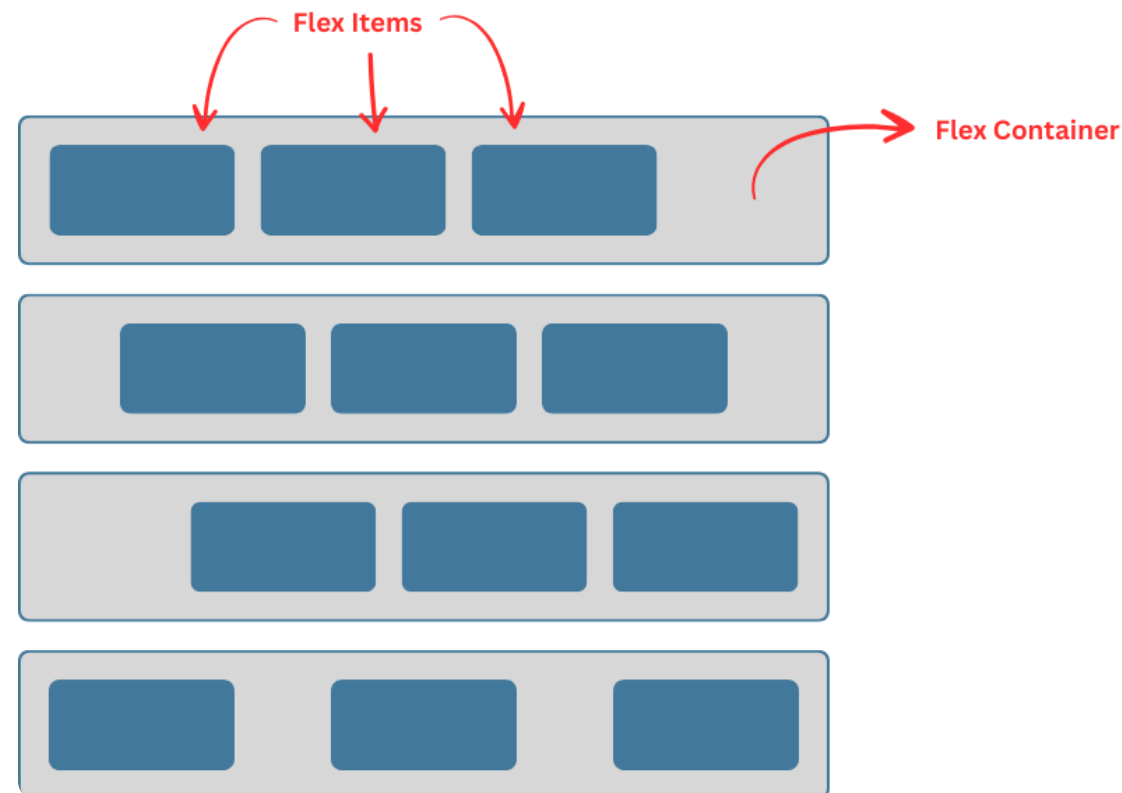


# Introdução ao Desenvolvimento Web

## CSS: Cascade Style Sheet

### 04 - CSS Flexbox



# Roteiro

---

- Introdução
- Elementos Básicos
- Propriedades Aplicadas ao Container
- Propriedades Aplicadas aos Itens
- Prática

# Flexbox

---

## Introdução e conceituação

- Houve um tempo em que posicionar elementos com CSS e estruturar o layout de um site ou sistema web era extremamente trabalhoso, dependendo do que se quisesse fazer.
- O **Flexbox** surgiu justamente para resolver esse problema.
- Antes dele, existiam basicamente quatro formas de trabalhar o layout de um site:
  - **Block** — para seções em uma página
  - **Inline** — para textos
  - **Tabela** — tabela de dados bidimensional
  - **Position e Float** — propriedades para posicionamento explícito e flutuação de elementos

# Flexbox

---

## Introdução e conceituação

- O **Flexbox** (*Flexible box*) é um modelo de layout que tem por objetivo **tornar mais eficiente** a criação de layouts, alinhamento e espaçamento entre itens de um container.
- Ele funciona muito bem, mesmo quando as dimensões dos itens são dinâmicas ou desconhecidas. Logo, podemos ter um layout com maior **flexibilidade**.
- Flexbox não é somente uma propriedade. Ele é um módulo CSS completo

# Flexbox

## Elementos básicos

- Para começar a trabalhar com Flexbox, vamos conhecer os elementos básicos que o compõe:
  - **Container Flex** (área em azul claro, abaixo): é o elemento pai, que envolve os itens flexíveis. Definimos um container flex com a propriedade `display` e os valores `flex` e `inline-flex`.
  - **Itens Flex** (elementos em azul escuro, abaixo): ao atribuir o valor `flex` à propriedade `display` no container, os filhos diretos destes se tornam, automaticamente, itens flexíveis.



*container flex*

# Flexbox

## Elementos básicos

- Eixos

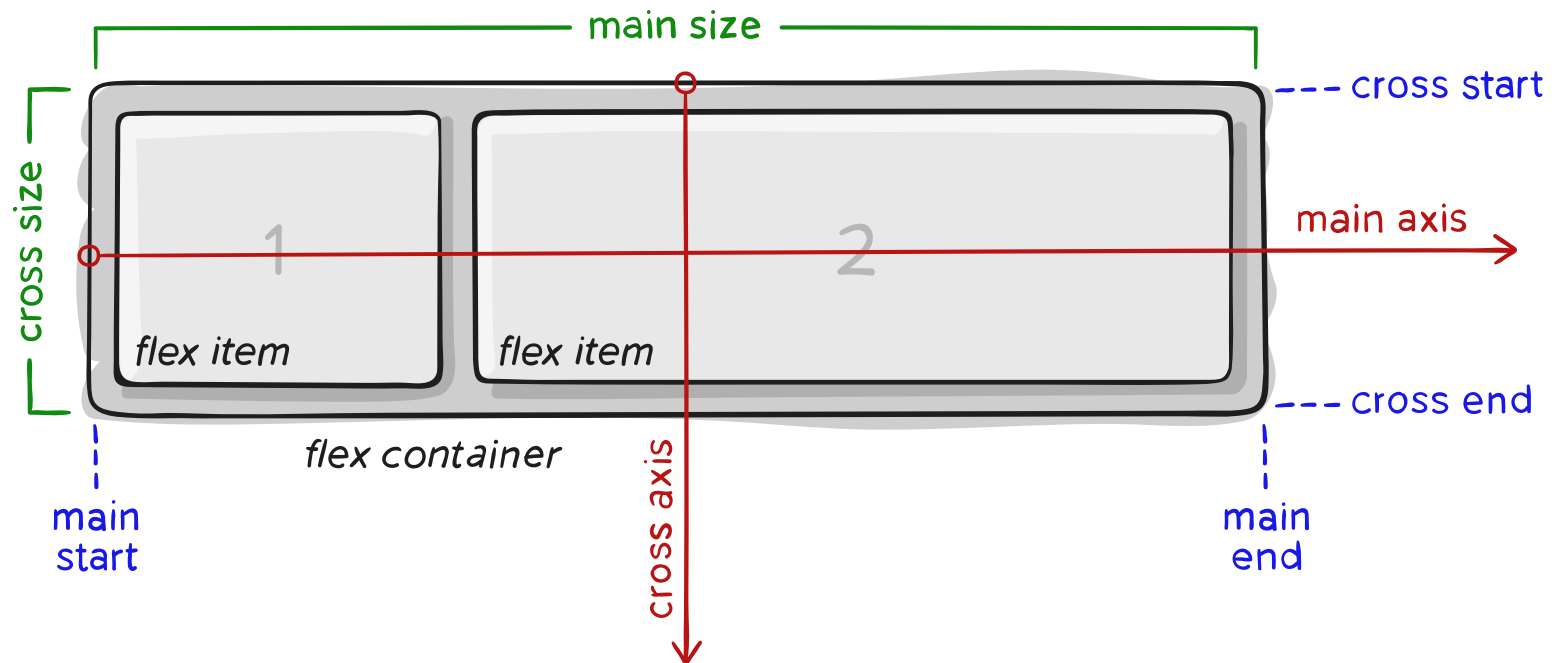
- O layout de todo container flex segue dois eixos: **eixo principal (main axis)** e **eixo transversal (cross axis)** perpendicular ao principal.
- Por padrão, o eixo principal é orientado na **horizontal**, enquanto o eixo transversal é orientado na **vertical**.

! É possível mudar esse comportamento padrão com a propriedade `flex-direction`.

# Flexbox

## Elementos básicos

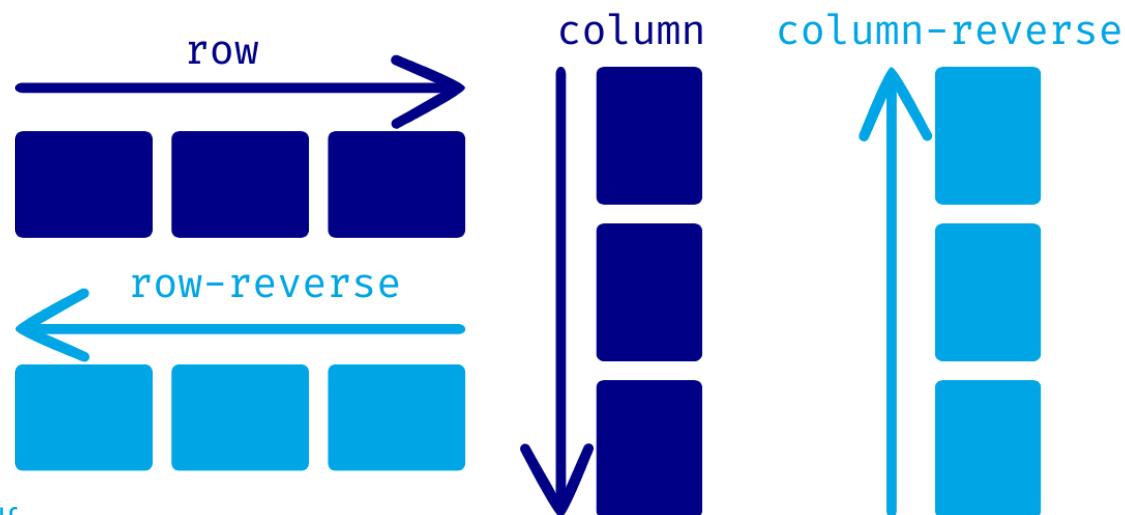
- Eixos



# Flexbox: Propriedades

## Propriedades aplicadas ao container

- **display** — define um **flex container** e coloca todos os elementos filhos diretos em um contexto **flex**
  - valores: `flex` | `inline-flex`
- **flex-direction** — estabelece a direção do eixo principal, sendo **row** o valor padrão
  - valores: `row` | `row-reverse` | `column` | `column-reverse`

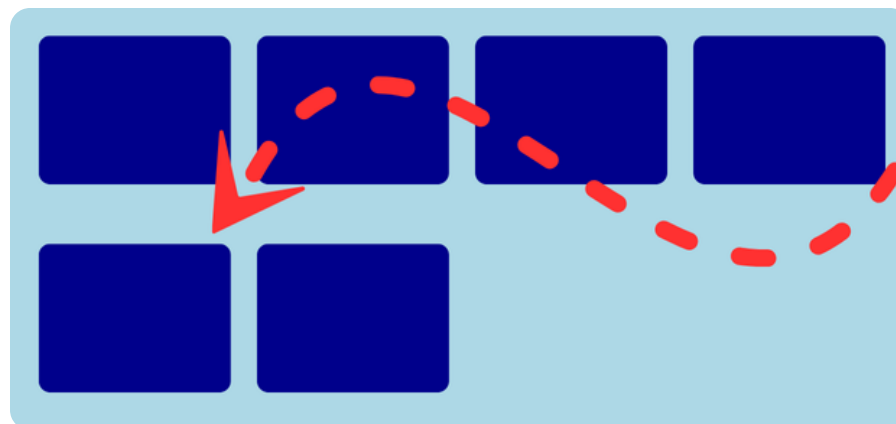




# Flexbox: Propriedades

## Propriedades aplicadas ao container

- **flex-wrap** — especifica se os itens podem ou não quebrarem para uma linha seguinte, caso necessário
  - valores: `wrap` | `nowrap` (padrão) | `wrap-reverse`



- **flex-flow** — um *shorthand* (atalho) para as propriedades `flex-direction` e `flex-wrap`
  - Por exemplo, se quisermos definir um container em coluna e permitir a quebra dos itens, poderíamos usar: `flex-flow: column wrap`

# Flexbox: Propriedades

---

Propriedades aplicadas ao **container**

- **Propriedades de Alinhamento:** o flexbox torna simples a manipulação do alinhamento dos itens e da distribuição do espaçamento entre eles, com as seguintes propriedades:
  - `justify-content` — define o alinhamento dos itens no eixo principal
  - `align-items` — define o alinhamento dos itens de acordo com o eixo transversal
  - `align-content` — controla o espaçamento entre as linhas no eixo transversal (só faz sentido caso o container possua mais de uma linha)

# Flexbox: Propriedades

## Propriedades aplicadas ao container

- Valores para a propriedade **justify-content**: `flex-start` | `flex-end` | `center` | `space-between` | `space-around` | `space-evenly`

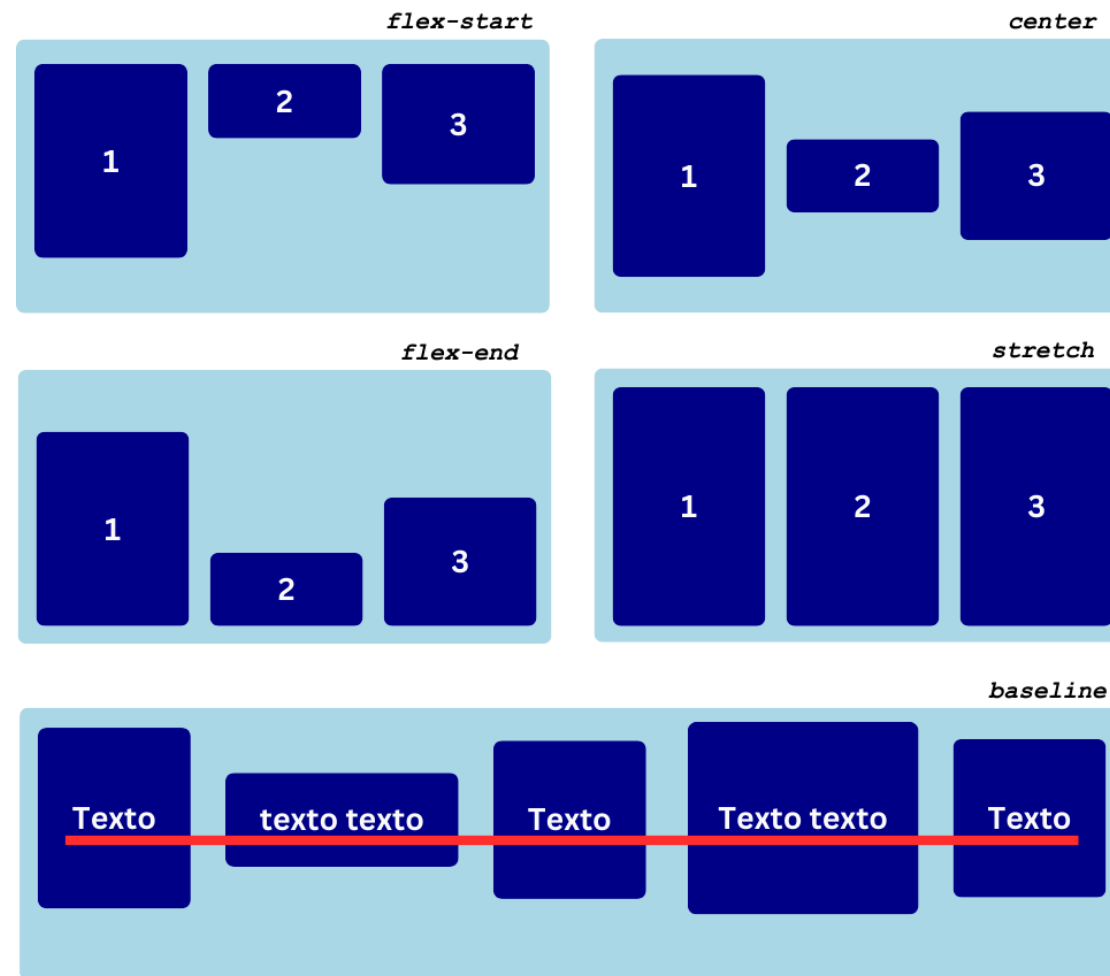
The diagram illustrates the `justify-content` property in Flexbox using six examples, each showing three numbered items (1, 2, 3) within a light blue container. The items are represented as dark blue rectangles with white numbers.

- flex-start**: The items are aligned to the start (left) of the container.
- flex-end**: The items are aligned to the end (right) of the container.
- center**: The items are centered within the container.
- space-between**: The items are distributed evenly, with space between them and the container edges.
- space-around**: The items are distributed evenly, with space between them and the container edges, and the space between items is equal to the space between the items and the container edges.
- space-evenly**: The items are distributed evenly, with space between them and the container edges, and the space between items is equal to the space between the items and the container edges, and the space between the items and the container edges is equal to the space between the items and the container edges.

# Flexbox: Propriedades

Propriedades aplicadas ao container

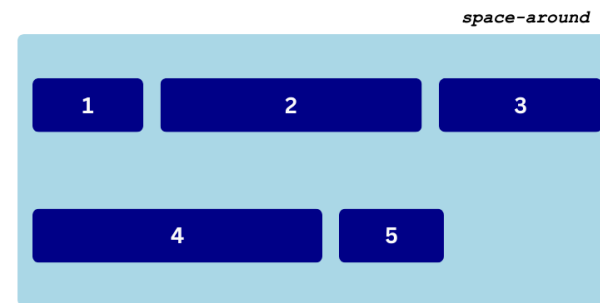
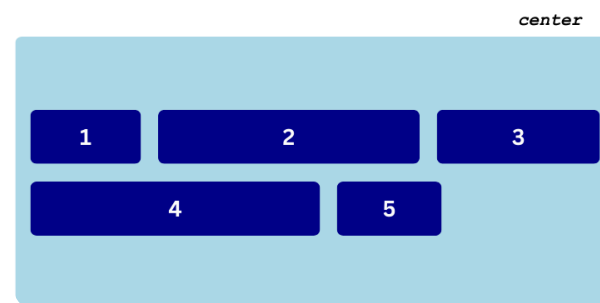
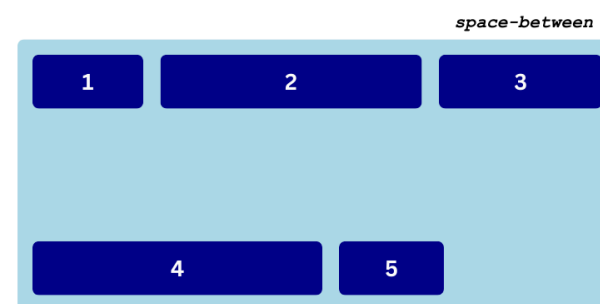
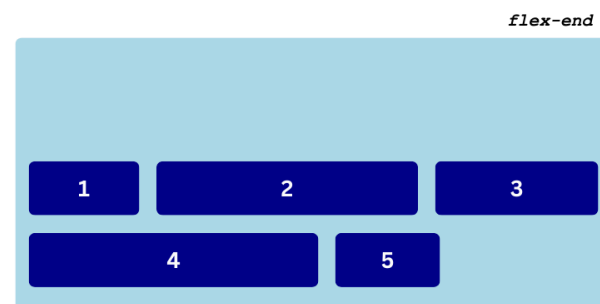
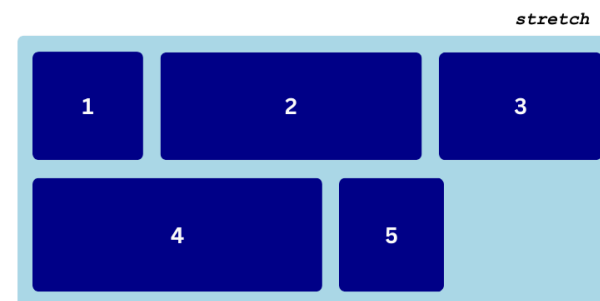
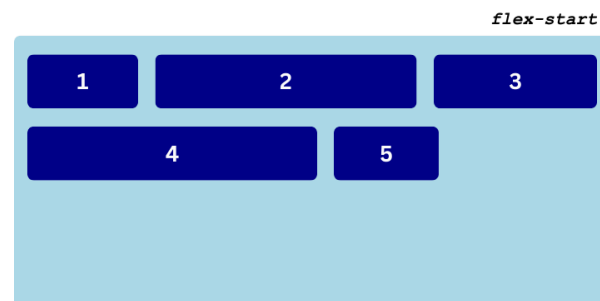
- Valores para a propriedade `align-items`:
  - `flex-start`
  - `flex-end`
  - `center`
  - `stretch`
  - `baseline`



# Flexbox: Propriedades

## Propriedades aplicadas ao container

- Valores para a propriedade `align-content` :
  - `flex-start`
  - `flex-end`
  - `center`
  - `stretch`
  - `space-between`
  - `space-around`
  - `space-evenly`



# Flexbox: Propriedades

## Propriedades aplicadas aos itens flexíveis

- **order** — especifica a ordem dos itens flexíveis no container.
  - Por padrão, os itens são dispostos no container na ordem em que foram definidos no código
  - Usando a propriedade **order** podemos mudar a ordem em que um item aparece no container
  - **Sintaxe:**

```
.flex-item {  
  order: <número>; /** padrão é 0 (zero) */  
}
```

# Flexbox: Propriedades

---

## Propriedades aplicadas aos itens flexíveis

- **flex-grow** — define a flexibilidade de crescimento de um item, caso necessário.
  - Recebe um valor numérico sem unidade de medida, que especifica a quantidade proporcional de espaço disponível no container que será ocupado pelo item
  - Por exemplo, em um container com dois elementos, especificando um valor 1 (um) para o primeiro item e o valor 2 (dois) para o segundo item, este último crescerá duas vezes mais (utilizará duas vezes mais espaço disponível do que o primeiro)
  - O valor padrão é 0 (zero), que indica que o item não crescerá, mesmo tendo espaço disponível

# Flexbox: Propriedades

## Propriedades aplicadas aos itens flexíveis

- **flex-shrink** — define a flexibilidade de encolhimento de um item, caso necessário.
  - Similar ao **flex-grow**, porém definindo a habilidade de redução do tamanho do item
  - O valor padrão é 1 (um), permitindo que os itens possam ter seu tamanho reduzido para caber no container
- **flex-basis** — define o tamanho inicial de um item, antes que o espaço restante no container seja distribuído
- **flex** — *shorthand* para **flex-grow**, **flex-shrink** e **flex-base**
  - O segundo e o terceiro parâmetro são opcionais ( **flex-shrink** e **flex-basis** )
  - O padrão é **0 1 auto**



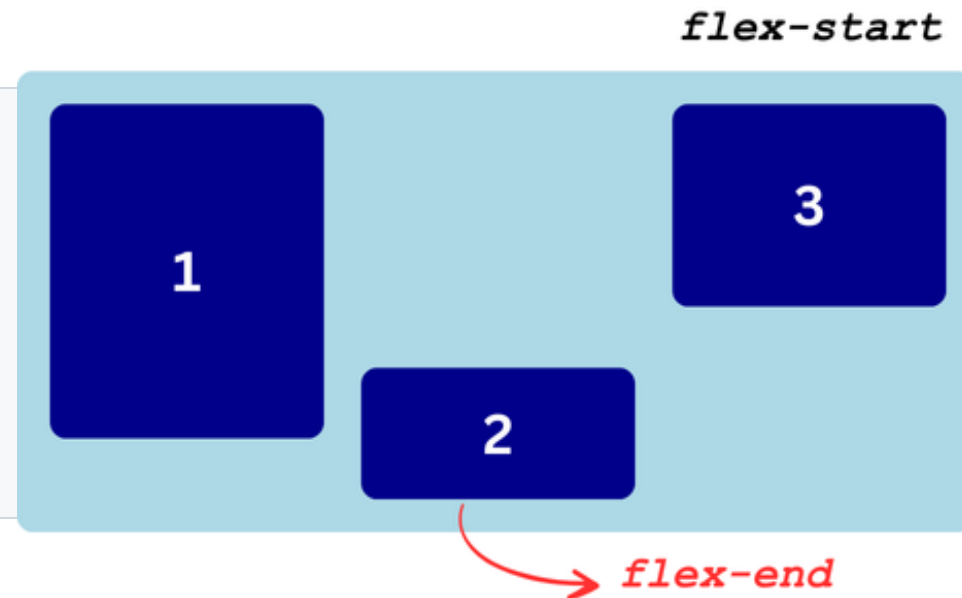
# Flexbox: Propriedades

## Propriedades aplicadas aos itens flexíveis

- **align-self** — especifica o alinhamento de um item selecionado dentro do container, sobrescrevendo o que foi especificado de forma geral com a propriedade **align-items** no container

◦ Exemplo:

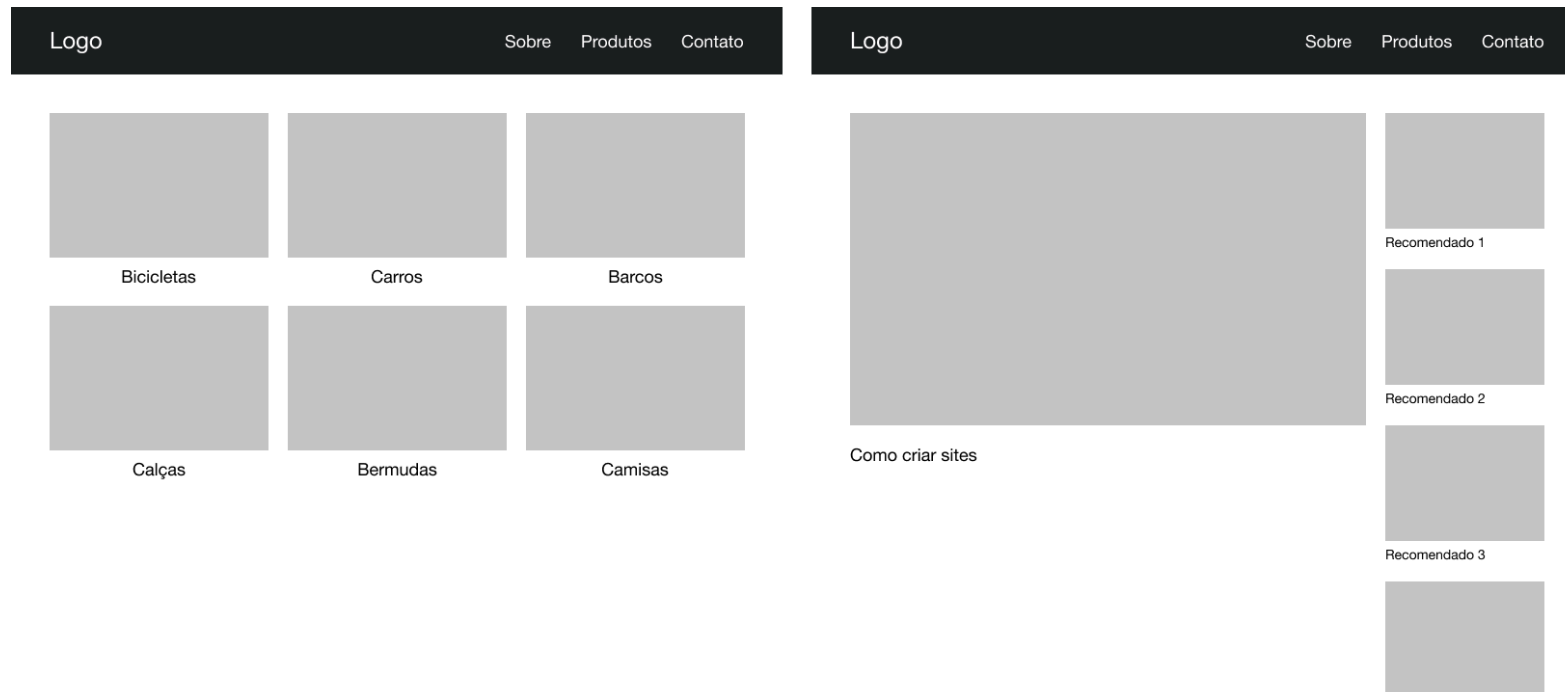
```
.container {  
  display: flex;  
  align-items: flex-start;  
}  
  
.item-2 {  
  align-self: flex-end;  
}
```



# Flexbox

Vamos praticar um pouco?!

- Usando Flexbox, vamos tentar construir os dois layouts a seguir.



# Isso é tudo, pessoal!

Sigam-me nas redes abaixo 📍 😊

 @lucas-lfm

 @prof\_lucasmendes

 @prof-lucasmendes

Email: [lucas.mendes@ifce.edu.br](mailto:lucas.mendes@ifce.edu.br)

Prof. Me. Lucas Mendes | GitHub: [@lucas-lfm](https://github.com/lucas-lfm)

Introdução ao Desenvolvimento Web 🧑💻

# Referências e Conteúdos Complementares

---

- W3Schools. CSS Flexbox. Disponível em: [https://www.w3schools.com/css/css3\\_flexbox.asp](https://www.w3schools.com/css/css3_flexbox.asp)
- MDN Web Docs. Flexbox. Disponível em: [https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS\\_layout/Flexbox](https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox)
- Alura. Flexbox CSS: Guia Completo, Elementos e Exemplos. Disponível em: <https://www.alura.com.br/artigos/css-guia-do-flexbox>
- Origamid. Guia Flexbox. Disponível em: <https://origamid.com/projetos/flexbox-guia-completo/>