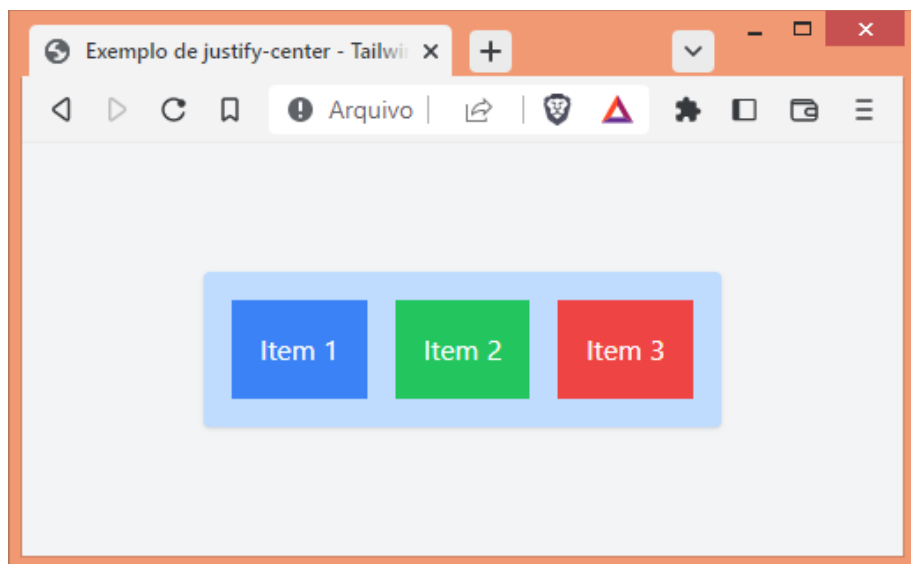
O documento começa com a declaração do tipo HTML e define o idioma como português do Brasil. No cabeçalho, são especificados o conjunto de caracteres, a viewport e o título da página, além de incluir o script do Tailwind CSS.

No corpo do documento, a classe bg-gray-100 aplica um fundo cinza claro à página, enquanto as classes h-screen, flex, items-center,

e justify-center são utilizadas no elemento body para centralizar todo o conteúdo vertical e horizontalmente na tela. Dentro desse corpo centralizado, há uma div que atua como contêiner principal. A classe flex transforma essa div em um contêiner flexível, e a classe justify-center alinha todos os itens filhos horizontalmente no centro do contêiner.

Essa div tem um fundo azul claro (bg-blue-200), um preenchimento (p-4), bordas arredondadas (rounded) e uma sombra (shadow). Dentro dela, existem três itens representados por divs, cada um com diferentes cores de fundo: o primeiro item é azul escuro (bg-blue-500), o segundo é verde escuro (bg-green-500) e o terceiro é vermelho escuro (bg-red-500). Todos os itens têm texto branco e preenchimento, com uma margem à esquerda (ml-4) nos itens subsequentes para proporcionar espaçamento.

Esse exemplo ilustra claramente como usar a classe justify-center para criar um layout responsivo e visualmente agradável, com itens centralizados horizontalmente dentro de um contêiner flexível.



2. grid

O sistema de grid do Tailwind CSS permite criar layouts responsivos e altamente personalizáveis utilizando CSS Grid. Ao contrário de outros frameworks que utilizam flexbox, o Tailwind utiliza o grid para construir layouts de qualquer número de colunas e linhas, oferecendo classes utilitárias como `grid`, `grid-cols-{n}` e `grid-rows-{n}` para definir o comportamento do contêiner e dos itens. Para criar um contêiner de grid, você deve usar a classe `grid`. Para definir o número de colunas, pode-se utilizar classes como `grid-cols-3` para um layout com três colunas. O Tailwind permite até 12 colunas por