

RESUMO DA AULA

Entendendo a Propriedade Flex

Nessa aula, você vai aprender sobre a propriedade **flex** e como ela combina as três propriedades do Flexbox: **flex-grow**, **flex-shrink** e **flex-basis** em uma única declaração.

Estrutura de Arquivos

Criamos uma pasta `10-flex`, contendo:

1. **HTML**:

- Um container `<div>` com a classe `container`.
- Três itens filhos `<div>` com a classe `flex-0`.

2. **globals.css**:

- Sem alterações em relação às aulas anteriores.

3. **style.css**:

- Inclui a configuração inicial do container e as classes de exemplo.

Os arquivos estão disponíveis para download na descrição da aula.

O que é a Propriedade Flex?

A propriedade **flex** é uma abreviação que permite combinar os valores das propriedades:

- **flex-grow**: Define a capacidade de crescimento de um item.
- **flex-shrink**: Define a capacidade de encolhimento de um item.
- **flex-basis**: Define o tamanho inicial de um item.

Ela simplifica a escrita, especialmente para valores comuns no dia a dia de desenvolvimento.

Como Funciona?

1. A Estrutura da Propriedade

A propriedade `flex` é composta por três valores:

```
```css
```

```
flex: <flex-grow> <flex-shrink> <flex-basis>;
```

```
```
```

2. Exemplo Prático

Criamos a seguinte classe:

```
```css
```

```
.flex-1 {
```

```
 flex: 0 1 150px;
```

```
}
```

```

- `0`: Representa o valor do **flex-grow**.
- `1`: Representa o valor do **flex-shrink**.
- `150px`: Representa o valor do **flex-basis**.

Implementação Passo a Passo

1. Configurando o Container

Primeiro, aplicamos o `display: flex` no container:

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

2. Aplicando o Flex nos Itens

Adicionamos a classe `.flex-1` nos itens do container:

```
```html
<div class="container">
 <div class="flex-1">Item 1</div>
 <div class="flex-1">Item 2</div>
 <div class="flex-1">Item 3</div>
</div>
```
```

...

Resultado:

- Cada item ocupará 150px inicialmente (definido pelo `flex-basis`).
- O tamanho será ajustado dinamicamente com base nos valores de `flex-grow` e `flex-shrink`.

Testando Diferentes Valores

1. Valor Padrão

Se omitirmos `flex-grow`, `flex-shrink` ou `flex-basis`, eles assumem os valores padrão:

```
```css
```

```
.flex-0 {
```

```
 flex: 0;
```

```
}
```

...

- `flex-grow`: `0` (não cresce).
- `flex-shrink`: `1` (pode encolher).
- `flex-basis`: `0` (sem tamanho inicial definido).

#### #### 2. Alterando Apenas o Grow

Podemos simplificar usando apenas o valor de `flex-grow`:

```
```css
```

```
.flex-1 {
```

```
    flex: 1;
```

```
}
```

...

Aqui:

- O item pode crescer para ocupar o espaço disponível.
- `flex-shrink` e `flex-basis` mantêm os valores padrão.

3. Combinando Valores

Podemos personalizar cada propriedade:

```
```css
```

```
.flex-custom {
 flex: 1 0 100px;
}
```

...

No HTML:

```
```html
```

```
<div class="container">  
    <div class="flex-custom">Item 1</div>  
    <div class="flex-0">Item 2</div>  
    <div class="flex-0">Item 3</div>  
</div>
```

...

Resultado:

- O **Item 1** cresce para ocupar espaço adicional, enquanto os outros permanecem no tamanho inicial.

Boas Práticas

1. Use apenas `flex: 1` para layouts simples:

- Ele já configura valores padrão para `flex-shrink` e `flex-basis`.

2. Personalize os valores apenas quando necessário:

- Por exemplo, `flex: 0 1 200px` para itens com tamanhos fixos.

Conclusão

Nessa aula, você aprendeu:

- A estrutura e os valores da propriedade `flex`.
- Como combinar `flex-grow`, `flex-shrink` e `flex-basis` em uma única declaração.
- Dicas práticas para usar o `flex` no dia a dia.

A propriedade `flex` é uma das mais usadas em layouts Flexbox, graças à sua simplicidade e flexibilidade.

Não esqueça de avaliar a aula e marcá-la como concluída. Até a próxima!
