

CSS Flexbox: Layout Flexível e Responsivo

O Flexbox é um módulo de layout unidirecional no CSS3, projetado para ajudar a alinhar, justificar e distribuir itens em um contêiner.

1. Ativando o Flexbox

A primeira etapa é definir o elemento pai como um **Contêiner Flex (Flex Container)**.

Propriedade	Exemplo de Declaração	Descrição
display	.container { display: flex; }	Define o contêiner como flexível.
display	.container { display: inline-flex; }	Define um contêiner flexível que se comporta como um elemento inline.

2. Propriedades do Contêiner Flex

Estas propriedades controlam a direção principal, o envolvimento das linhas e o alinhamento geral dos itens.

2.1. Direção e Fluxo (Main Axis)

Controla a direção do **eixo principal (main axis)**.

Propriedade	Exemplo de Declaração	Descrição dos Valores
flex-direction	.c { flex-direction: row; }	(Padrão) Itens dispostos horizontalmente, da esquerda para a direita.
flex-direction	.c { flex-direction: column; }	Itens dispostos verticalmente, de cima para baixo.
flex-direction	.c { flex-direction: row-reverse; }	Itens horizontalmente, da direita para a esquerda.
flex-direction	.c { flex-direction: column-reverse; }	Itens verticalmente, de baixo para cima.
flex-wrap	.c { flex-wrap: wrap; }	Permite que os itens quebrem para a próxima linha se não houver espaço.
flex-wrap	.c { flex-wrap: nowrap; }	(Padrão) Todos os itens na mesma linha.
flex-flow	.c { flex-flow: row wrap; }	Shorthand para flex-direction e flex-wrap .

2.2. Alinhamento e Distribuição

Controla como o espaço livre é distribuído ao longo do **eixo principal** (main axis) e do **eixo transversal** (cross axis).

Propriedade	Exemplo de Declaração	Descrição dos Valores
justify-content	.c { justify-content: center; }	Alinha os itens ao centro do eixo principal.
justify-content	.c { justify-content: space-between; }	Distribui o espaço uniformemente <i>entre</i> os itens (sem margem nas bordas).
justify-content	.c { justify-content: space-around; }	Distribui o espaço uniformemente <i>ao redor</i> de cada item.
justify-content	.c { justify-content: flex-start; }	Alinha os itens ao início do eixo principal.
align-items	.c { align-items: center; }	Alinha os itens ao centro do eixo transversal (verticalmente em row).
align-items	.c { align-items: stretch; }	(Padrão) Estica os itens para preencher o contêiner ao longo do eixo transversal.
align-content	.c { align-content: space-between; }	Alinha as <i>linhas</i> do contêiner (quando flex-wrap: wrap está ativo).
gap	.c { gap: 10px; }	Define o espaçamento entre itens (linhas e colunas).

3. Propriedades dos Itens Flex

Estas propriedades controlam o comportamento individual dos elementos filhos dentro do contêiner flex.

3.1. Ordenação e Alinhamento Individual

Propriedade	Exemplo de Declaração	Descrição
order	.item-3 { order: 1; }	Define a ordem visual do item (padrão é 0). Itens com order menor aparecem primeiro.
align-self	.item-1 { align-self: flex-end; }	Sobrescreve align-items do contêiner para alinhar um item específico no eixo transversal.

3.2. Flexibilidade e Dimensionamento

Controla como os itens crescem, encolhem e qual seu tamanho base.

Propriedade	Exemplo de Declaração	Descrição
flex-grow	.item { flex-grow: 1; }	Um fator numérico que indica o quanto o item deve crescer para ocupar o espaço extra disponível.
flex-shrink	.item { flex-shrink: 0; }	Um fator numérico que indica o quanto o item pode encolher se não houver espaço (padrão é 1).
flex-basis	.item { flex-basis: 150px; }	Define o tamanho inicial do item <i>antes</i> que o grow ou shrink sejam aplicados.
flex	.item { flex: 1 1 auto; }	<i>Shorthand</i> para flex-grow , flex-shrink e flex-basis .
flex	.item { flex: 1; }	(Comum) Equivalente a flex: 1 1 auto ; (cresce, encolhe, base automática).
flex	.item { flex: none; }	Equivalente a flex: 0 0 auto ; (não cresce nem encolhe).

Exemplos de Código Completos

Exemplo 1: Centrando Conteúdo

Um padrão comum é centralizar um elemento perfeitamente no meio da tela.

html

```
<div class="container-centro">
    <div class="item-centro">Conteúdo Centralizado</div>
</div>
```

css

```
.container-centro {
    display: flex;
    justify-content: center; /* Alinha horizontalmente */
    align-items: center;     /* Alinha verticalmente */
    height: 100vh;          /* Define a altura total da tela */
}

.item-centro {
    padding: 20px;
    background-color: lightblue;
}
```

Exemplo 2: Layout de Colunas Responsivas com **flex-wrap**

Criando um layout de galeria que se ajusta automaticamente.

html

```
<div class="galeria">
  <div class="foto">1</div>
  <div class="foto">2</div>
  <div class="foto">3</div>
  <div class="foto">4</div>
  <div class="foto">5</div>
</div>
```

css

```
.galeria {
  display: flex;
  flex-wrap: wrap; /* Permite que os itens quebrem para a próxima linha */
  gap: 10px;        /* Espaçamento entre as fotos */
}

.foto {
  /* flex-basis: 150px define um tamanho mínimo antes de distribuir espaço */
  /* flex-grow: 1 permite que eles cresçam para preencher a linha */
  flex: 1 1 150px;
  background-color: #f0f0f0;
  padding: 20px;
  text-align: center;
  min-width: 100px; /* Garante que não encolham muito */
}
```