Programação Web

Professor: Euclides Paim
euclidespaim@gmail.com

Fundamentos de CSS

Flexbox - Flexible Box Layout Module

Professor: Euclides Paim

euclidespaim@gmail.com

Fundamentos de CSS

Flexbox - Flexible Box Layout Module

Objetivos da Aula

- O que é *Flexbox*?
- *Flexbox* x *Grid*;
- *Flexbox Conteiner;*
- *Flex Items;*
- *Flex Responsive;*

Fundamentos de CSS

Flexbox - Flexible Box Layout Module

O que é *Flexbox*?

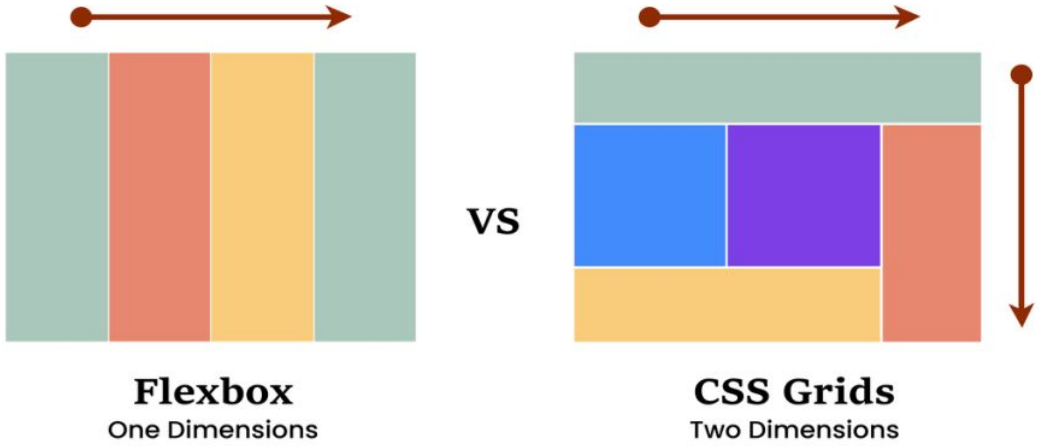
- O *Flexbox* é uma abreviação para o *Flex Box Layout Module* ou módulo de layout da caixa flexível.
- O *Flexbox* é um método de *layout* para organizar itens em linhas **ou** colunas.
- O *Flexbox* facilita o *design* de uma estrutura de *layout* responsiva flexível, sem usar flutuação ou posicionamento.

Fundamentos de CSS

Flexbox - Flexible Box Layout Module

Flexbox vs. Grid

- O *CSS Flexbox* deve ser usado para layout **unidimensional**, com linhas **ou** colunas.
- O *CSS Grid* deve ser usado para layout **bidimensional**, com linhas **e** colunas.



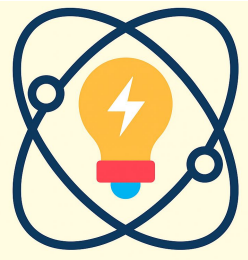
Fundamentos de CSS

Flexbox - Flexible Box Layout Module

Contexto Histórico

- Antes do *Flexbox* havia quatro modos de *layout*:
  a. *Block*, para seções em uma página da web
  b. *Inline*, para texto
  c. *Table*, para dados de tabela bidimensional
  d. *Position*, para posição explícita de um elemento

O CSS Flexbox é suportado em todos os navegadores modernos.



Fundamentos de CSS

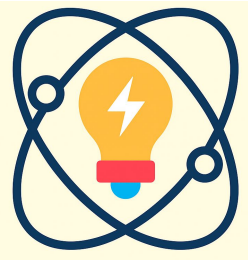
Flexbox - Flexible Box Layout Module

Componentes do CSS Flexbox

- Um Flexbox sempre consiste em:

  - Um ***Flex Container*** - o elemento pai (contêiner pai) `<div>`

  - E ***Flex Items*** - os itens dentro do contêiner (elementos filhos) `<div>`

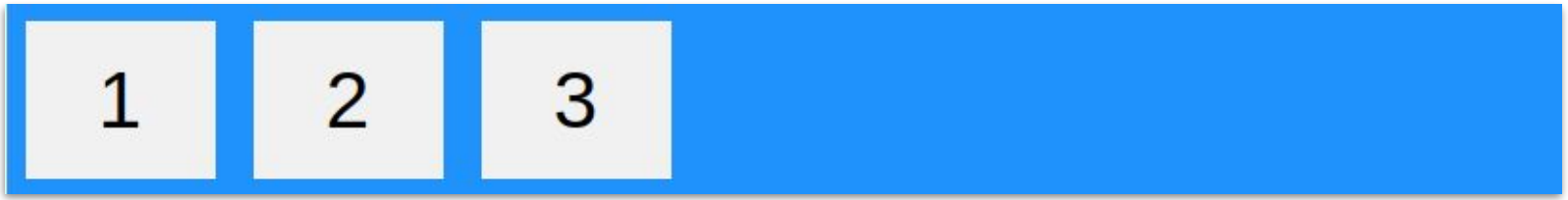


Fundamentos de CSS

Flexbox - Flexible Box Layout Module

Componentes do CSS Flexbox

- Para começar a usar o layout *CSS Flexbox*, precisamos primeiro definir um *Flex Conteiner*.
- O *flex container* se torna “flexível” quando setamos a propriedade `display` para o valor `flex`.



O elemento acima representa um ***Flex Conteiner*** (área em azul) contendo três ***Flex Items.***

Ex.: https://github.com/euclidespaim/ECS-301/blob/main/ProgramacaoWeb-301/Exemplos/6-flex/site-1/index.html

Fundamentos de CSS

Flexbox - Flexible Box Layout Module

As propriedades CSS que usamos para o contêiner Flex são:

- `flex-direction`
- `flex-wrap`
- `flex-flow`
- `justify-content`
- `align-items`
- `align-content`

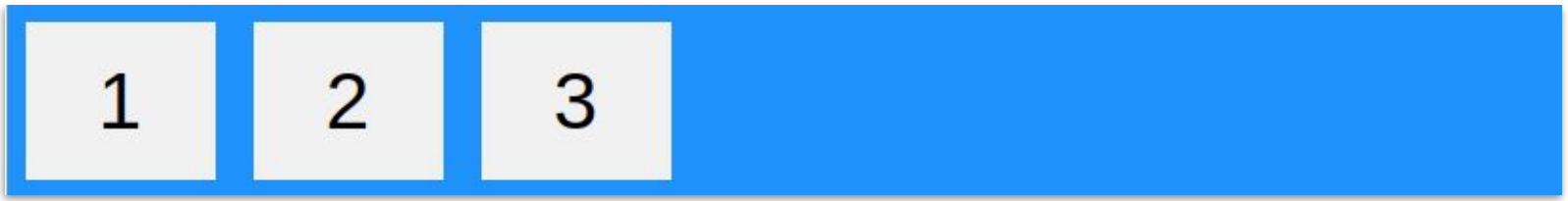
A propriedade *flex-direction*

A propriedade `flex-direction` especifica a direção de exibição de itens flexíveis no contêiner flex.

A propriedade `flex-direction` pode receber um dos seguintes valores:

- `row`
- `column`
- `row-reverse`
- `column-reverse`

O valor `row` é o valor padrão e exibe os itens flexíveis horizontalmente (da esquerda para a direita):



Fundamentos de CSS

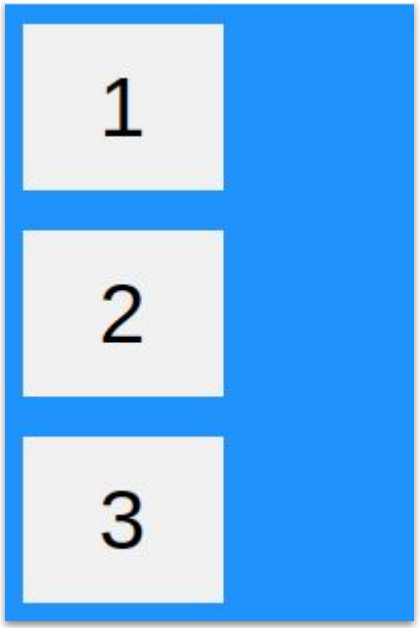
Flexbox - Flexible Box Layout Module

`flex-direction:`

- O valor `column` exibe os *flex items* verticalmente (de cima para baixo):

```
.flex-container {
  display: flex;
  flex-direction: column;
}
```

Resultado:



`flex-direction:`

- O valor `row-reverse` exibe os *flex items* horizontalmente (mas da direita para a esquerda):

```
.flex-container {
  display: flex;
  flex-direction: row-reverse;
}
```

Resultado:

Fundamentos de CSS

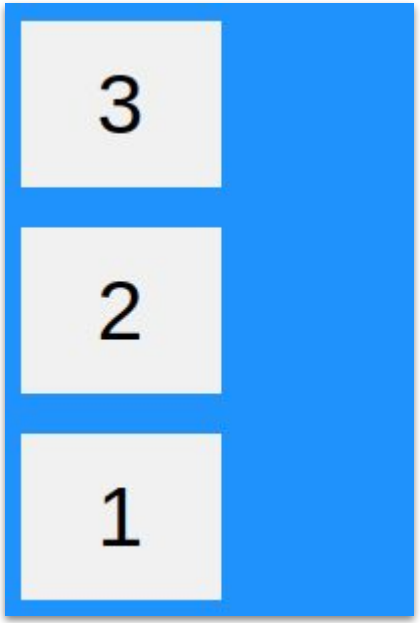
Flexbox - Flexible Box Layout Module

`flex-direction:`

- O valor `column-reverse` exibe os *flex items* verticalmente (de baixo para cima):

```
.flex-container {
  display: flex;
  flex-direction: column-reverse;
}
```

Resultado:



Fundamentos de CSS

Flexbox - Flexible Box Layout Module

`flex-wrap:`

A propriedade `flex-wrap` especifica se os *flex items* devem, ou não, “quebrar” se não houver espaço suficiente para eles em uma linha flexível.

A propriedade `flex-wrap` pode ter um dos seguintes valores:

- `nowrap`
- `wrap`
- `wrap-reverse`

Ex.: *https://github.com/euclidespaim/ECS-301/blob/main/ProgramacaoWeb-301/Exemplos/6-flex/site-2/index.html*

Fundamentos de CSS

Flexbox - Flexible Box Layout Module

flex-flow:

A propriedade `flex-flow` é uma propriedade abreviada para definir o `flex-direction` e `flex-wrap`.

```
.flex-container {
  display: flex;
  flex-flow: row wrap;
}
```

Fundamentos de CSS

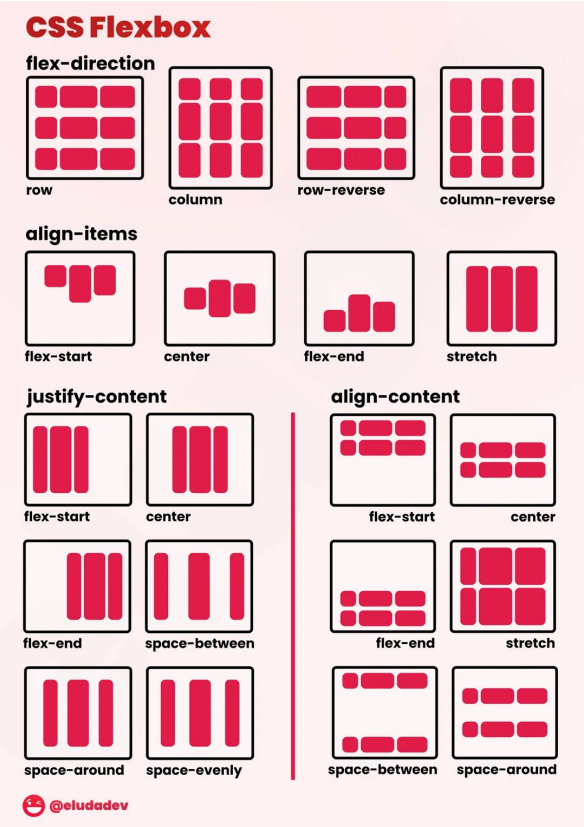
Flexbox - Flexible Box Layout Module

A propriedade *justify-content*

A propriedade `justify-content` é usado para alinhar os itens flexíveis quando eles não usam todo o espaço disponível no eixo principal (**horizontalmente**).

A propriedade `justify.content` pode receber um dos seguintes valores:

- `center`
- `flex-start`
- `flex-end`
- `space-around`
- `space-between`
- `space-evenly`



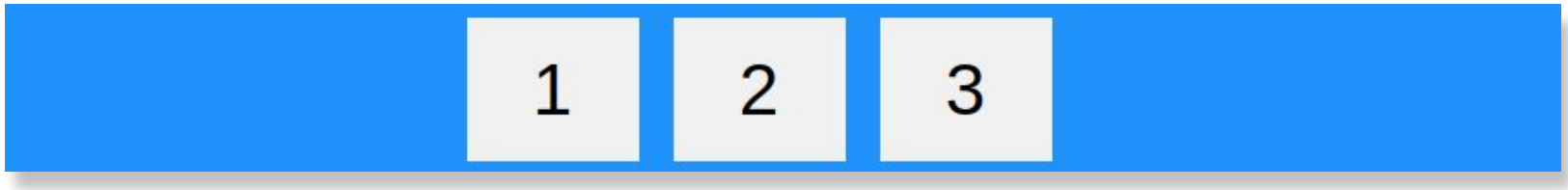
Fundamentos de CSS

Flexbox - Flexible Box Layout Module

`justify-content:`

O valor `center` posiciona os *flex items* no **centro** do contêiner.

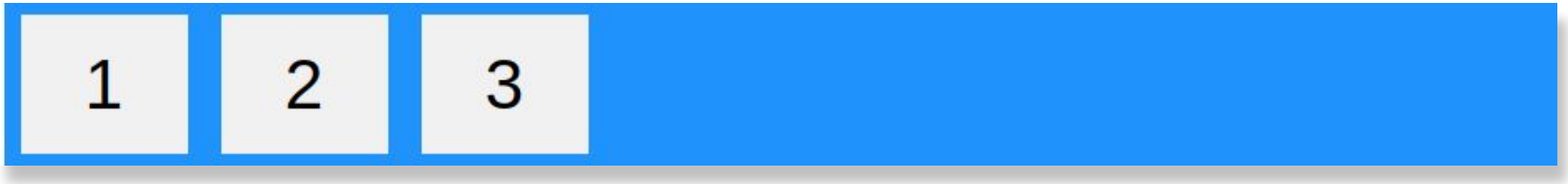
```css
.flex-container {
  display: flex;
  justify-content: center;
}
```



`justify-content:`

O valor `flex-start` posiciona os *flex items* no **início** do contêiner (esse é o valor padrão).

```css
.flex-container {
  display: flex;
  justify-content: flex-start;
}
```



`justify-content:`

O valor `flex-end` posiciona os *flex items* no **final** do contêiner (esse é o valor padrão).

```css
.flex-container {
  display: flex;
  justify-content: flex-end;
}
```

Fundamentos de CSS

Flexbox - Flexible Box Layout Module

`justify-content:`

O valor `space-around` exibe os *flex items* com espaço **ao redor** deles.

```css
.flex-container {
  display: flex;
  justify-content: space-around;
}
```

Fundamentos de CSS

Flexbox - Flexible Box Layout Module

`justify-content:`

O valor `space-between` exibe os *flex items* com espaço **entre** eles.

```css
.flex-container {
  display: flex;
  justify-content: space-between;
}
```

`justify-content:`

O valor `space-evenly` exibe os *flex items* com espaço **igual** entre eles.

```css
.flex-container {
  display: flex;
  justify-content: space-evenly;
}
```

Fundamentos de CSS

Flexbox - Flexible Box Layout Module

A propriedade *align-content*

A propriedade `align-content` é usado para **alinhar** os itens flexíveis.

A propriedade `align.content` é similar a propriedade `align.items` mas, em vez de alinhar os *flex items*, ela alinha as linhas flexíveis (*flex lines*).

A propriedade `align.content` pode ter um dos valores a seguir:

- `center`
- `stretch`
- `flex-start`
- `flex-end`
- `space-around`
- `space-between`
- `space-evenly`

Nos exemplos a seguir, usamos um contêiner de 600 pixels de altura, com a propriedade `flex-wrap` definida para `wrap`, para demonstrar melhor a propriedade `align-content`.

Fundamentos de CSS

Flexbox - Flexible Box Layout Module

`align-content:`

O valor `center` as linhas flexíveis são agrupadas em direção ao centro do contêiner.

```
.flex-container {
  display: flex;
  height: 600px;
  flex-wrap: wrap;
  align-content: center;
}
```

Fundamentos de CSS

Flexbox - Flexible Box Layout Module

`align-content:`

Com `stretch`, as linhas do flexbox se estendem para ocupar o espaço restante do contêiner (este é o valor padrão).

```css
.flex-container {
  display: flex;
  height: 600px;
  flex-wrap: wrap;
  align-content: stretch;
}
```

Fundamentos de CSS

Flexbox - Flexible Box Layout Module

`align-content:`

Com `space-evenly`, as linhas do flexbox são distribuídas uniformemente no contêiner flex, com espaço igual acima, abaixo e entre elas.

```
.flex-container {
  display: flex;
  height: 600px;
  flex-wrap: wrap;
  align-content: space-evenly;
}
```

Fundamentos de CSS

Flexbox - Flexible Box Layout Module

Perfeitamente centralizado

No exemplo a seguir, resolvemos um problema de estilo comum: centralização perfeita..

```css
.flex-container {
  display: flex;
  height: 300px;
  justify-content: center;
  align-items: center;
}
```

SOLUÇÃO: Defina as propriedades `justify-content` e `align-items` como `center`, e o *flex item* será centralizado perfeitamente.

Fundamentos de CSS

Flexbox - Flexible Box Layout Module

Os *Itens Flex* do CSS

Os elementos **filhos diretos** de um *contêiner flex* se tornam automaticamente *itens flex*.

O elemento acima representa quatro *flex items* azuis dentro de um *flex container* cinza.

Fundamentos de CSS

Flexbox - Flexible Box Layout Module

Os *Itens Flex* do CSS

As propriedades CSS que usamos para *flex items* são:

- `order`
- `flex-grow`
- `flex-shrink`
- `flex-basis`
- `flex`
- `align-self`

Fundamentos de CSS

Flexbox - Flexible Box Layout Module

A propriedade `order`

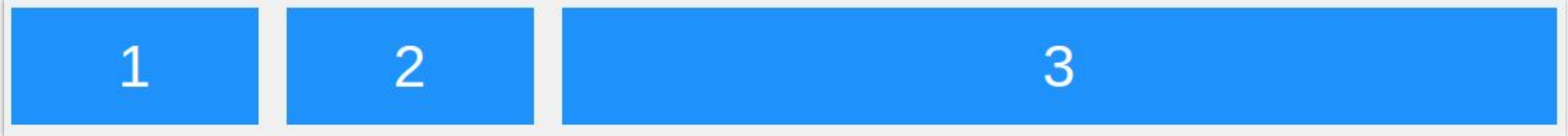
A propriedade `order` especifica a ordem dos *itens flexíveis* dentro do *contêiner flexível*:

```
<div class="flex-container">
  <div style="order: 3">1</div>
  <div style="order: 2">2</div>
  <div style="order: 4">3</div>
  <div style="order: 1">4</div>
</div>
```

A propriedade `flex-grow`

A propriedade `flex-grow` especifica quanto um *item flexível* crescerá em relação ao restante dos itens flexíveis:

```
<div class="flex-container">
  <div style="flex-grow: 1">1</div>
  <div style="flex-grow: 1">2</div>
  <div style="flex-grow: 8">3</div>
</div>
```



Fundamentos de CSS

Flexbox - Flexible Box Layout Module

A propriedade `flex-shrink`

A propriedade `flex-shrink` especifica quanto um *item flexível* **diminuirá** em relação ao restante dos itens flexíveis:

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div style="flex-shrink: 0">3</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
  <div>7</div>
</div>
```

A propriedade `flex-basis`

A propriedade `flex-basis` especifica o tamanho **inicial** de um *item flexível*:

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div style="flex-basis: 200px">3</div>
  <div>4</div>
</div>
```

Fundamentos de CSS

Flexbox - Flexible Box Layout Module

A propriedade `flex`

A propriedade `flex` é uma propriedade abreviada para as propriedades `flex-grow`, `flex-shrink` e `flex-basis`.

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div style="flex: 0 0 200px">3</div>
  <div>4</div>
</div>
```

Fundamentos de CSS

Flexbox - Flexible Box Layout Module

A propriedade `align-self`

A propriedade `align-self` especifica o alinhamento para o item selecionado dentro do recipiente flexível.

A propriedade `align-self` substitui o alinhamento padrão definido pela propriedade `align-items` do contêiner.

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div style="align-self: center">3</div>
  <div>4</div>
</div>
```

Fundamentos de CSS

Flexbox - Flexible Box Layout Module

Flexbox Responsivo

Aprendemos anteriormente sobre *Media Queries* em CSS que é possível usar essas consultas de mídia para criar diferentes *layouts* adaptados a diferentes tamanhos de tela e dispositivos.

Por exemplo, se quisermos criar um *layout* com **duas colunas** para a maioria dos tamanhos de tela e um layout com **uma coluna** para telas pequenas (como celulares e tablets), podemos alterar a propriedade `flex-direction` de `row` para `column` em um ponto de quebra específico (como `800px` por exemplo)

Outra forma é alterar a porcentagem da propriedade `flex` dos *itens flex* para criar diferentes *layouts* para diferentes tamanhos de tela. Observe que também é necessário incluir `flex-wrap: wrap;` no contêiner flex para que esse exemplo funcione corretamente.

Fundamentos de CSS

Flexbox - Flexible Box Layout Module

Revisão:

- O que é *Flexbox*?
- *Flexbox* x *Grid*;
- *Flexbox Conteiner;*
- *Flex Items;*
- *Flex Responsive;*

Referências

Referências Básicas

RESIG, John; BIBEAULT, Bear; MARZ, Josip. ***Secrets of the JavaScript Ninja***. 2. ed. Shelter Island: Manning Publications, 2016.

SILVA, Maurício Samy. HTML 5: a linguagem de marcação que revolucionou a web. São Paulo: Novatec, 2011.

GRONER, Loiane. ***Estrutura de dados e algoritmos com JavaScript:*** *escreva aplicações de JS modernas e performáticas utilizando estruturas de dados e algoritmos*. São Paulo: Novatec Editora, 2019.

Referências Complementares

FLANAGAN, David. **o guia definitivo**. . O Really. 2012

SILVA, Maurício Samy. **Criando sites com HTML: sites de alta qualidade com HTML e CSS**. . Novatec. 2010

IEPSEN, Edécio Fernandes. **Lógica de Programação e Algoritmos com JavaScript.** Novatec. 2018.

Referências na Internet

https://www.w3schools.com

https://developer.mozilla.org/pt-BR/docs/Web

https://illustrated.dev/advancedjs