

RESUMO DA AULA

Align Content e seus Valores

Nessa aula, você vai aprender sobre a propriedade **Align Content** e os seus principais valores. Além disso, vamos mostrar como essa propriedade se comporta de acordo com as regras do **Flex Direction**.

O que é Align Content?

A propriedade **Align Content** é usada para alinhar as linhas do container em relação ao eixo transversal (vertical, por padrão). Para que ela funcione, é necessário que:

1. O **Flex Wrap** esteja configurado como ``wrap`` ou ``wrap-reverse``.
2. O container tenha múltiplas linhas de **Flex Items**.
3. O container seja maior que a soma das linhas dos itens.

Se essas condições não forem atendidas, a propriedade **Align Content** não terá efeito.

Estrutura de Arquivos

Criamos uma pasta ``05-align-content`` com os seguintes arquivos:

- **index.html**: Estrutura HTML com sete containers representando cada valor do Align Content.
- **globals.css**: Estilos globais ajustados para suporte ao Align Content.
- **style.css**: Estilos específicos para demonstrar os valores do Align Content.

No HTML, cada container possui:

- Uma classe `container`.
- Uma classe específica para o valor do Align Content, como `flex-start`, `center`, etc.
- Dentro de cada container, há seis divs representando os itens (ou Flex Items).

Valores do Align Content

1. **Flex Start**

- Os itens são alinhados no início do eixo transversal.
- Exemplo:

```
```css
.flex-start {
 display: flex;
 align-content: flex-start;
 flex-wrap: wrap;
}
```

- Com `flex-direction: row` (padrão), os itens são alinhados ao topo do container.
- Com `flex-direction: column`, os itens são alinhados à esquerda do container.

---

#### #### 2. **Flex End**

- Os itens são alinhados no final do eixo transversal.
- Exemplo:

```
```css
```

```
.flex-end {  
  display: flex;  
  align-content: flex-end;  
  flex-wrap: wrap;  
}
```

```
```
```

- Com `flex-direction: row`, os itens são alinhados à base do container.
- Com `flex-direction: column`, os itens são alinhados à direita.

---

### #### 3. **Center**

- Os itens são centralizados ao longo do eixo transversal.
- Exemplo:

```
```css
```

```
.center {  
  display: flex;  
  align-content: center;  
  flex-wrap: wrap;  
}
```

```
```
```

- Pode ser combinado com `justify-content: center` para centralizar em ambos os eixos.

---

### #### 4. **Space Between**

- Espaça as linhas uniformemente, com o primeiro item no início e o último no final do eixo transversal.

- Exemplo:

```
```css
.space-between {
  display: flex;
  align-content: space-between;
  flex-wrap: wrap;
}
```

- Com `flex-direction: row`, cria espaço vertical uniforme entre as linhas.

5. **Space Around**

- Espaça as linhas de forma que cada uma tenha espaço ao redor de si.

- Exemplo:

```
```css
.space-around {
 display: flex;
 align-content: space-around;
 flex-wrap: wrap;
}
```

- O espaço entre as linhas é maior do que entre as linhas e as bordas do container.

---

#### #### 6. **\*\*Space Evenly\*\***

- Espaça as linhas de forma que o espaço entre elas e as bordas do container seja exatamente igual.

- Exemplo:

```
```css
.space-evenly {
  display: flex;
  align-content: space-evenly;
  flex-wrap: wrap;
}
```

```

#### #### 7. **\*\*Stretch\*\*** (Padrão)

- Estica as linhas para ocupar todo o espaço disponível no eixo transversal.

- Exemplo:

```
```css
.stretch {
  display: flex;
  align-content: stretch;
  flex-wrap: wrap;
}
```
```

- Para que funcione, os itens não devem ter altura ou largura fixa.

---

### ### Considerações Importantes

- **Align Content** trabalha no **eixo transversal**, enquanto **Justify Content** atua no **eixo principal**.
- A propriedade é útil em casos onde há múltiplas linhas de itens e queremos controlar seu alinhamento global dentro do container.
- Ela é uma fusão conceitual entre **Align Items** e **Justify Content**, mas funciona apenas com múltiplas linhas.

---

### ### Conclusão

Nessa aula, aprendemos:

- O que é o **Align Content** e como ele funciona.
- Como ele se relaciona com o **Flex Wrap** e o **Flex Direction**.
- Seus principais valores e como aplicá-los no layout.

Por hoje é isso! Não esqueça de avaliar e marcar a aula como concluída. Até mais!

---