

Flexbox

Objetivo:

Ensinar de forma prática e aprofundada o uso do **CSS Flexbox**, abordando desde os conceitos fundamentais até a criação de layouts responsivos, com exemplos completos para desktop e mobile.

1. Introdução ao Flexbox

O **Flexbox** (Flexible Box Layout) é um modelo de layout unidimensional do CSS, criado para alinhar e distribuir espaço entre elementos de forma eficiente, mesmo quando o tamanho dos elementos é desconhecido ou dinâmico.

Ele é ideal para:

- Criar alinhamentos verticais e horizontais;
- Distribuir espaço proporcional entre itens;
- Tornar layouts mais responsivos e adaptáveis.

Para usar o Flexbox, aplicamos a propriedade `display: flex` ao **container**.

2. Conceitos básicos

O Flexbox funciona em **dois eixos**:

- **Eixo principal (main axis):**

Definido pela propriedade `flex-direction`.

Valores: `row`, `row-reverse`, `column`, `column-reverse`.

- **Eixo transversal (cross axis):**

Perpendicular ao eixo principal. Controlado por `align-items` ou `align-self`.

Exemplo inicial:

```
.container {  
  display: flex;  
}
```

3. Propriedades do container

Propriedade	Descrição
<code>display</code>	Ativa o flex: <code>display: flex</code> ou <code>inline-flex</code>
<code>flex-direction</code>	Define a direção dos itens (<code>row</code> , <code>column...</code>)
<code>flex-wrap</code>	Permite quebra de linha (<code>nowrap</code> , <code>wrap</code>)
<code>justify-content</code>	Alinha itens no eixo principal
<code>align-items</code>	Alinha itens no eixo transversal
<code>align-content</code>	Alinha múltiplas linhas no eixo transversal (quando há quebra)
<code>gap</code>	Espaço entre os itens (substitui <code>margin</code> individual para espaçamento interno)

4. Propriedades dos itens flexíveis

Propriedade	Descrição
<code>flex-grow</code>	Fator de crescimento do item
<code>flex-shrink</code>	Fator de encolhimento do item
<code>flex-basis</code>	Tamanho base do item antes de ajustes
<code>order</code>	Muda a ordem de exibição
<code>align-self</code>	Alinhamento individual no eixo transversal

5. Exemplo prático – Galeria de cards com Flexbox

HTML – `index.html`

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Galeria Flexbox</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <header>
    <h1>Minha Galeria</h1>
  </header>

  <main class="container">
    <div class="card">Card 1</div>
    <div class="card">Card 2</div>
    <div class="card">Card 3</div>
    <div class="card">Card 4</div>
  </main>
</body>
</html>
```

CSS – `style.css`

```
/* Reset básico */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

/* Estilo geral */
body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
}

header {
  background-color: #007bff;
  color: white;
```

```
padding: 20px;
text-align: center;
}

/* Container Flex */
.container {
display: flex;
flex-wrap: wrap; /* Permite quebrar linhas */
justify-content: space-between;
gap: 16px;
padding: 20px;
}

/* Cards */
.card {
flex: 1 1 calc(25% - 16px); /* Cresce, encolhe e ocupa 25% menos o gap */
background-color: white;
border-radius: 8px;
padding: 20px;
text-align: center;
box-shadow: 0 2px 8px rgba(0,0,0,0.1);
}
```

6. Tornando responsivo (Mobile First)

Vamos adicionar media queries para adaptar o layout a diferentes tamanhos de tela.

```
/* Mobile: 100% largura */
@media (max-width: 600px) {
  .card {
    flex: 1 1 100%;
  }
}

/* Tablet: 2 colunas */
@media (min-width: 601px) and (max-width: 900px) {
  .card {
    flex: 1 1 calc(50% - 16px);
  }
}

/* Desktop: 4 colunas (padrão definido no CSS principal) */
```

7. Boas práticas ao usar Flexbox

- Usar `gap` para espaçamento entre itens, evitando `margin` manual repetida. Combinar com **Mobile First** para garantir acessibilidade e desempenho.
- Evitar definir larguras fixas, preferindo valores flexíveis (`flex-basis` ou `%`).
- Lembrar que Flexbox é **unidimensional** (use CSS Grid para estruturas mais complexas).

8. Resultado esperado

No celular:

- Cards empilhados, ocupando 100% da largura.

No desktop:

- Cards dispostos lado a lado, 4 por linha, com espaçamento uniforme.