

CSS FlexBox Layout



Módulo FlexBox

O **FlexBox** torna mais fácil **projetar uma estrutura de layout responsiva e flexível** sem usar flutuação (**float**) ou posicionamento.

Ele é descrito como **unidimensional**, já que ele **lida com o layout em uma dimensão de cada vez**, seja uma **linha** ou uma **coluna**.



A ideia principal é dar ao contêiner a **capacidade de alterar a largura/altura (e ordem) de seus itens** para **melhor preencher o espaço disponível**.

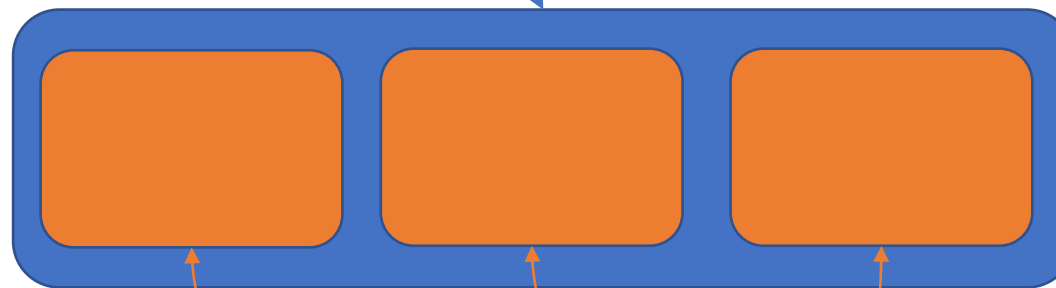
Principalmente para acomodar **todos os tipos de dispositivos de exibição e tamanhos de tela**.



Terminologia básica

Como o **Flexbox** é um **módulo inteiro** e **não uma única propriedade**, ele envolve muitas coisas, incluindo todo o seu **próprio conjunto de propriedades**.

flex container



Algumas propriedade devem ser definidas no elemento pai, ou "**flex container**".

flex
items

Outras propriedades devem ser definidas nos filhos, ou "**flex items**".



Eixos do FlexBox

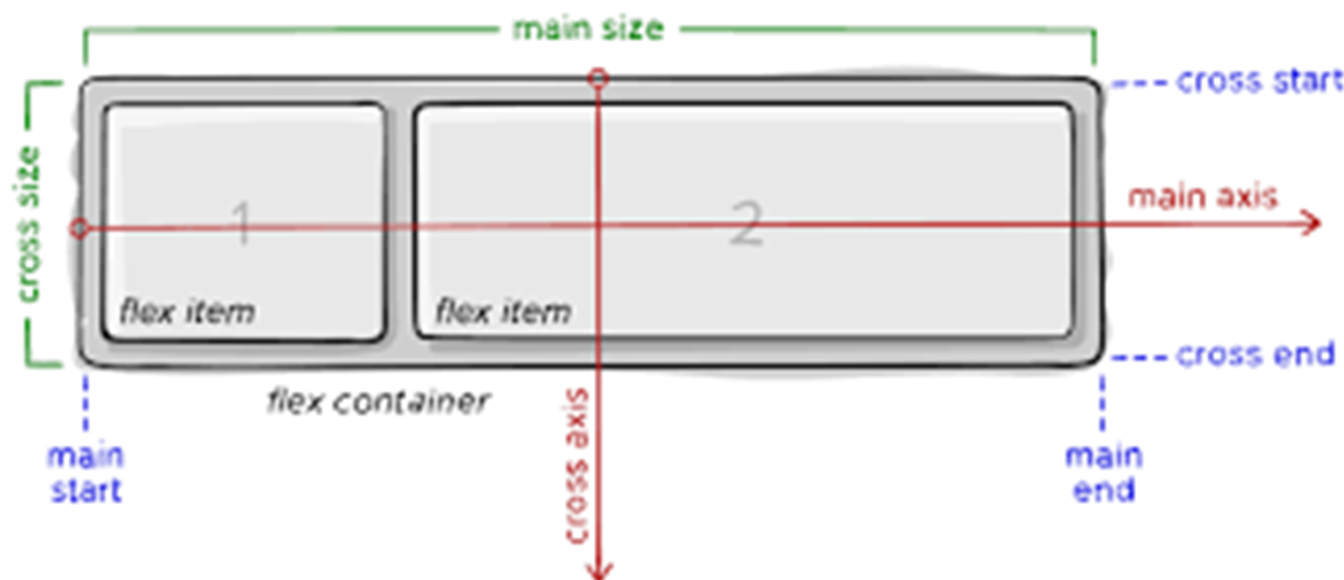
Todas as operações realizadas relacionam-se a **dois eixos**:

main-axis (principal)

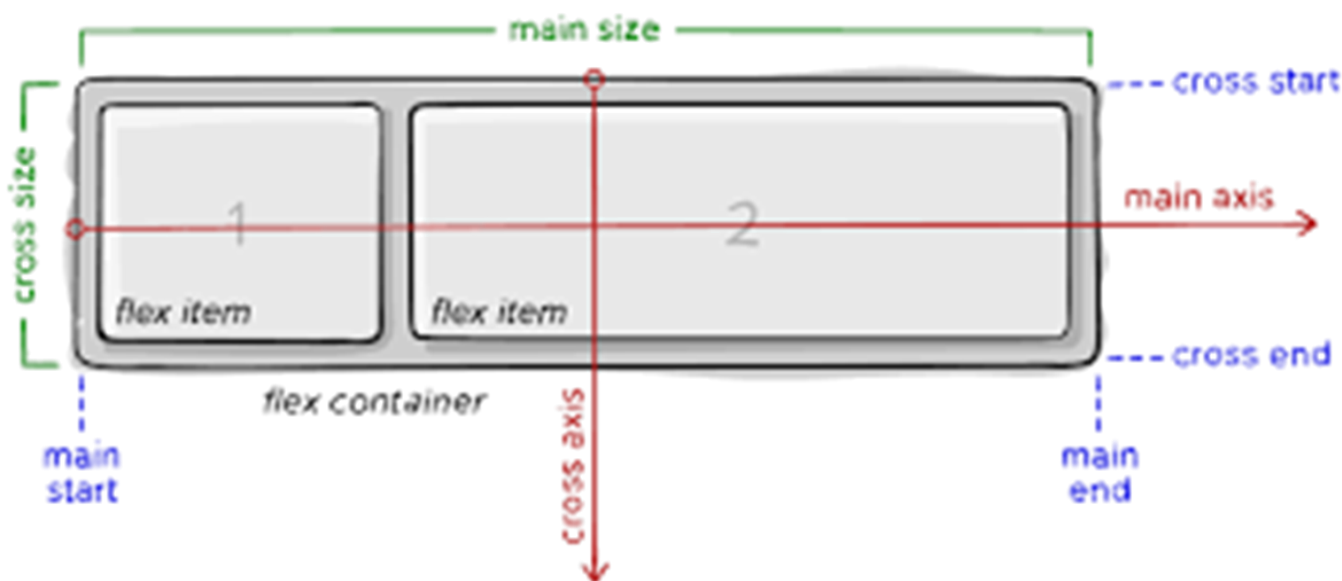
Definido pela propriedade **flex-direction**.

cross-axis (transversal)

Encontra-se na direção **perpendicular** ao eixo principal.



Os *flex items* serão dispostos seguindo ou o eixo principal (do **main-start** ao **main-end**) ou o eixo transversal (do **cross-start** ao **cross-end**).



main axis - eixo principal de um *flex container*, ou seja, o **eixo primário no qual os *flex items* são dispostos**. Não é necessariamente horizontal, **já que depende da propriedade *flex-direction***.

main-start | main-end - Os *flex items* são colocados dentro do contêiner começando do **main-start** e indo para o **main-end**.

main size - a **largura ou altura de um *flex item***. O que estiver na dimensão principal, é o tamanho principal do item.

cross axis - O eixo perpendicular ao eixo principal é chamado de **eixo transversal** (ou cruzado). **Sua direção depende da direção do eixo principal.**

cross-start | cross end - as **linhas flexíveis** são preenchidas com itens e colocadas no contêiner, começando no lado **cross-start** do *flex container* indo em direção ao lado da **cross-end**.

cross size - A largura ou altura de um *flex item*. O que estiver na dimensão **cross**, é o tamanho do item transversal.

CSS

Flex Container





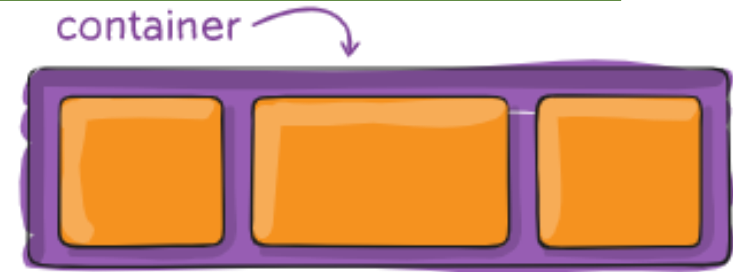
Flex container

Para utilizar o FlexBox, é preciso definir um **flex container**.

Um Layout Flexível deve ter um elemento pai com a propriedade **display** definida para **flex**

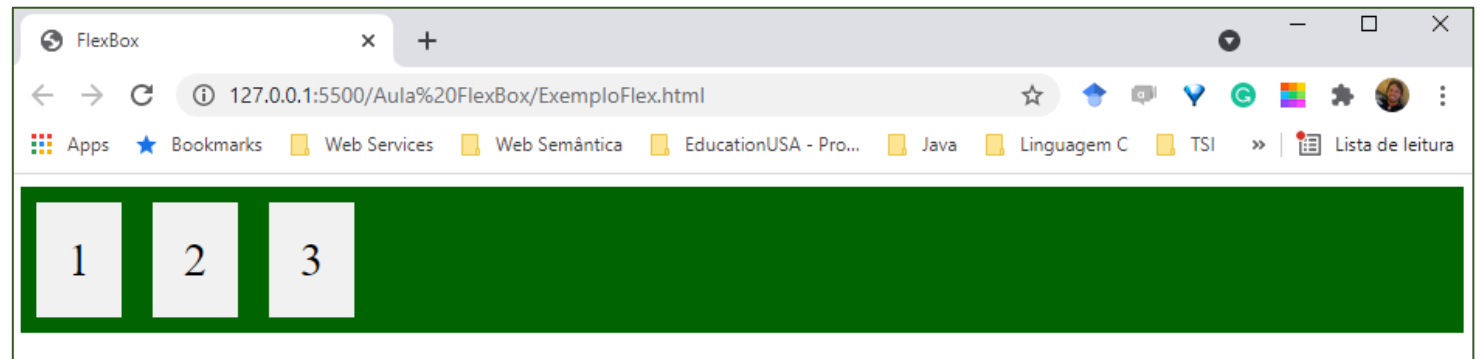
```
<body>  
  <div class="flex-container">  
    <div>1</div>  
    <div>2</div>  
    <div>3</div>  
  </div>  
</body>
```

Elementos filhos do *flex container*
tornam-se automaticamente *flex items*



```
.flex-container {  
  display: flex;  
  background-color: darkgreen;  
}
```

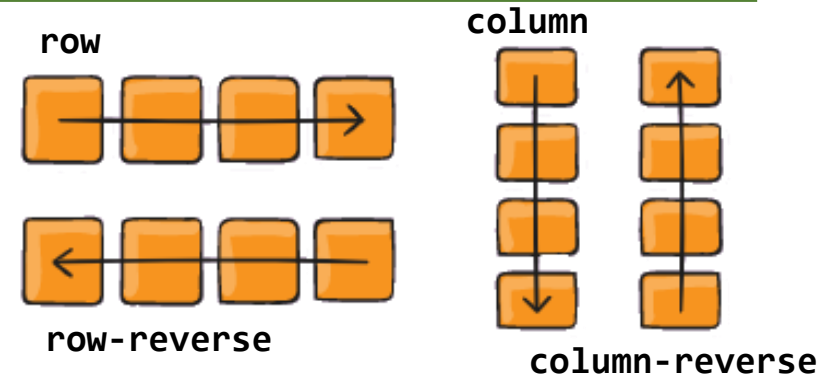
```
.flex-container > div {  
  background-color: #f1f1f1;  
  margin: 10px;  
  padding: 20px;  
  font-size: 30px;  
}
```





Flex container

```
.flex-container {
  display: flex;
  background-color: darkgreen;
}
```



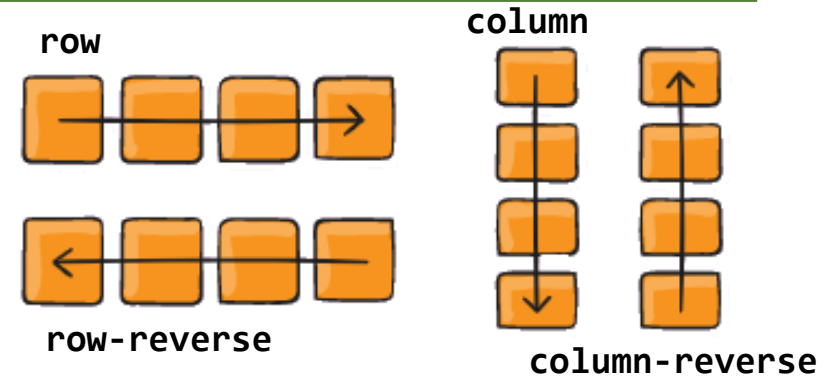
Propriedades

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



Flex container: flex-direction

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

[Exemplo](#)

Propriedades

flex-direction

flex-wrap

flex-flow

justify-content

align-items

align-content

flex-direction define em qual direção o contêiner deseja empilhar os *flex items*.

column: empilha os *flex items* verticalmente (de cima para baixo);

column-reverse: empilha os *flex items* verticalmente (de baixo para cima);

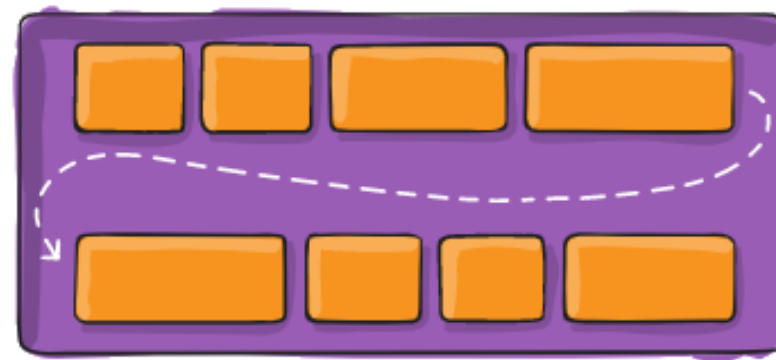
row: empilha os *flex items* horizontalmente (da esquerda para a direita);

row-reverse: empilha os *flex items* horizontalmente (da direita para a esquerda).



Flex container: flex-wrap

```
.flex-container {  
  display: flex;  
  background-color: darkgreen;  
  flex-wrap: wrap;  
}
```



[Exemplo](#)

Propriedades

flex-direction

flex-wrap

flex-flow

justify-content

align-items

align-content

flex-wrap especifica se os *flex items* devem quebrar ou não as linhas.

Por padrão, todos os itens flexíveis tentarão caber em uma linha. Essa propriedade permite alterar isso para permitir que os itens sejam agrupados conforme desejado.

wrap: especifica que os *flex items* irão quebrar linhas se necessário;

nowrap: especifica que os *flex items* não serão quebrados em linhas (**padrão**);

wrap-reverse: especifica que os *flex items* serão agrupados, se necessário, na ordem inversa:



Flex container: flex-flow

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

Propriedades

flex-direction

flex-wrap

flex-flow

justify-content

align-items

align-content

flex-flow define um **atalho** para definir ambas as propriedades:

flex-direction; e

flex-wrap.

[Exemplo](#)



Flex container: justify-content

```
.flex-container {  
  display: flex;  
  background-color: darkgreen;  
  justify-content: center;  
}
```

Propriedades

flex-direction

flex-wrap

flex-flow

justify-content

align-items

align-content

justify-content é usada para alinhar os *flex items*.

flex-start: alinha os *flex items* no início do contêiner (padrão);

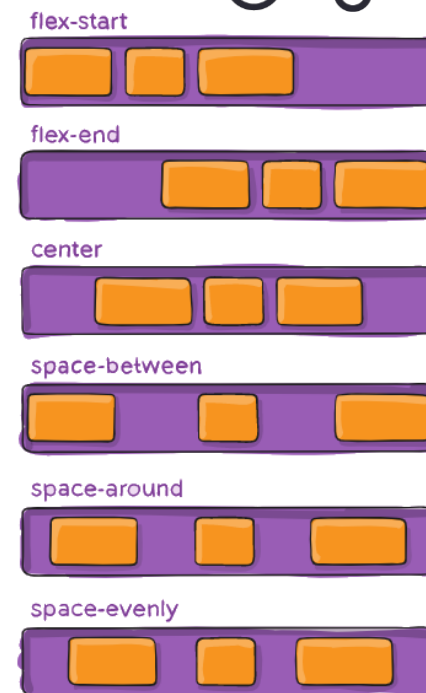
flex-end: alinha os *flex items* no final do contêiner;

center: alinha os *flex items* no centro do contêiner;

space-between: exibe os *flex items* com espaço entre as linhas

space-around: exibe os *flex items* com espaço antes, entre e depois das linhas

space-evenly: *flex items* são distribuídos uniformemente dentro do contêiner



[Exemplo](#)

Flex container: align-items



```
.flex-container {  
  display: flex;  
  background-color: darkgreen;  
  align-items: flex-start;  
}
```

Propriedades

flex-direction

flex-wrap

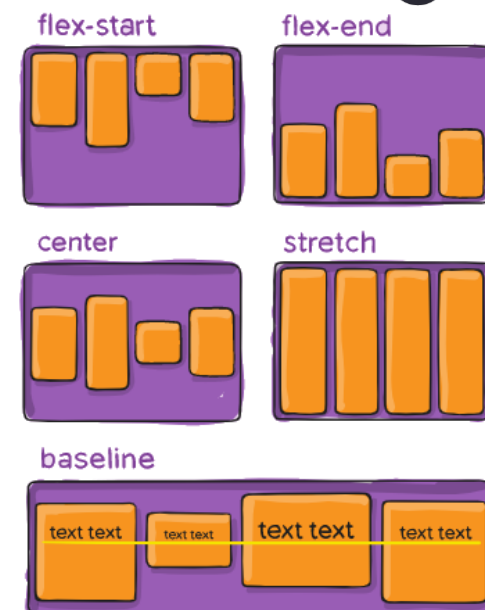
flex-flow

justify-content

align-items

align-content

align-items é usada para alinhar os *flex items*.



flex-start: valor alinha os *flex items* na parte superior do contêiner;

flex-end: valor alinha os *flex items* na parte inferior do contêiner;

center: valor alinha os *flex items* no meio do contêiner;

stretch: estica os *flex items* para preencher o contêiner (padrão);

baseline: alinha os *flex items*, como os alinhamentos de suas linhas de base.



Flex container: align-content

```
.flex-container {  
  display: flex;  
  background-color: darkgreen;  
  align-content: space-between;  
}
```

Propriedades

flex-direction

flex-wrap

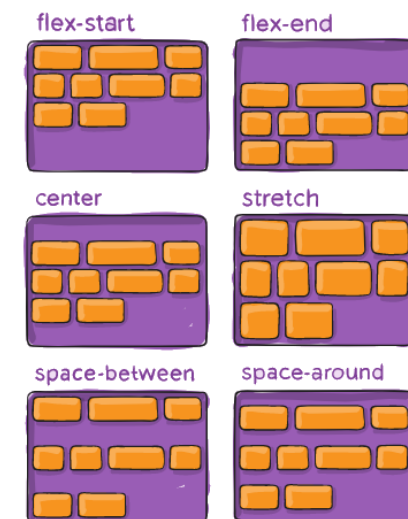
flex-flow

justify-content

align-items

align-content

align-content é usada para **alinhar** as linhas flex.



space-between: exibe as linhas flexíveis com espaço igual entre elas;

space-around: exibe as linhas flexíveis com espaço antes, entre e depois delas;

stretch: alonga as linhas flexíveis para ocupar o espaço restante (**padrão**);

center: exibe as linhas flexíveis no meio do contêiner;

flex-start: exibe as linhas flexíveis no início do contêiner;

flex-end: exibe as linhas flexíveis no final do contêiner;

[Exemplo](#)

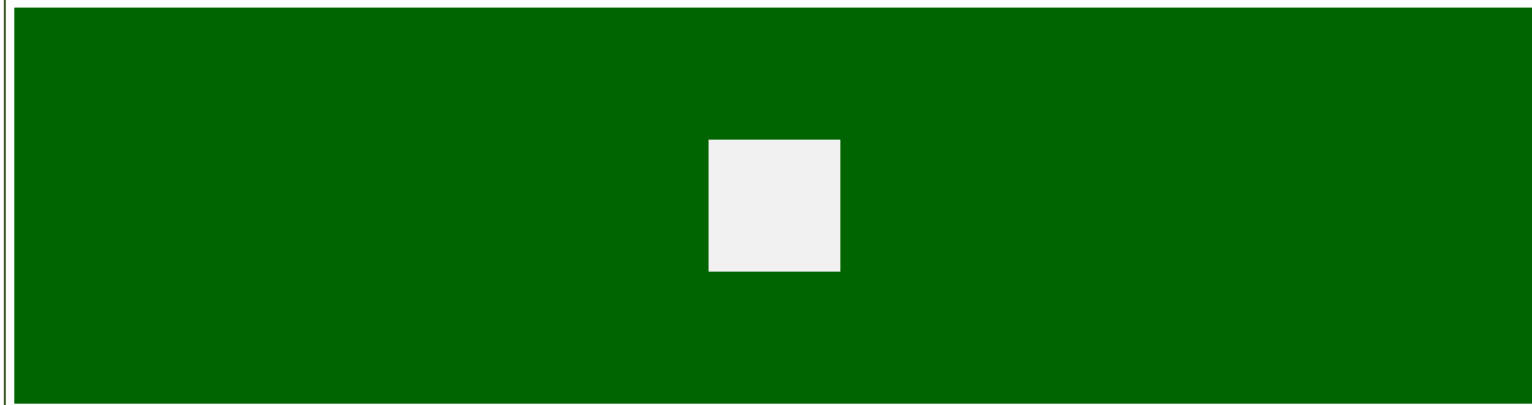


Centralização perfeita com flex

Combinação das propriedades **justify-content** e **align-items** para **center** e o *flex item* ficará centralizado:

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

Centralização Perfeita



[Exemplo](#)



Flex container: gap

```
.flex-container {  
  display: flex;  
  background-color: darkgreen;  
  gap: 30px;  
}
```

Propriedades

flex-direction

flex-wrap

flex-flow

justify-content

align-items

align-content

gap

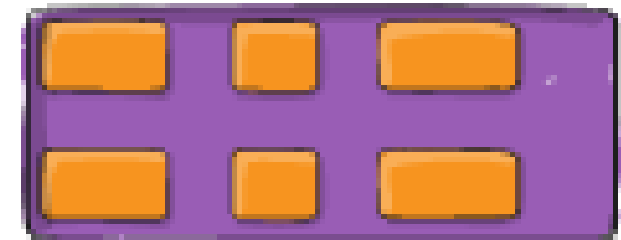
gap controla explicitamente o espaço entre os itens flexíveis.

É uma abreviação para **row-gap** e **column-gap**, especificando o tamanho do espaço entre as linhas e colunas nos layouts de grid, flex e multi-colunas.

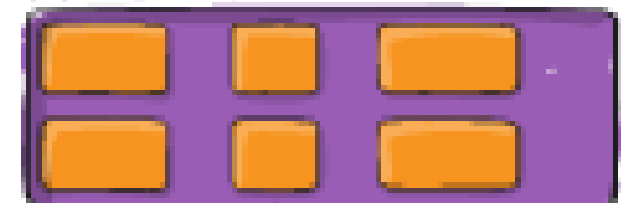
gap: 10px



gap: 30px



gap: 10px 30px



[Exemplo](#)



Propriedades do Flex container

Propriedade	Descrição
align-content	Modifica o comportamento da propriedade <code>flex-wrap</code> . É semelhante a itens de alinhamento, mas em vez de alinhar itens flexíveis, ele alinha as linhas flexíveis
align-items	Alinha verticalmente os <i>flex items</i> quando eles não usam todo o espaço disponível no eixo transversal.
display	Especifica o tipo de box usado para um elemento HTML
flex-direction	Especifica a direção dos <i>flex items</i> dentro de um <i>flex container</i>
flex-flow	Propriedade abreviada para <code>flex-direction</code> e <code>flex-wrap</code>
flex-wrap	Especifica se os <i>flex items</i> devem ser agrupados ou não, se não houver espaço suficiente para eles em uma linha flexível
justify-content	Alinha horizontalmente os <i>flex items</i> quando os eles não usam todo o espaço disponível no eixo principal

CSS

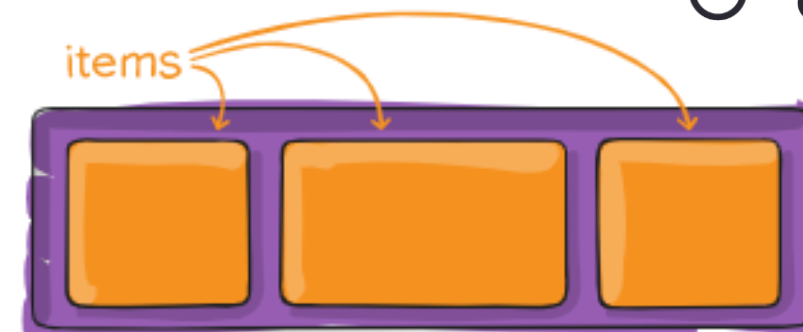
Flex items



Flex items



Elementos filhos diretos de um **flex container** automaticamente se tornam **flex items**.

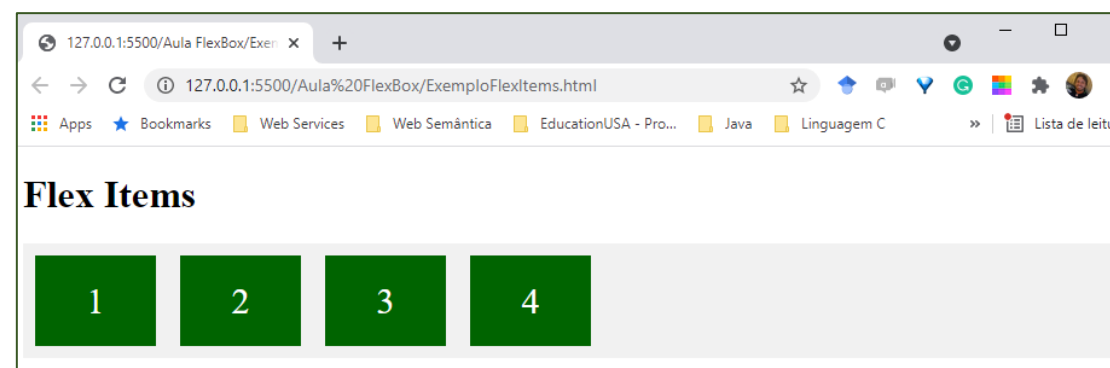


```
.flex-container {
  display: flex;
  background-color: #f1f1f1;
}
```

```
.flex-container > div {
  background-color: darkgreen;
  color: white;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
```

[Exemplo](#)

```
<body>
  <div class="flex-container">
    <div>1</div>
    <div>2</div>
    <div>3</div>
    <div>4</div>
  </div>
</body>
```



Flex items



Propriedades

`order`

`flex-grow`

`flex-shrink`

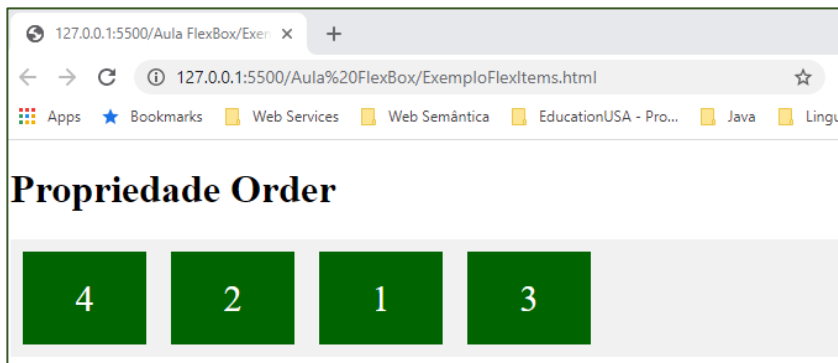
`flex-basis`

`flex`

`align-self`



Flex items: order



O primeiro flex item no código **não precisa necessariamente aparecer como o primeiro item no layout.**

[Exemplo](#)

Propriedades

order

flex-grow

flex-shrink

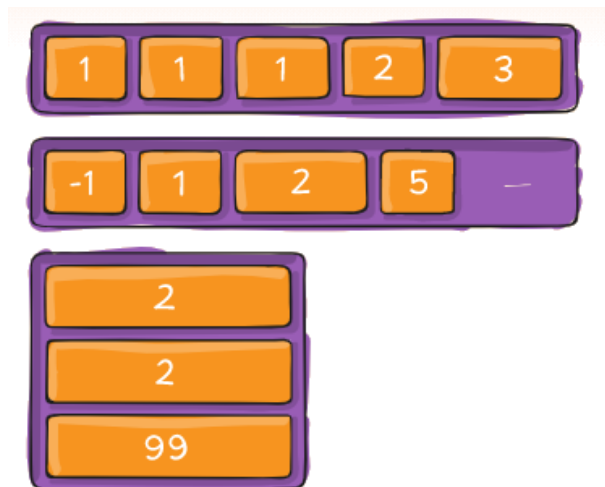
flex-basis

flex

align-self

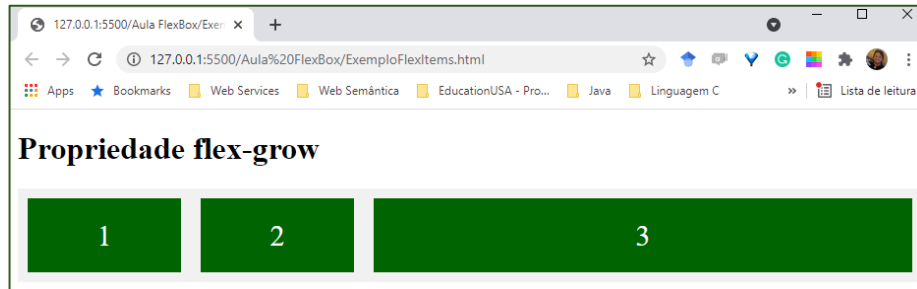
order define a ordem dos flex items.

```
<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>
```





Flex items: flex-grow



flex-grow especifica quanto um *flex item* crescerá em relação aos demais.

Propriedades

order

flex-grow

flex-shrink

flex-basis

flex

align-self

[Exemplo](#)

```
<div class="flex-container">
```

```
<div style="flex-grow: 1">1</div>
```

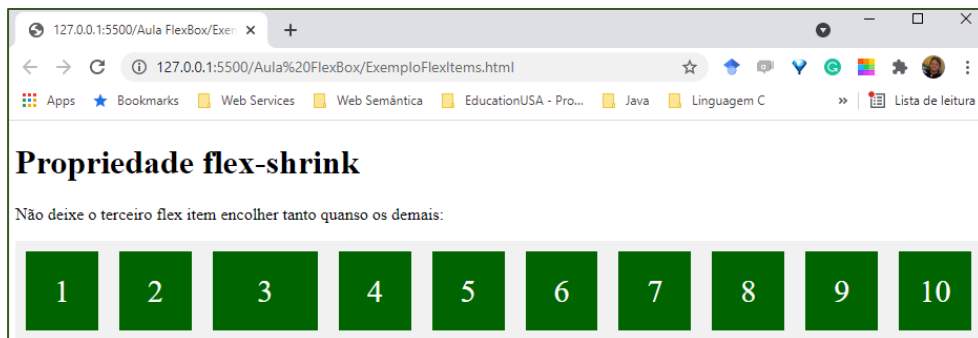
```
<div style="flex-grow: 1">2</div>
```

```
<div style="flex-grow: 8">3</div>
```

```
</div>
```



Flex items: flex-shrink



flex-shrink especifica quanto um *flex item* encolherá em relação aos demais.

Propriedades

order

flex-grow

flex-shrink

flex-basis

flex

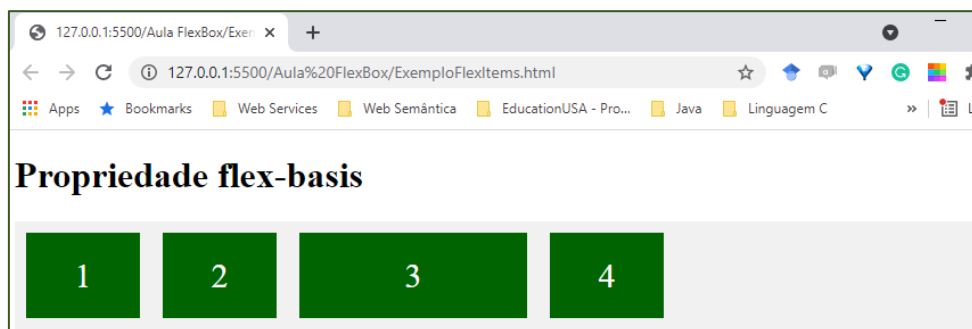
align-self

```
<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>8</div>
  <div>9</div>
  <div>10</div>
</div>
```

[Exemplo](#)



Flex items: flex-basis



[Exemplo](#)

flex-basis especifica o comprimento inicial de um flex item.

Propriedades

order

flex-grow

flex-shrink

flex-basis

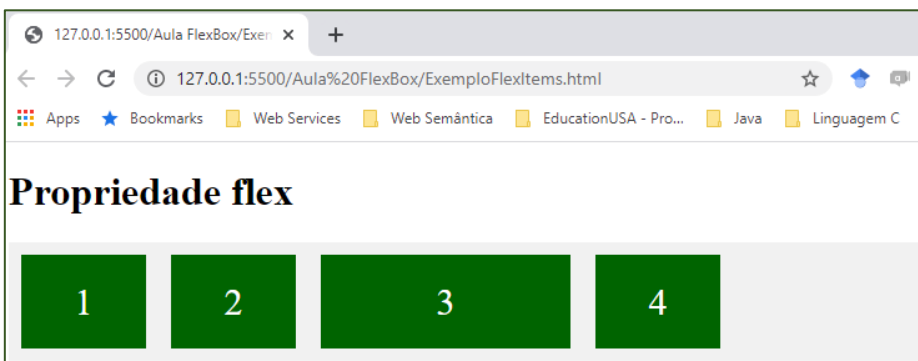
flex

align-self

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




Flex items: flex



[Exemplo](#)

Propriedades

order

flex-grow

flex-shrink

flex-basis

flex

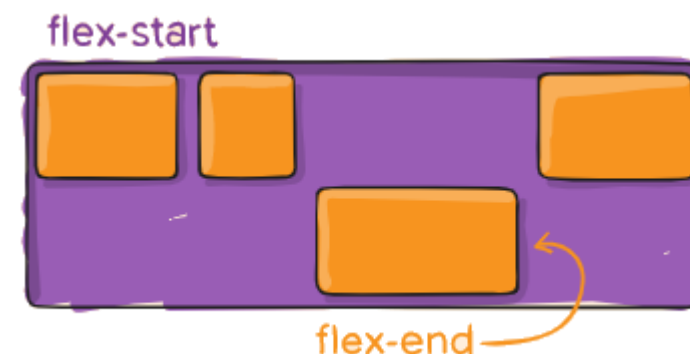
align-self

flex é um **atalho** das 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>
```



Flex items: align-self



[Exemplo](#)

align-self especifica o alinhamento do item selecionado dentro do contêiner.

Ela substitui o alinhamento padrão definido pela propriedade **align-items** do contêiner.

Propriedades

order

flex-grow

flex-shrink

flex-basis

flex

align-self

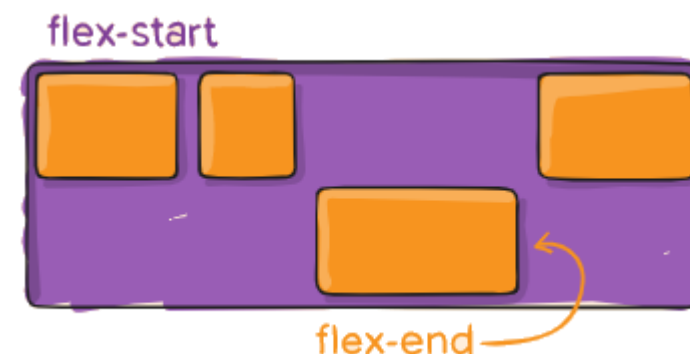
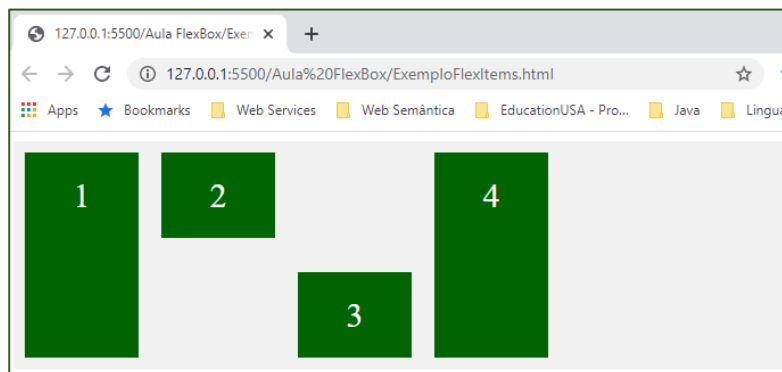
```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div style="align-self: center">3</div>
  <div>4</div>
</div>
```

```
.flex-container {
  display: flex;
  background-color: #f1f1f1;
  height: 200px;
}
```

Usa os mesmos valores da propriedade do Flex container: **align-items**



Flex items: align-self



Exemplo

align-self especifica o alinhamento do item selecionado dentro do contêiner.

Ela substitui o alinhamento padrão definido pela propriedade **align-items** do contêiner.

Propriedades

order

flex-grow

flex-shrink

flex-basis

flex

align-self

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

```
.flex-container {
  display: flex;
  background-color: #f1f1f1;
  height: 200px;
}
```



Propriedades dos Flex items

Propriedade	Descrição
align-self	Especifica o alinhamento para um <i>flex item</i> (substitui a propriedade <i>align-items</i> do contêiner)
flex	Uma abreviação para as propriedades <i>flex-grow</i> , <i>flex-shrink</i> e <i>flex-basis</i>
flex-basis	Especifica o comprimento inicial de um <i>flex item</i>
flex-grow	Especifica quanto um <i>flex item</i> crescerá em relação ao resto dos <i>flex items</i> dentro do contêiner
flex-shrink	Especifica o quanto um <i>flex item</i> encolherá em relação ao resto dos <i>flex items</i> dentro do contêiner
order	Especifica a ordem dos <i>flex items</i> dentro do contêiner

CSS

Media Queries



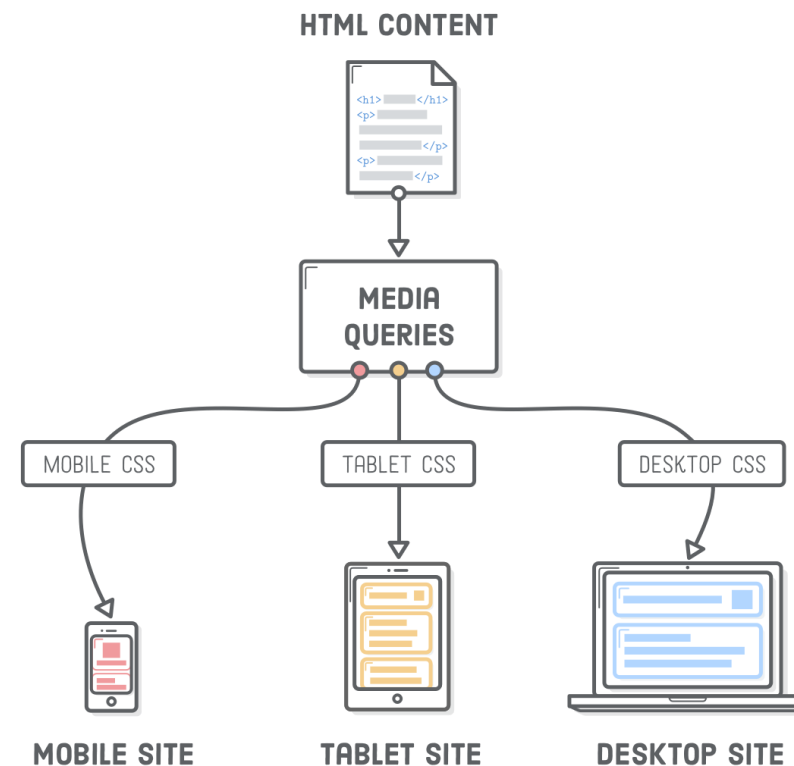


Media Queries – recordando...

Usa a regra **@media** para incluir um bloco de propriedades CSS apenas se uma determinada **condição for verdadeira**.

```
@media only screen and (max-width: 600px) {  
  body {  
    background-color: lightblue;  
  }  
}
```

* Se a janela do navegador tiver 600 px ou menos, a cor de fundo será azul claro



Flexbox

Responsivo





Dektops, Laptops

1	2	3
---	---	---

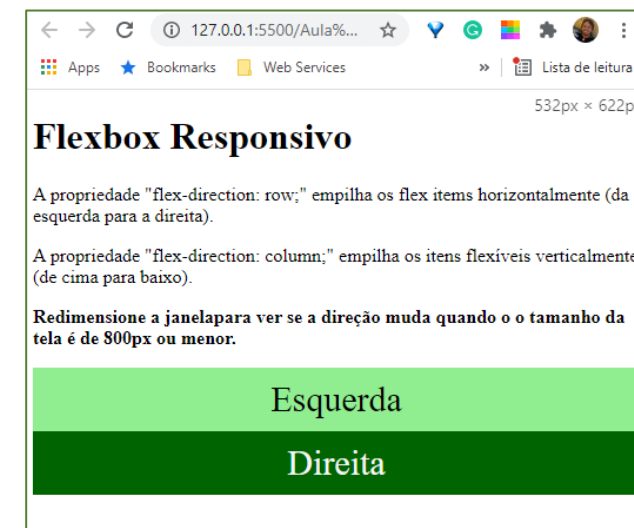
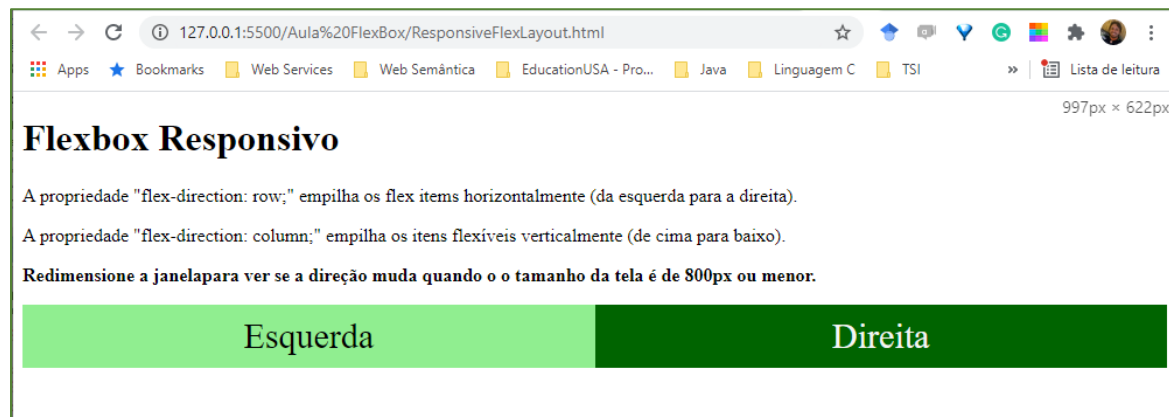
Tablets, smartphones

1
2
3

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

```
@media (max-width: 800px) {
  .flex-container {
    flex-direction: column;
  }
}
```

/* Layout responsivo altera o layout para 1-coluna ao invés de 2-colunas quando telas forem menores que 800px*/





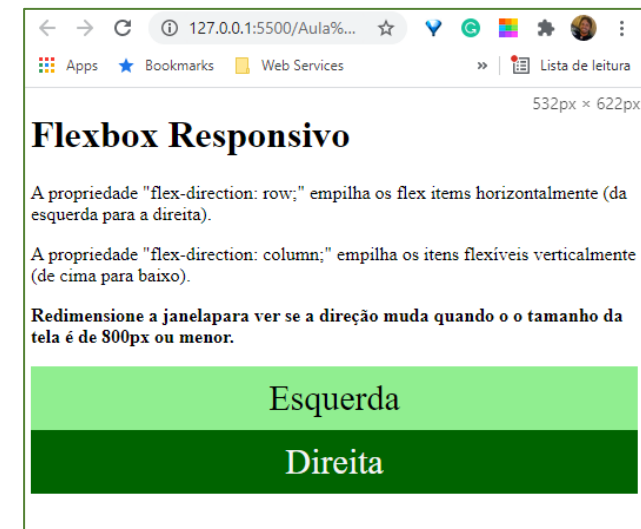
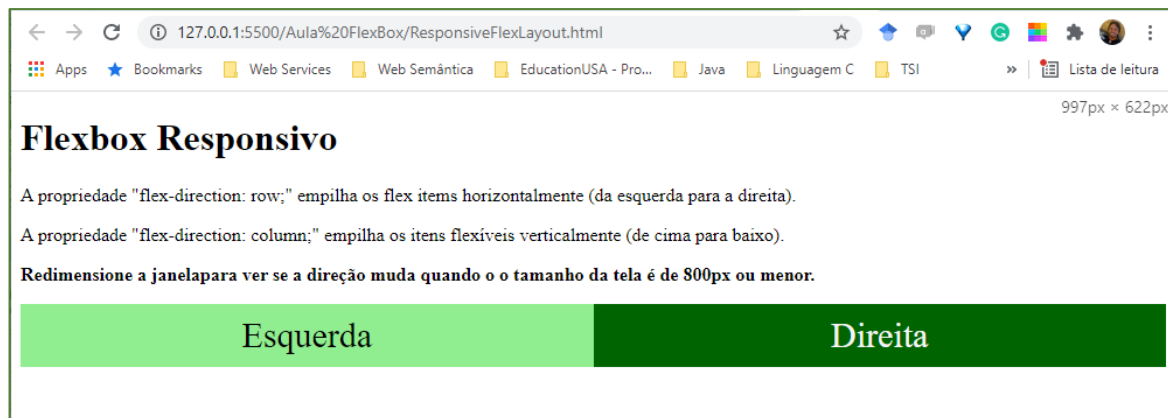
```
.flex-container {
  display: flex;
  flex-wrap: wrap;
}
```

```
.flex-item-left {
  flex: 50%;
}
```

```
.flex-item-right {
  flex: 50%;
}
```

```
@media (max-width: 800px) {
  .flex-item-right, .flex-item-left {
    flex: 100%;
  }
}
```

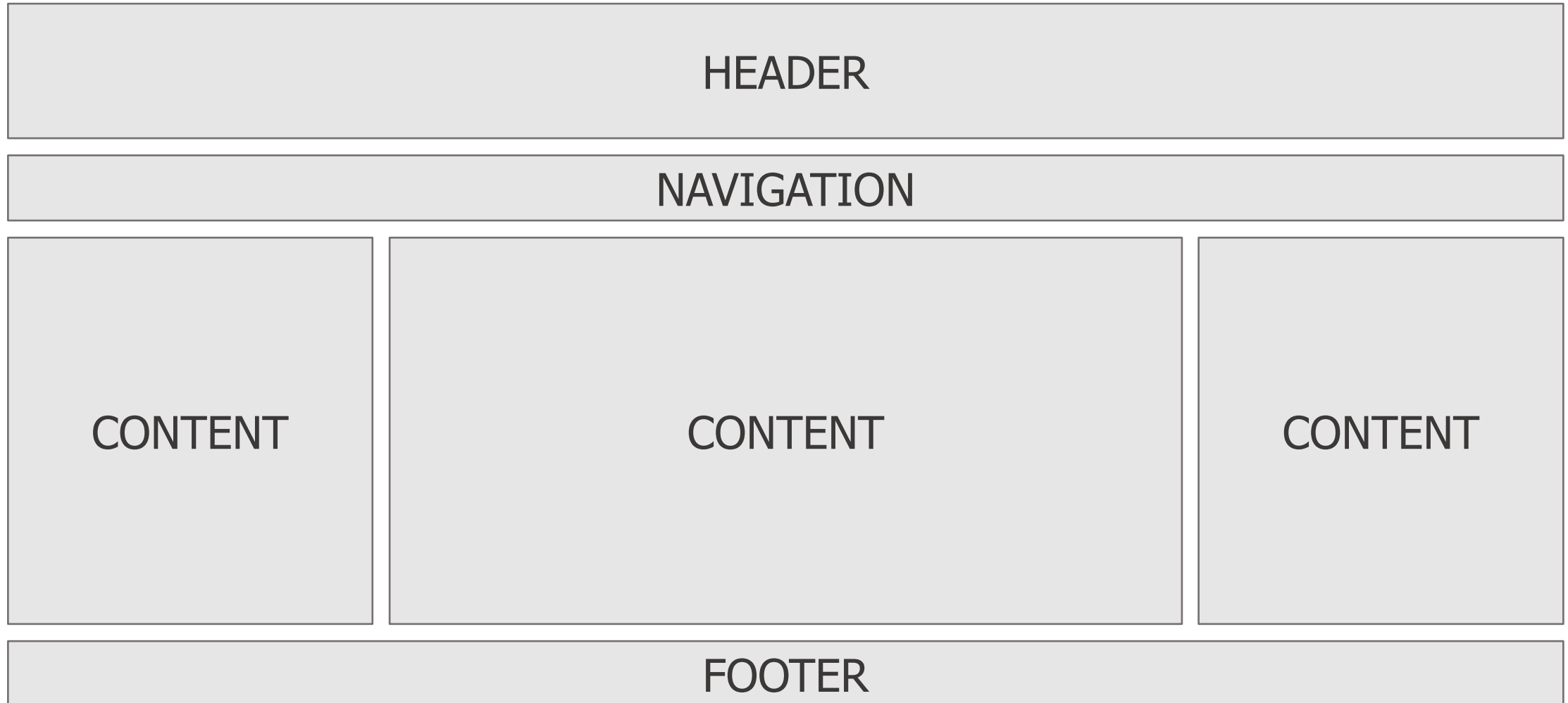
/* Layout responsivo altera o layout para 1-coluna ao invés de 2-colunas quando telas forem menores que 800px*/



Atividade



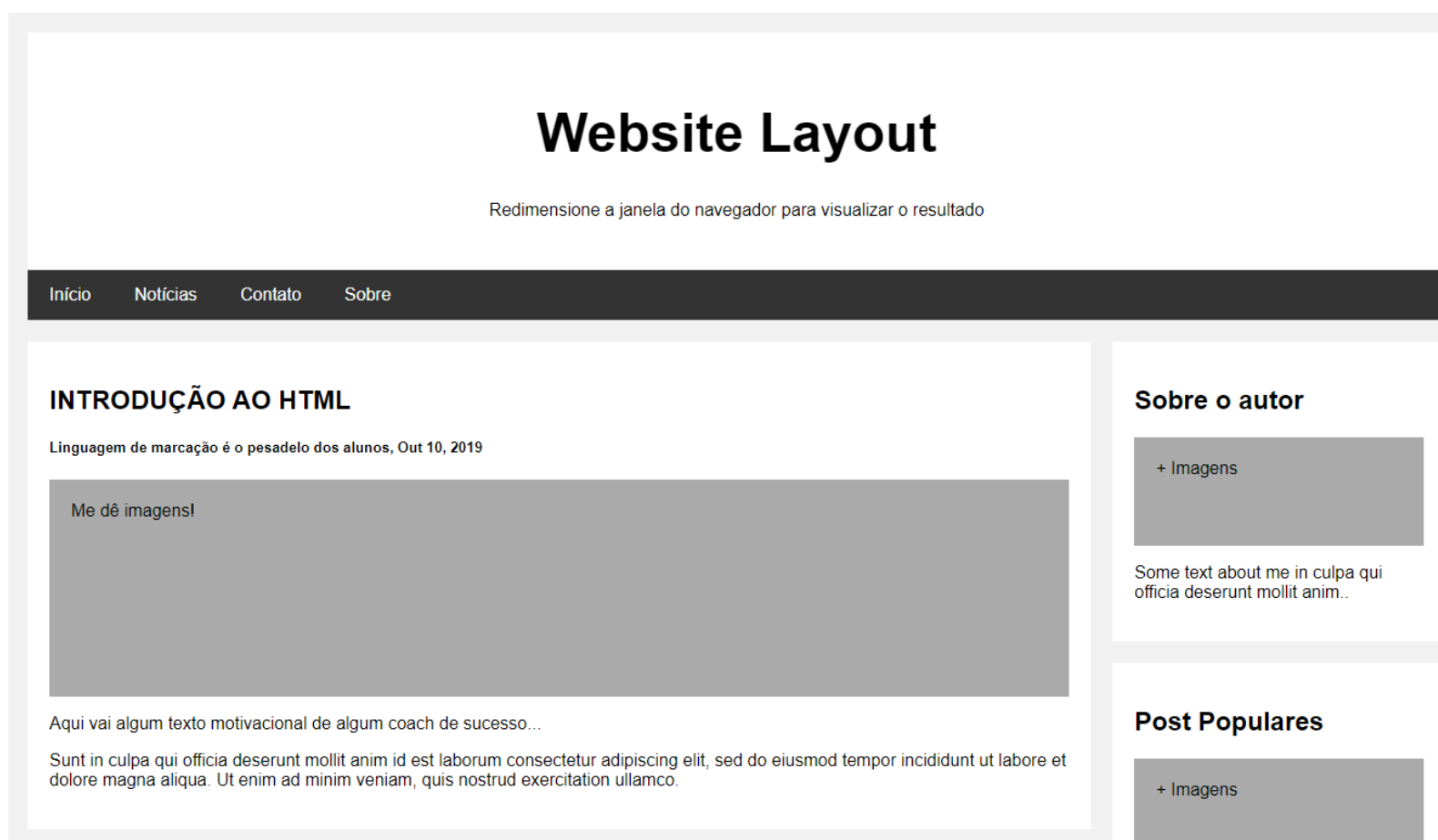
Website Layout





Atividade – Flex responsivo

Use o flexbox para criar um site responsivo, contendo uma barra de navegação flexível e conteúdo flexível:





Atividade – Flex responsivo

Website Layout

Redimensione a janela do navegador para visualizar o resultado

Início Notícias Contato Sobre

INTRODUÇÃO AO HTML

Linguagem de marcação é o pesadelo dos alunos, Out 10, 2019

Me dê imagens!

Website Layout

Redimensione a janela do navegador para visualizar o resultado

Início

Notícias

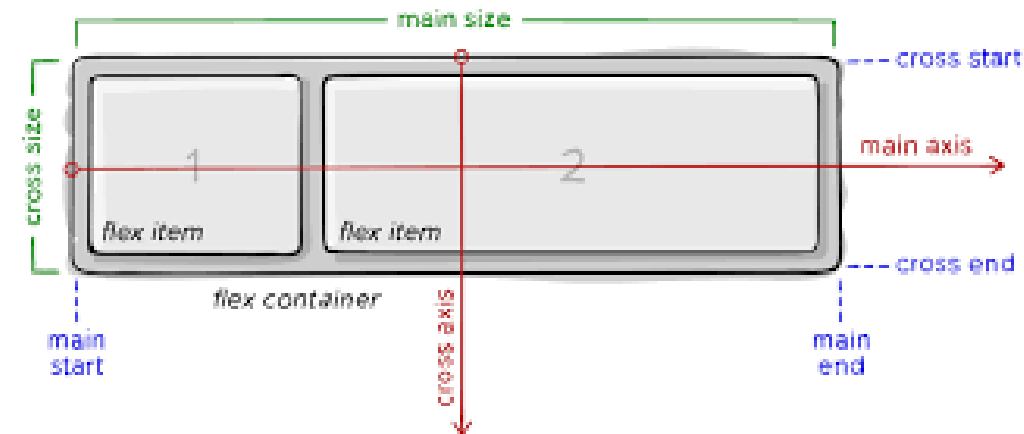
Contato

Sobre

INTRODUÇÃO AO HTML

Linguagem de marcação é o pesadelo dos alunos, Out 10, 2019

Me dê imagens!



CSS

FlexBox Layout