

**INSTITUTO  
FEDERAL**

Pernambuco

---

Campus  
Jaboatão dos  
Guararapes

**4**

# Desenvolvimento Web I

Informática para Internet

CSS3 – Flexbox

Conserva o modelo das sãs palavras que de mim tens ouvido, na fé e no amor que há em Cristo Jesus.  
2 Timóteo 1:13



INSTITUTO  
FEDERAL  
Pernambuco

Campus  
Jaboatão dos  
Guararapes



# Contatos

**Carlos José**

[carlos.silva@jaboatao.ifpe.edu.br](mailto:carlos.silva@jaboatao.ifpe.edu.br)

[@carlosjoser2n](https://twitter.com/carlosjoser2n)



# Introdução

O Flexbox é o nome comumente usado para CSS Flexible Box Layout Module, um modelo de layout para exibir itens em uma única dimensão - em linha ou em coluna.

Na especificação, o flexbox é descrito como um modelo de layout para o design da interface do usuário.

A característica principal do flexbox é o fato de que itens em um layout flexível podem crescer e encolher.



# Introdução

O espaço pode ser atribuído aos próprios itens, ou distribuído entre ou ao redor dos itens.

O Flexbox também permite o alinhamento de itens no eixo principal (*main axis*) ou transversal (*cross axis*), proporcionando assim um alto nível de controle sobre o tamanho e o alinhamento de um grupo de itens.



# Introdução



ALIGNMENT



DIRECTION



ORDER



SIZE

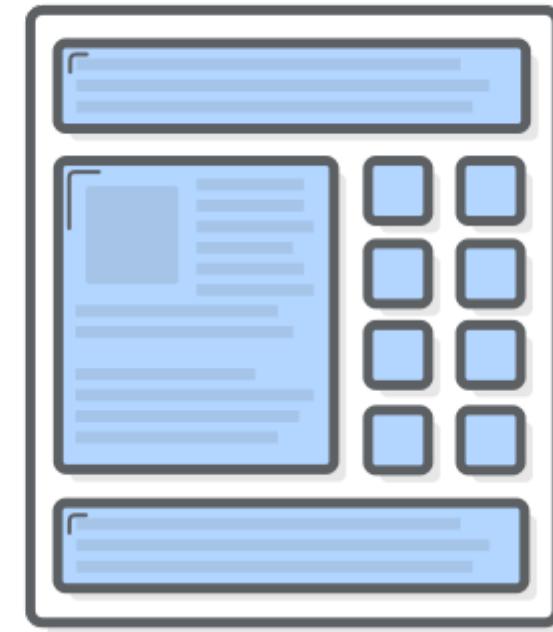


# Introdução



**FLOATS**

(MAGAZINE-STYLE LAYOUTS)



**FLEXBOX**

(OVERALL PAGE STRUCTURE)



# Introdução



“FLEX CONTAINER”



“FLEX ITEMS”



# Flex: container



# Flex: container

Destinado a criar um contexto de formatação flex para seus elementos-filhos.

O contexto de formatação estabelecido é o *nível flex*, com a particularidade de não reconhecer as propriedades *float* e *clear* da técnica CSS Box Model.



# Flex: container

## Propriedades

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



# Flex: container

Propriedades > display

Para criar um container flex, definimos o valor da propriedade de exibição do container da área para *flex* ou *inline-flex*.

Assim que fazemos isso, os filhos diretos desse recipiente se tornam itens flexíveis.

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

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



# Flex: container

## Propriedades > display

Ao criar um contêiner flexível, todos os itens flex contidos se comportarão da seguinte maneira:

- Os termos são exibidos em uma linha (o padrão da propriedade de *flex-direction* é a *row*).
- Os itens começam a partir da borda inicial do eixo principal (*main axis*).
- Os itens não se estendem (*stretch*) na dimensão principal, mas podem diminuir (*shrink*).
- Os itens se estenderão para preencher o tamanho do eixo transversal.
- A propriedade de *flex-basis* é definida como *auto*.
- A propriedade *flex-wrap* está configurada para *nowrap*.



# flexbox-01.html

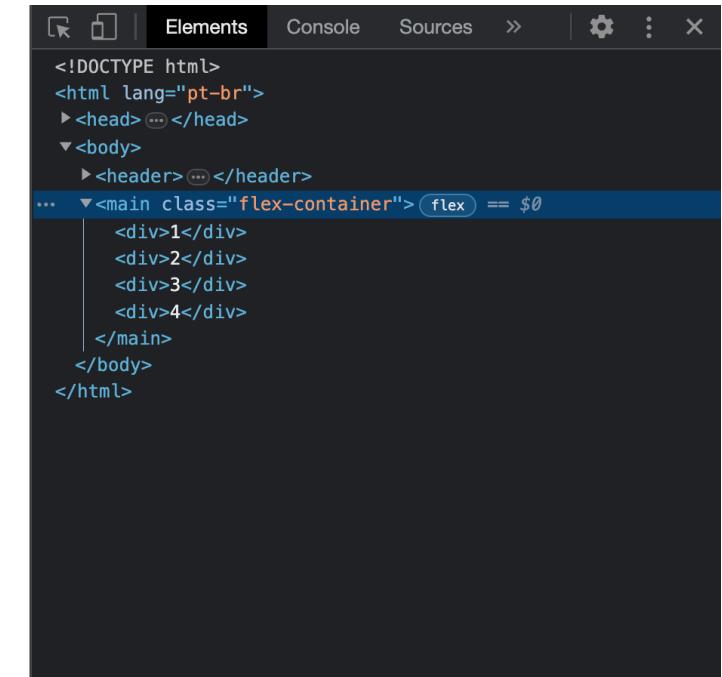
```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8"/>
5   <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
6   <title>Flexbox - 01</title>
7   <style>
8     *,:*,:before,*,:after{box-sizing: border-box;}
9     body{margin: 3em;}
10    header h1{margin-bottom: 1em; text-align: center;}
11    .flex-container{
12      display: flex; margin-left: auto; margin-right: auto;
13      width: 600px; height: 450px; border: 1px solid #004d40;
14    }
15    div{color: #004d40; font-size: 2.5em; padding: 10px; background-color: #f4ff81;}
16  </style>
17 </head>
18 <body>
19   <header><h1>Flexbox - 01</h1></header>
20   <main class="flex-container">
21     <div>1</div><div>2</div>
22     <div>3</div><div>4</div>
23   </main>
24 </body>
25 </html>
```



# flexbox-01.html

## Flexbox - 01

1 2 3 4



The screenshot shows the browser's developer tools with the 'Elements' tab selected. The DOM tree is visible, starting with the <html> tag, followed by <head>, <body>, <header>, and <main class="flex-container">. Inside the main container, there are four <div> elements with the numbers 1, 2, 3, and 4 respectively. The 'flex-container' class is highlighted with a blue background.

```
<!DOCTYPE html>
<html lang="pt-br">
  <head> ...
  <body>
    <header> ...
    <main class="flex-container">
      <div>1</div>
      <div>2</div>
      <div>3</div>
      <div>4</div>
    </main>
  </body>
</html>
```

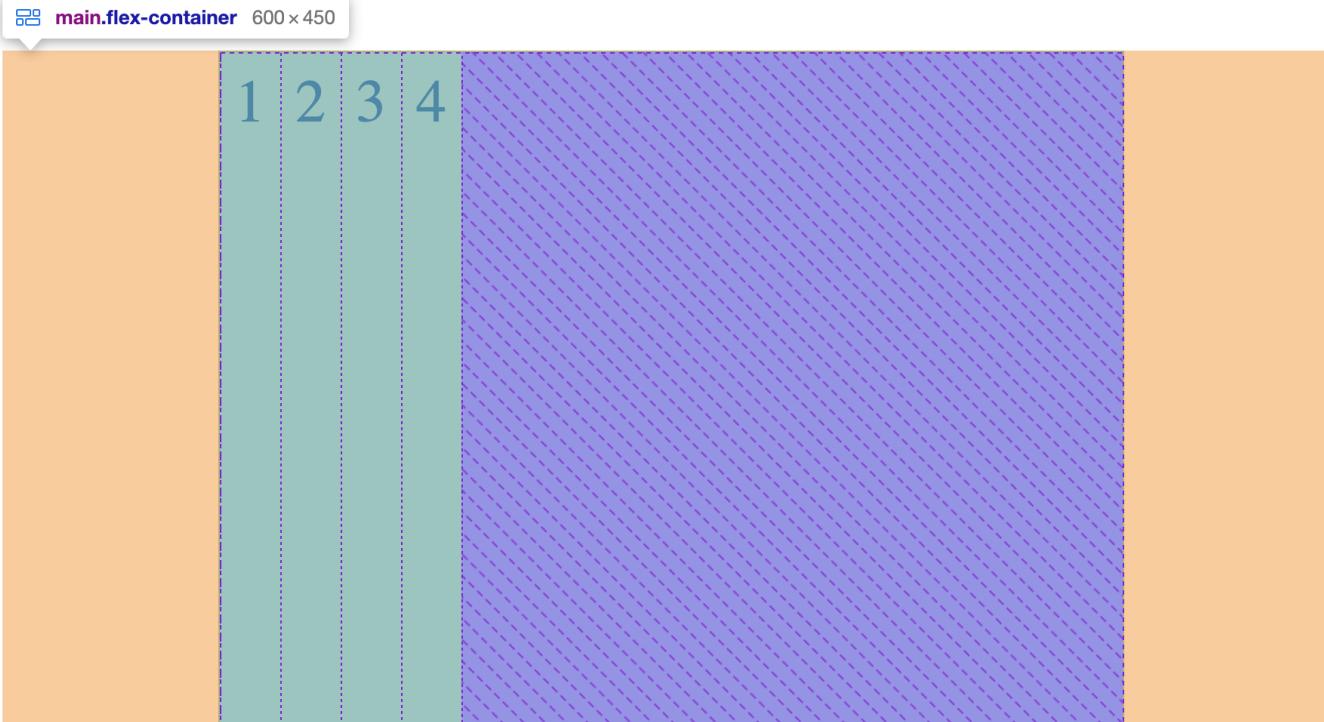
Below the DOM tree, the 'Styles' tab is selected in the developer tools. It shows the CSS rules for the 'flex-container' class and other elements. The 'flex-container' rule includes properties like display: flex, flex-direction: row, margin-left: auto, margin-right: auto, width: 600px, height: 450px, and border: 1px solid #004d40.

```
html body main.flex-container
Styles Computed Layout Event Listeners >
Filter :hov .cls + ↻
element.style {
}
.flex-container {
  display: flex; ⚡
  flex-direction: row;
  margin-left: auto;
  margin-right: auto;
  width: 600px;
  height: 450px;
  border: 1px solid #004d40;
}
*, *::before, *::after {
  box-sizing: border-box;
}
main {
```



# flexbox-01.html

## Flexbox - 01



```
<!DOCTYPE html>
<html lang="pt-br">
  <head> ...
  </head>
  <body>
    <header> ...
    <main class="flex-container">
      <div>1</div>
      <div>2</div>
      <div>3</div>
      <div>4</div>
    </main>
  </body>
</html>
```

html body main.flex-container

Styles Computed Layout Event Listeners >>

Filter :hover .cls + ↻

```
element.style { }
.main {
}
.flex-container {
  display: flex;
  flex-direction: row;
  margin-left: auto;
  margin-right: auto;
  width: 600px;
  height: 450px;
  border: 1px solid #004d40;
}
*, *::before, *::after {
  box-sizing: border-box;
}
.main { }
```

flexbox-01.html:11

flexbox-01.html:8

user agent stylesheet



# Flex: container

Propriedades > flex-direction

O eixo principal (*main axis*) é definido pela propriedade *flex-direction* na horizontal da esquerda para direita e o eixo transversal (*cross axis*) na vertical de cima para baixo.

Tudo o que nós fazemos com o flexbox remete a esses eixos, por isso vale a pena entender como eles funcionam desde o início.

```
.flex-container{  
  display: flex;  
  flex-direction: row; /* este é o valor padrão*/  
}
```



# Flex: container

Propriedades > flex-direction

Valores possíveis:

- row; row-reverse
- column; column-reverse

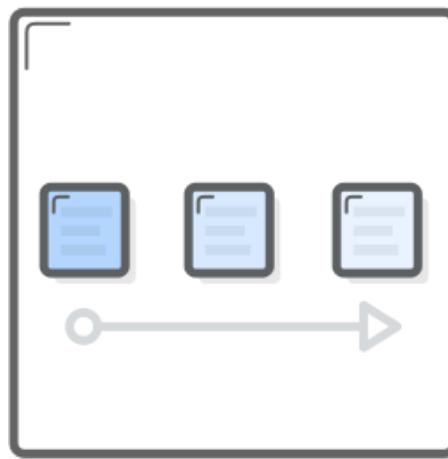
Se você escolher row ou row-reverse, o eixo se moverá junto com a linha na **direção inline**.

Se você escolher column ou column-reverse, o eixo se moverá do topo da página para baixo — na **direção block**.

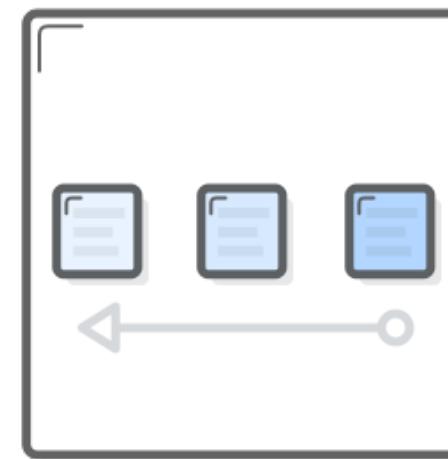


# Flex: container

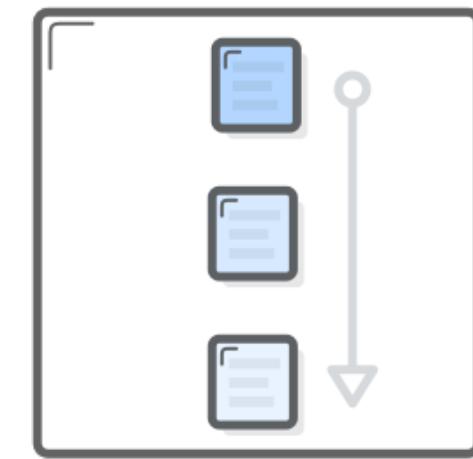
Propriedades > flex-direction



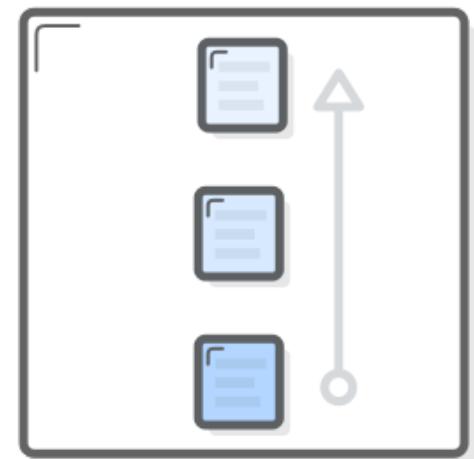
ROW



ROW-REVERSE



COLUMN

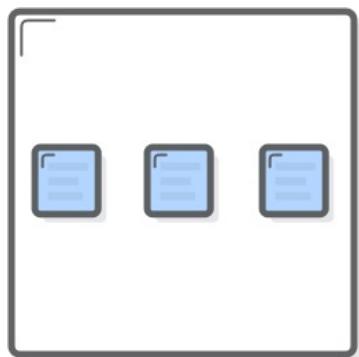


COLUMN-REVERSE



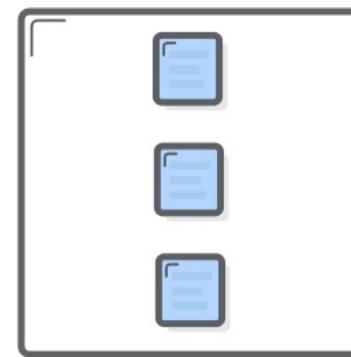
# Flex: container

Propriedades > flex-direction



**ROW**

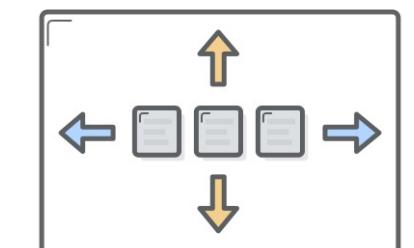
FLEX-DIRECTION: ROW;



**COLUMN**

FLEX-DIRECTION: COLUMN;

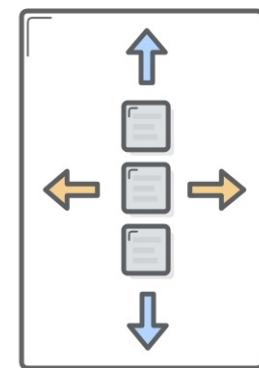
FLEX-DIRECTION: ROW;



ALIGN-ITEMS

JUSTIFY-CONTENT

FLEX-DIRECTION: COLUMN;



JUSTIFY-CONTENT

ALIGN-ITEMS



# flexbox-02.html

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8"/>
5   <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
6   <title>Flexbox - 02</title>
7   <style>
8     *,:before,:after{box-sizing: border-box;}
9     body{margin: 3em;}
10    header h1{margin-bottom: 1em; text-align: center;}
11    .flex-container{
12      display: flex; flex-direction: row; margin-left: auto; margin-right: auto;
13      width: 600px; height: 450px; border: 1px solid #004d40;
14    }
15    div{color: #004d40; font-size: 2.5em; padding: 10px; background-color: #f4ff81;}
16  </style>
17 </head>
18 <body>
19   <header><h1>Flexbox 02 - <code>flex-direction</code></h1></header>
20   <main class="flex-container">
21     <div>1</div><div>2</div>
22     <div>3</div><div>4</div>
23   </main>
24 </body>
25 </html>
```



# Flex: container

## Propriedades > justify-content

Destinada a alinhar os flex items dentro do container segundo a direção e sentido do eixo principal (*main axis*).

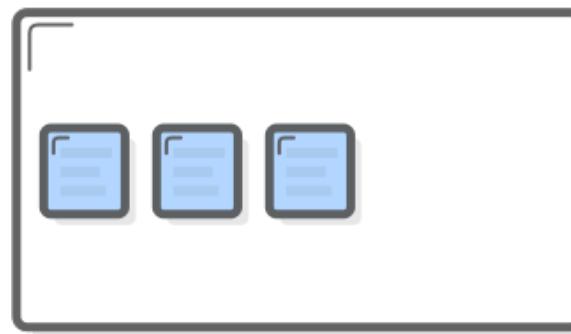
Os valores possíveis são: *center*; *flex-start*; *flex-end*; *space-around*; *space-between* e *space-evenly*

```
.flex-container{  
    display: flex;  
    flex-direction: row; /* este é o valor padrão */  
    justify-content: space-around; /* o valor padrão é flex-start */  
}
```

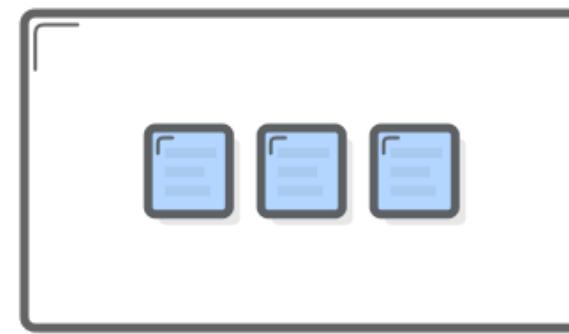


# Flex: container

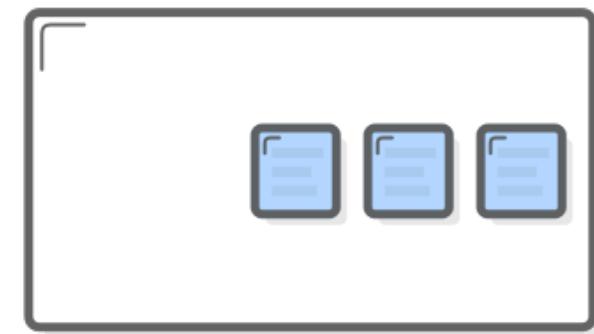
Propriedades > justify-content



FLEX-START



CENTER

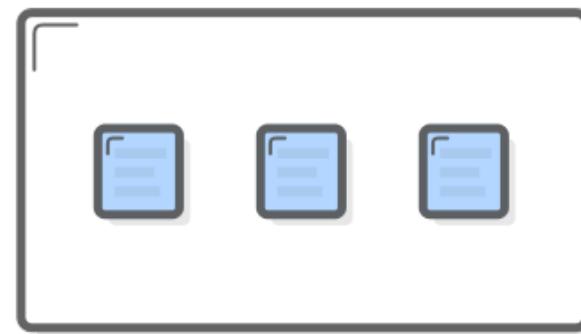


FLEX-END



# Flex: container

Propriedades > justify-content



**SPACE-AROUND**



**SPACE-BETWEEN**



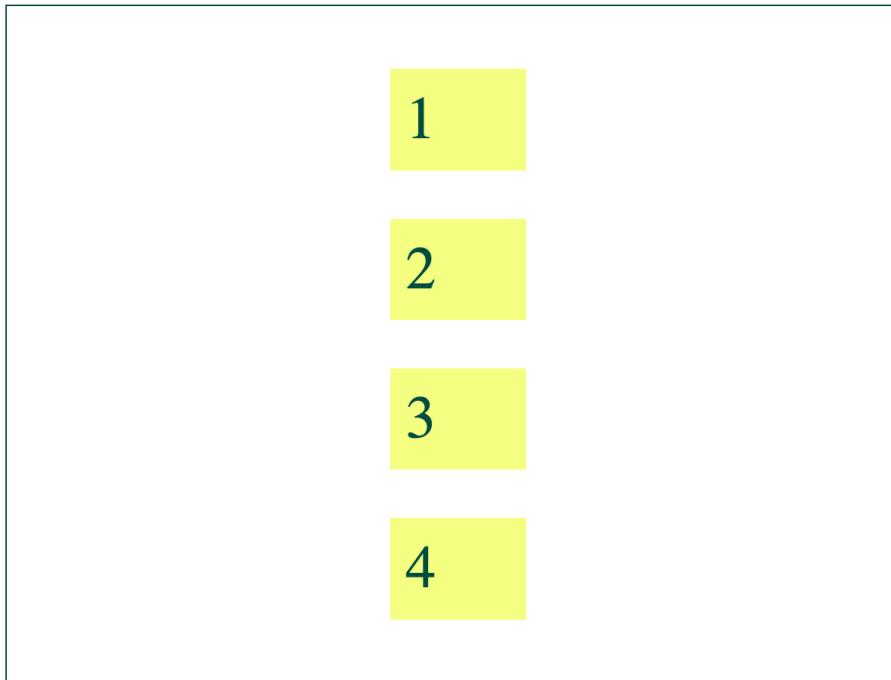
# flexbox-03.html

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8"/>
5   <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
6   <title>Flexbox - 03</title>
7   <style>
8     *,:before,:after{box-sizing: border-box;}
9     body{margin: 3em;}
10    header h1{margin-bottom: 1em; text-align: center;}
11    .flex-container{
12      display: flex; flex-direction: column; justify-content: center; align-items: center;
13      margin-left: auto; margin-right: auto;
14      width: 600px; height: 450px; border: 1px solid #004d40;
15    }
16    div{color: #004d40; font-size: 2.5em; padding: 10px; background-color: #f4ff81; width: 15%;
17      height: 15%; margin: 16px;}
18  </style>
19 </head>
20 <body>
21   <header><h1>Flexbox 03 - <code>justify-content</code></h1></header>
22   <main class="flex-container">
23     <div>1</div><div>2</div>
24     <div>3</div><div>4</div>
25   </main>
26 </body>
27 </html>
```



# flexbox-03.html

## Flexbox 03 - justify-content



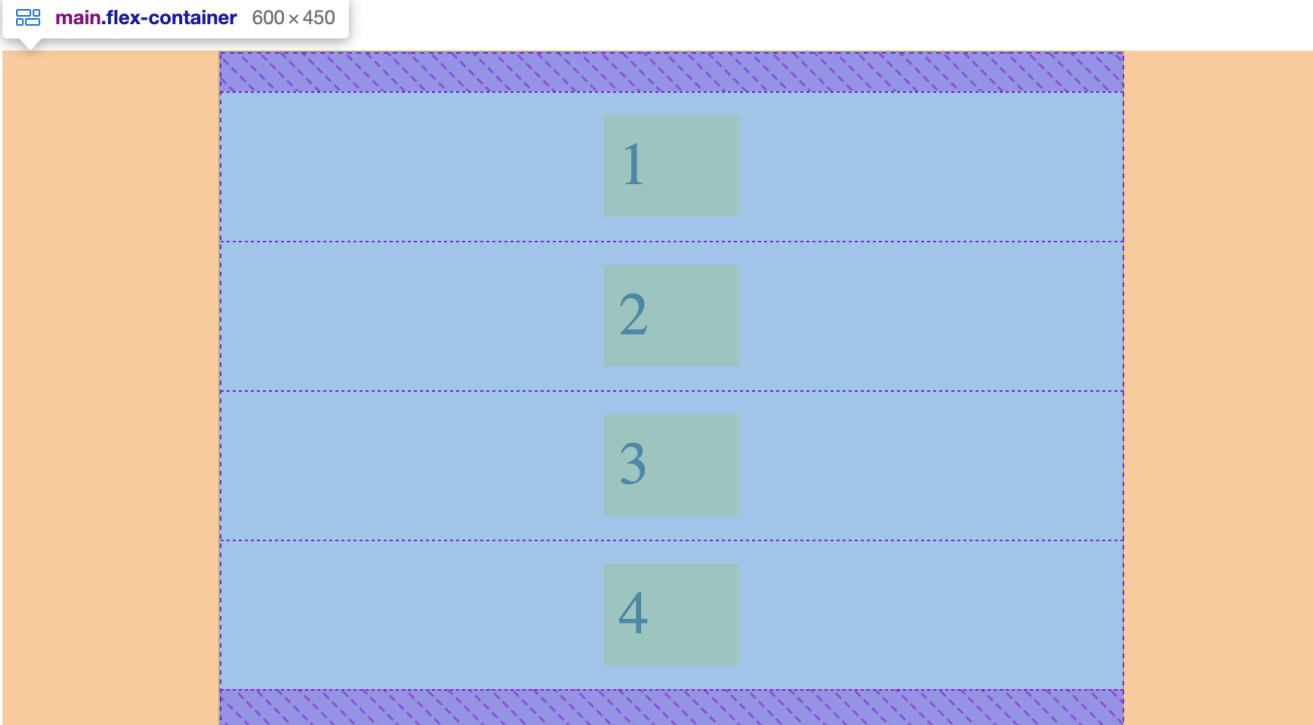
```
<!DOCTYPE html>
<html lang="pt-br">
  <head> ...
  </head>
  <body>
    <header> ...
    </header>
    ... <main class="flex-container"> flex == $0
      <div>1</div>
      <div>2</div>
      <div>3</div>
      <div>4</div>
    </main>
  </body>
</html>
```

```
html body main.flex-container
Styles Computed Layout Event Listeners >
Filter :hov .cls + ↻
element.style {
}
.flex-container {
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
  margin-left: auto;
  margin-right: auto;
  width: 600px;
  height: 450px;
  border: 1px solid #004d40;
}
*, *::before, *::after {
  box-sizing: border-box;
}
```



# flexbox-03.html

## Flexbox 03 - justify-content



```
<!DOCTYPE html>
<html lang="pt-br">
  <head> ... </head>
  <body>
    <header> ... </header>
    ... <main class="flex-container"> flex == $0
      <div>1</div>
      <div>2</div>
      <div>3</div>
      <div>4</div>
    </main>
  </body>
</html>
```

html body main.flex-container

Styles Computed Layout Event Listeners >>

Filter :hov .cls + ↻

element.style {  
}  
.flex-container {  
 display: flex;  
 flex-direction: column;  
 justify-content: center;  
 align-items: center;  
 margin-left: auto;  
 margin-right: auto;  
 width: 600px;  
 height: 450px;  
 border: 1px solid #004d40;  
}  
, \*::before, \*::after {  
 box-sizing: border-box;  
}

flexbox-03.html:11

flexbox-03.html:8



# Flex: container

## Propriedades > flex-wrap

Destinada a controlar a maneira como um container contém seus *flex items*; se uma linha continua ou com quebras de linha.

Os valores possíveis são: *nowrap*; *wrap*; *wrap-reverse*

```
.flex-container{  
    display: flex;  
    flex-direction: row; /* este é o valor padrão */  
    justify-content: space-around; /* o valor padrão é flex-start */  
    flex-wrap: wrap; /* o valor padrão é nowrap */  
}
```



# Flex: container

Propriedades > flex-wrap

Enquanto o Flexbox é um modelo unidimensional, é possível fazer com que nossos itens flexíveis se encaixem em várias linhas.

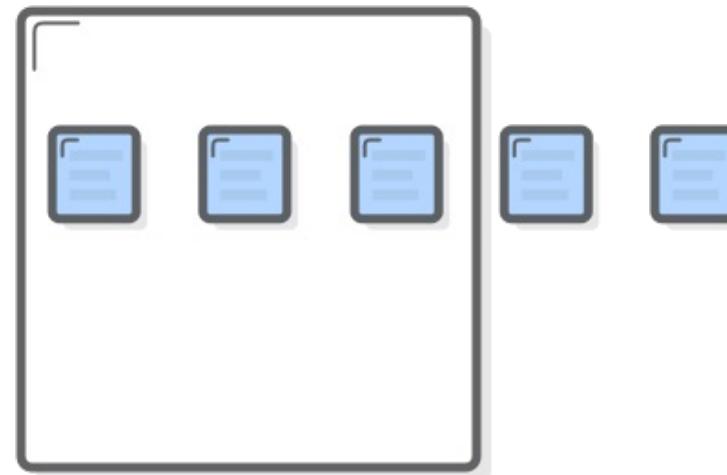
Ao fazê-lo, você deve considerar cada linha como um novo recipiente flex.

Qualquer distribuição espacial ocorrerá através dessa linha, sem referência às linhas de ambos os lados.



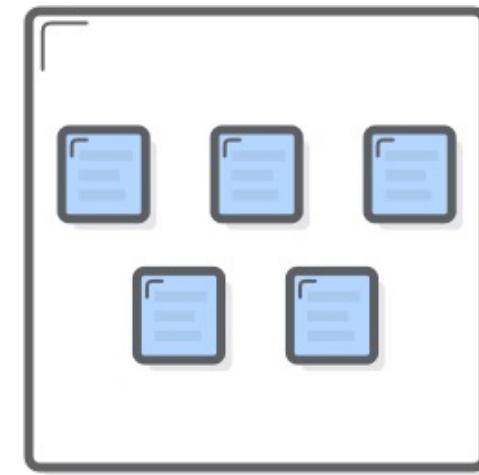
# Flex: container

Propriedades > flex-wrap



**NO WRAPPING**

FLEX-WRAP: NOWRAP;



**WITH WRAPPING**

FLEX-WRAP: WRAP;



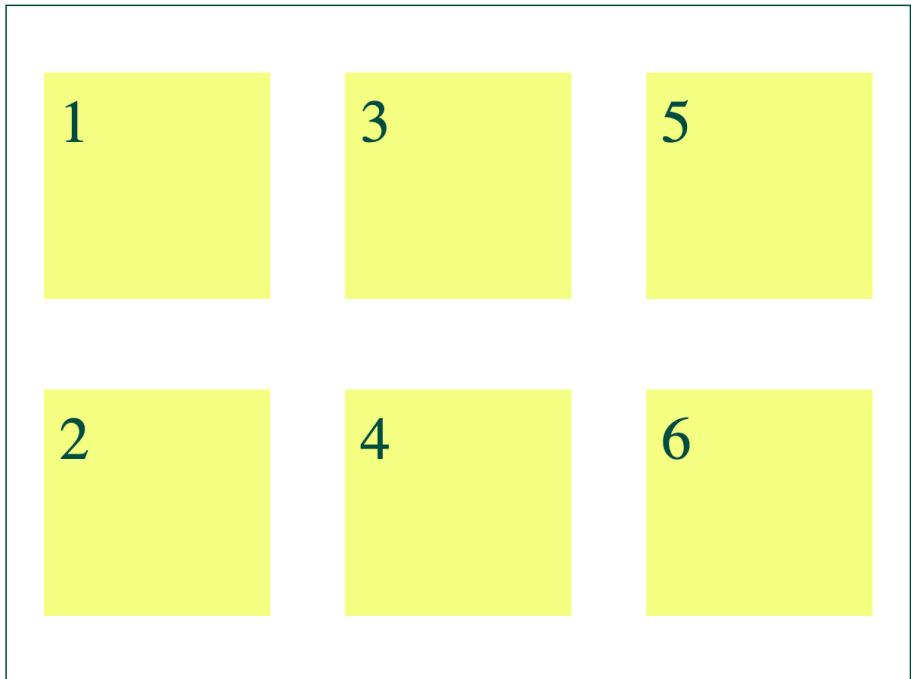
# flexbox-04.html

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8"/>
5   <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
6   <title>Flexbox - 04</title>
7   <style>
8     *,:before,:after{box-sizing: border-box;}
9     body{margin: 3em;}
10    header h1{margin-bottom: 1em; text-align: center;}
11    .flex-container{
12      display: flex;justify-content: space-evenly;align-items: center;
13      flex-flow: column wrap; /* flex-wrap: wrap; */
14      margin-left: auto; margin-right: auto;
15      width: 600px; height: 450px; border: 1px solid #004d40;
16    }
17    div{color: #004d40;font-size: 2.5em;padding: 10px;background-color: #f4ff81;width: 150px;
18      height: 150px; margin: 16px;}
19  </style>
20 </head>
21 <body>
22   <header><h1>Flexbox 04 - <code>flex-wrap</code></h1></header>
23   <main class="flex-container">
24     <div>1</div><div>2</div>
25     <div>3</div><div>4</div>
26     <div>5</div><div>6</div>
27   </main>
28 </body>
29 </html>
```



# flexbox-04.html

## Flexbox 04 - flex-wrap



DevTools is now available in Portuguese!

Always match Chrome's language [Switch DevTools to Portuguese](#)

Don't show again

Elements Console Sources > | [Settings](#) : X

```
<!DOCTYPE html>
<html lang="pt-br">
  <head> ... </head>
  <body>
    <header> ... </header>
    ... <main class="flex-container"> (flex) == $0
      <div>1</div>
      <div>2</div>
      <div>3</div>
      <div>4</div>
      <div>5</div>
      <div>6</div>
    </main>
  </body>
</html>
```

html body main.flex-container

Styles Computed Layout Event Listeners >

Filter :hov .cls + ↻

```
element.style { }
.flex-container {
  display: flex; ⚡
  justify-content: space-evenly;
  align-items: center;
  flex-flow: column wrap;
  flex-wrap: wrap;
  margin-left: auto;
  margin-right: auto;
  width: 600px;
  height: 450px;
  border: 1px solid #004d40;
}
*, *::before, *::after { }
```

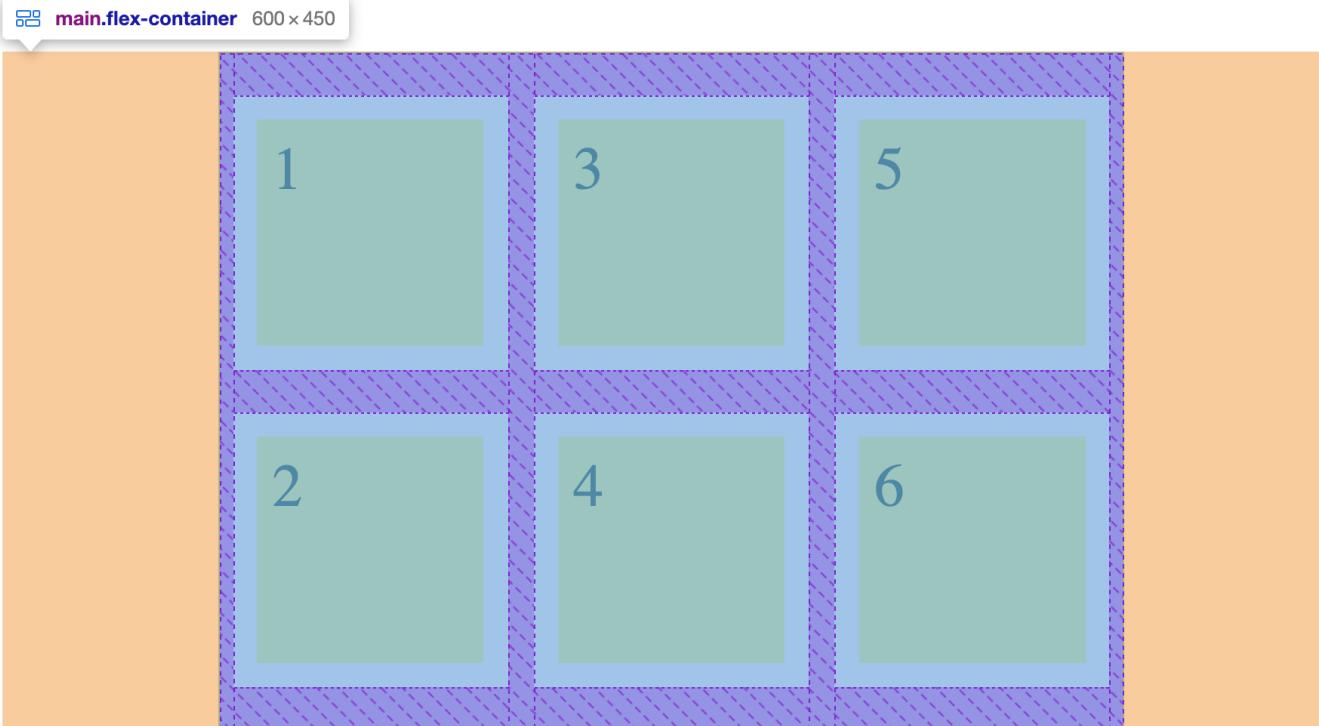
flexbox-04.html:11

flexbox-04.html:8



# flexbox-04.html

## Flexbox 04 - flex-wrap



DevTools is now available in Portuguese!

Always match Chrome's language Switch DevTools to Portuguese

Don't show again

Elements Console Sources > | ⋮ X

```
<!DOCTYPE html>
<html lang="pt-br">
  <head> ...
  </head>
  <body>
    <header> ...
    </header>
    ... <main class="flex-container"> flex == $0
      <div>1</div>
      <div>2</div>
      <div>3</div>
      <div>4</div>
      <div>5</div>
      <div>6</div>
    </main>
  </body>
</html>
```

html body main.flex-container

Styles Computed Layout Event Listeners >

Filter :hov .cls + ↻

```
element.style {
}
.flex-container {
  display: flex; ⚡
  justify-content: space-evenly;
  align-items: center;
  flex-flow: column wrap;
  flex-wrap: wrap;
  margin-left: auto;
  margin-right: auto;
  width: 600px;
  height: 450px;
  border: 1px solid #004d40;
}
*, ::before, ::after {
```

flexbox-04.html:11

flexbox-04.html:8



# Flex: container

## Propriedades > flex-flow

É a forma abreviada (*shorthand*) de declarar os valores para *flex-direction* e *flex-wrap* nessa ordem.

As combinações possíveis são: *row nowrap*, *row wrap*, *row wrap-reverse*, *column nowrap*, *column wrap*, *column wrap-reverse*, *row-reverse nowrap*, *row-reverse wrap*, *row-reverse wrap-reverse*, *column-reverse nowrap*, *column-reverse wrap*, *column-reverse wrap-reverse*.

O valor padrão é: *row nowrap*.



# Flex: container

## Propriedades > align-items

Destinada a alinhar os *flex items* dentro do container segundo a direção e o sentido do eixo transversal (*cross axis*).

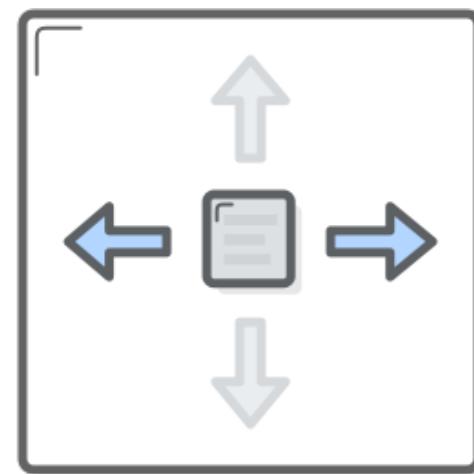
Os valores possíveis são: *flex-start*; *center*; *flex-end*; *stretch* e *baseline*

```
.flex-container{  
    display: flex;  
    flex-direction: row; /* este é o valor padrão */  
    justify-content: space-around; /* o valor padrão é flex-start */  
    flex-wrap: wrap; /* o valor padrão é nowrap */  
    align-items: stretch; /* valor padrão, que expande os flex items em toda altura*/  
}
```

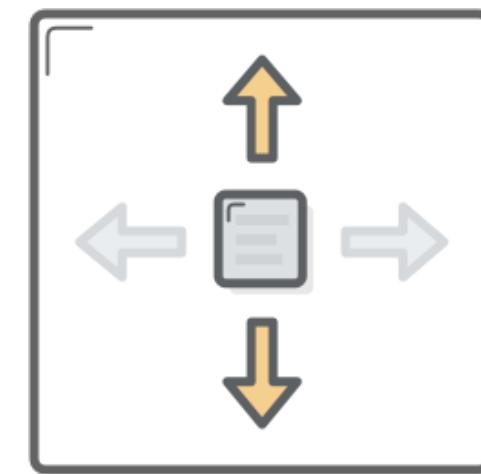


# Flex: container

Propriedades > align-items



**JUSTIFY-CONTENT**



**ALIGN-ITEMS**

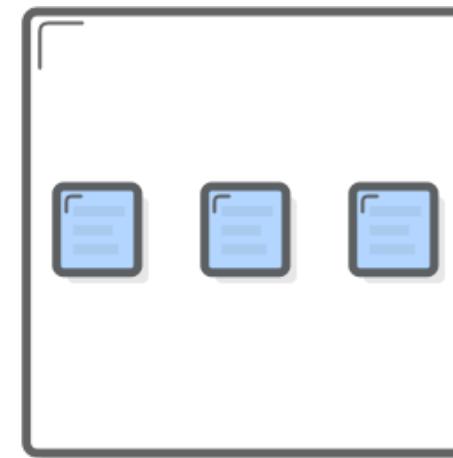


# Flex: container

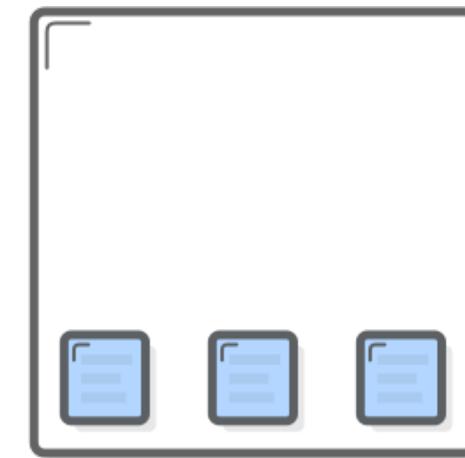
Propriedades > align-items



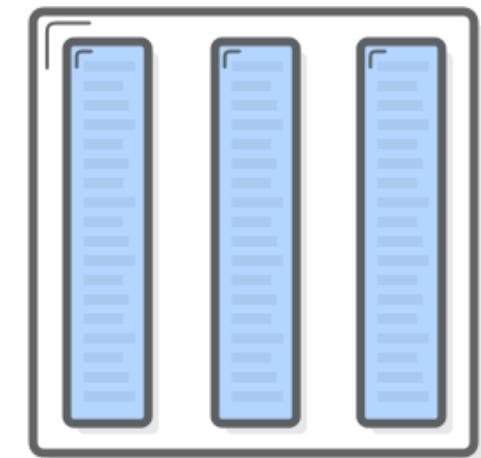
**FLEX-START**



**CENTER**



**FLEX-END**



**STRETCH**



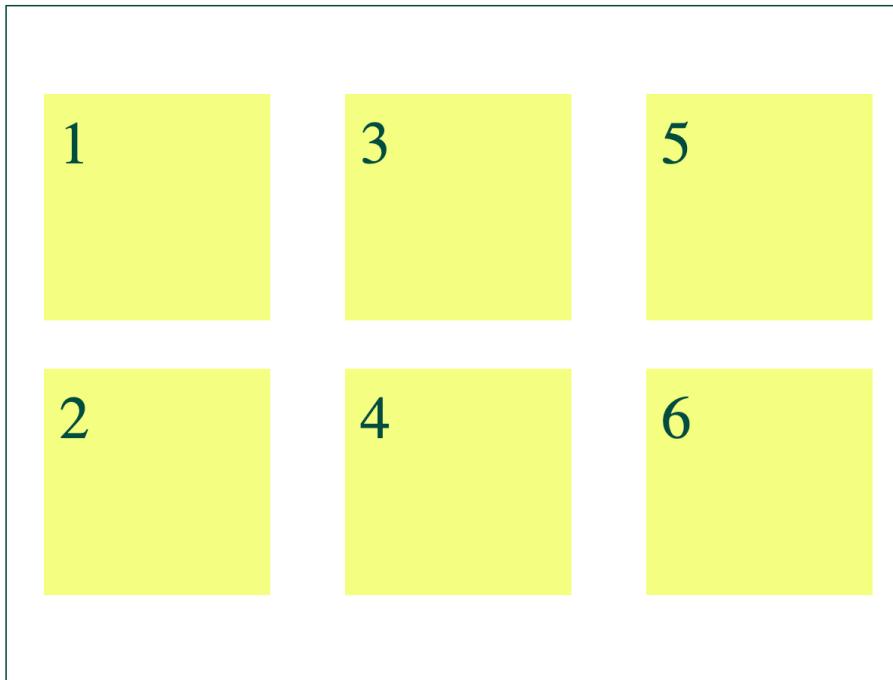
# flexbox-05.html

```
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8"/>
5   <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
6   <title>Flexbox - 05</title>
7   <style>
8     *,:before,:after{box-sizing: border-box;}
9     body{margin: 3em;}
10    header h1{margin-bottom: 2em;}
11    .flex-container{
12      display: flex; justify-content: center;flex-flow: column wrap;align-items: center;
13      margin-left: auto; margin-right: auto;
14      width: 600px; height: 450px; border: 1px solid #004d40;
15    }
16    div{
17      color: #004d40; font-size: 2.5em; padding: 10px;background-color: #f4ff81;
18      width: 150px; height: 150px; margin: 16px;}
19  </style>
20 </head>
21 <body>
22   <header><h1>Flexbox 05 - <code>align-items</code></h1></header>
23   <main class="flex-container">
24     <div>1</div><div>2</div>
25     <div>3</div><div>4</div>
26     <div>5</div><div>6</div>
27   </main>
28 </body>
29 </html>
```



# flexbox-05.html

## Flexbox 05 - align-items



DevTools is now available in Portuguese!

Always match Chrome's language [Switch DevTools to Portuguese](#)

Don't show again

Elements Console Sources > | [Settings](#) : X

```
<!DOCTYPE html>
<html lang="pt-br">
  <head> ...
  <body>
    <header> ...
    <main class="flex-container">
      <div>1</div>
      <div>2</div>
      <div>3</div>
      <div>4</div>
      <div>5</div>
      <div>6</div>
    </main>
  </body>
</html>
```

html body main.flex-container

Styles Computed Layout Event Listeners >

Filter :hov .cls + ↻

```
element.style { }
.flex-container {
  display: flex;
  justify-content: center;
  flex-flow: column wrap;
  align-items: center;
  margin-left: auto;
  margin-right: auto;
  width: 600px;
  height: 450px;
  border: 1px solid #004d40;
}
*, *::before, *::after {
  box-sizing: border-box;
```

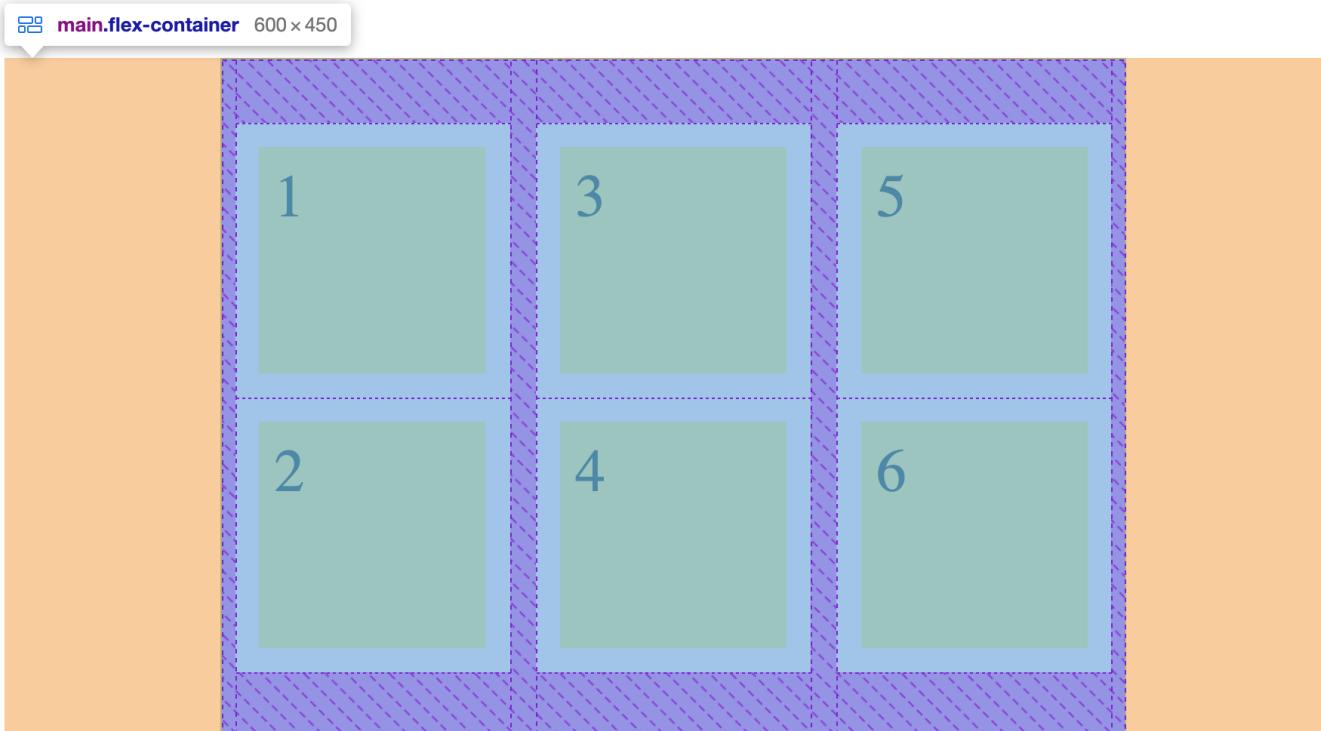
flexbox-05.html:11

flexbox-05.html:8



# flexbox-05.html

## Flexbox 05 - align-items



DevTools is now available in Portuguese!

Always match Chrome's language [Switch DevTools to Portuguese](#)

Don't show again

Elements Console Sources > | [Settings](#) : X

```
<!DOCTYPE html>
<html lang="pt-br">
  <head> ...
  </head>
  <body>
    <header> ...
    </header>
    ... <main class="flex-container"> flex == $0
      <div>1</div>
      <div>2</div>
      <div>3</div>
      <div>4</div>
      <div>5</div>
      <div>6</div>
    </main>
  </body>
</html>
```

html body main.flex-container

Styles Computed Layout Event Listeners >

Filter :hov .cls + ↻

```
element.style { }
.flex-container {
  display: flex; grid
  justify-content: center;
  flex-flow: column wrap;
  align-items: center;
  margin-left: auto;
  margin-right: auto;
  width: 600px;
  height: 450px;
  border: 1px solid #004d40;
}
*, *::before, *::after {
  box-sizing: border-box;
```

flexbox-05.html:11

flexbox-05.html:8



# Flex: container

## Propriedades > align-content

Destinada a alinhar os *flex items* dentro do container segundo a direção e o sentido do eixo transversal (*cross axis*) e aplica-se no caso em que os flex items são dispostos dentro do container em múltiplas linhas (*flex-wrap: wrap*).

Os valores possíveis são: *flex-start*; *center*; *flex-end*; *stretch* e *baseline*

```
.flex-container{  
    display: flex;  
    flex-direction: row; /* este é o valor padrão */  
    justify-content: space-around; /* o valor padrão é flex-start */  
    flex-wrap: wrap; /* o valor padrão é nowrap */  
    align-items: stretch; /* valor padrão, que expande os flex items em toda altura*/  
    align-content: flex-start; /* o valor padrão é stretch */  
}
```



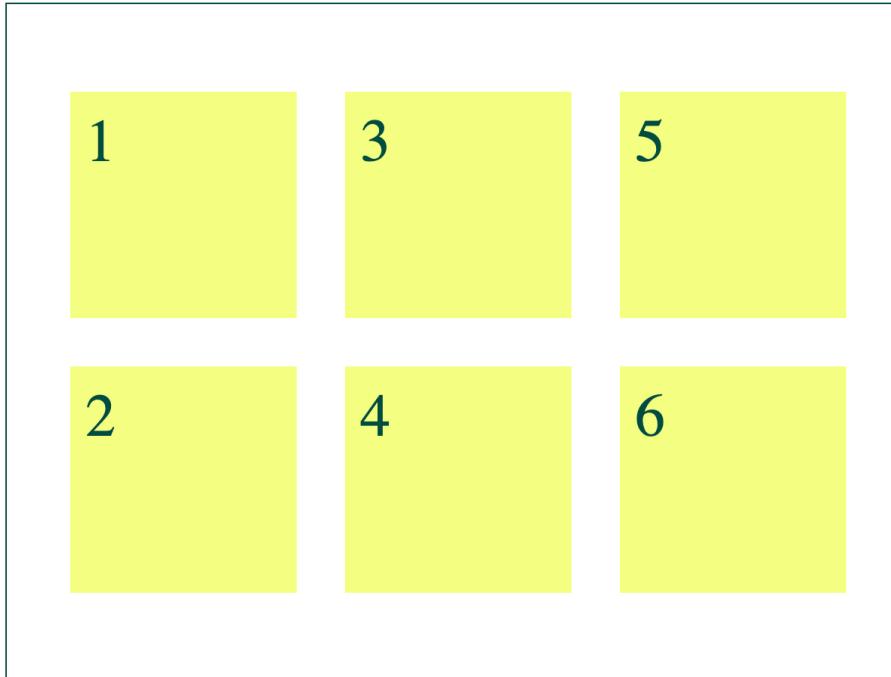
# flexbox-06.html

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8"/>
5   <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
6   <title>Flexbox - 06</title>
7   <style>
8     *,:before,:after{box-sizing: border-box;}
9     body{margin: 3em;}
10    header h1{margin-bottom: 1em;text-align: center;}
11    .flex-container{
12      margin-left: auto; margin-right: auto;
13      width: 600px; height: 450px; border: 1px solid #004d40;
14      display: flex;flex-flow: column wrap;justify-content: center;align-content: center;}
15    div{color: #004d40;font-size: 2.5em;padding: 10px;background-color: #f4ff81;width: 150px;
16      height: 150px; margin: 16px;
17    }
18  </style>
19 </head>
20 <body>
21   <header><h1>Flexbox 06 - <code>align-content</code></h1></header>
22   <main class="flex-container">
23     <div>1</div><div>2</div><div>3</div>
24     <div>4</div><div>5</div><div>6</div>
25   </main>
26 </body>
27 </html>
```



# flexbox-06.html

## Flexbox 06 - align-content



DevTools is now available in Portuguese!

Always match Chrome's language [Switch DevTools to Portuguese](#)

Don't show again

Elements Console Sources > | [Settings](#) : X

```
<!DOCTYPE html>
<html lang="pt-br">
  <head> ...
  </head>
  <body>
    <header> ...
    </header>
    <main class="flex-container"> flex == $0
      <div>1</div>
      <div>2</div>
      <div>3</div>
      <div>4</div>
      <div>5</div>
      <div>6</div>
    </main>
  </body>
</html>
```

html body main.flex-container

Styles Computed Layout Event Listeners >

Filter :hov .cls + ↻

```
element.style { }
.flex-container {
  margin-left: auto;
  margin-right: auto;
  width: 600px;
  height: 450px;
  border: 1px solid #004d40;
  display: flex; ⚡
  flex-flow: column wrap;
  justify-content: center;
  align-content: center;
}
*, *::before, *::after {
  box-sizing: border-box;
```

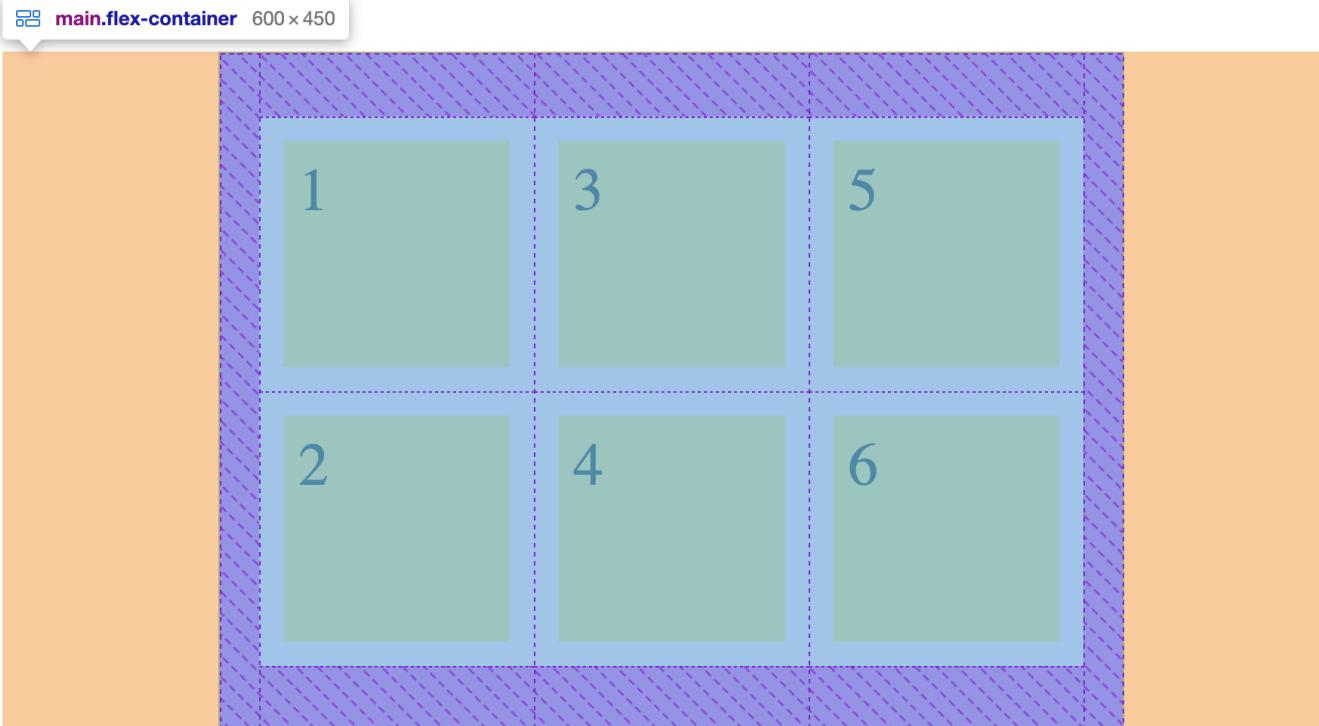
flexbox-06.html:11

flexbox-06.html:8



# flexbox-06.html

## Flexbox 06 - align-content



DevTools is now available in Portuguese!

Always match Chrome's language Switch DevTools to Portuguese

Don't show again

Elements Console Sources > | ⋮ X

```
<!DOCTYPE html>
<html lang="pt-br">
  <head> ...
  </head>
  <body>
    <header> ...
    </header>
    <main class="flex-container"> flex == $0
      <div>1</div>
      <div>2</div>
      <div>3</div>
      <div>4</div>
      <div>5</div>
      <div>6</div>
    </main>
  </body>
</html>
```

html body main.flex-container

Styles Computed Layout Event Listeners >

Filter :hov .cls + ↻

```
element.style { }
.flex-container {
  margin-left: auto;
  margin-right: auto;
  width: 600px;
  height: 450px;
  border: 1px solid #004d40;
  display: flex; flex
  flex-flow: column wrap;
  justify-content: center;
  align-content: center;
}
*, *::before, *::after {
  box-sizing: border-box;
```

flexbox-06.html:11

flexbox-06.html:8



# Flex: items



# Flex: items

## Propriedades

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



# Flex: items

## Propriedades > order

Destinada a reordenar os *flex items* dentro do container sem interferir com a ordem que eles ocupam na marcação.

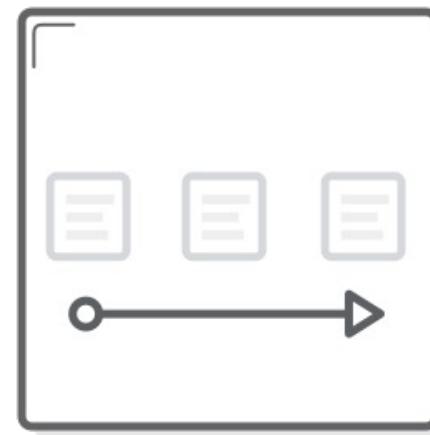
Os valores possíveis são: um número inteiro positivo ou negativo. O valor inicial é 0 (zero).

```
.photo-grid {  
  flex-direction: row;  
  align-items: center;  
}  
.first-item { order: 1; }  
.last-item { order: -1; }
```



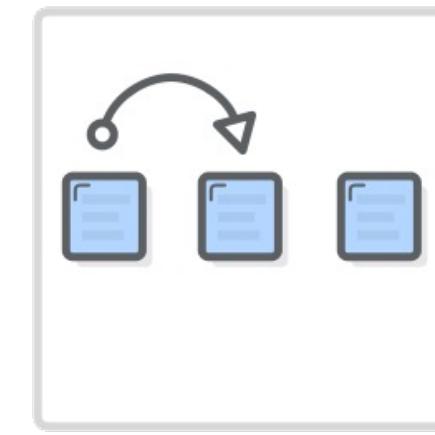
# Flex: items

Propriedades > order



**FLEX-DIRECTION**

(WHOLE CONTAINER)



**ORDER**

(INDIVIDUAL ITEMS)



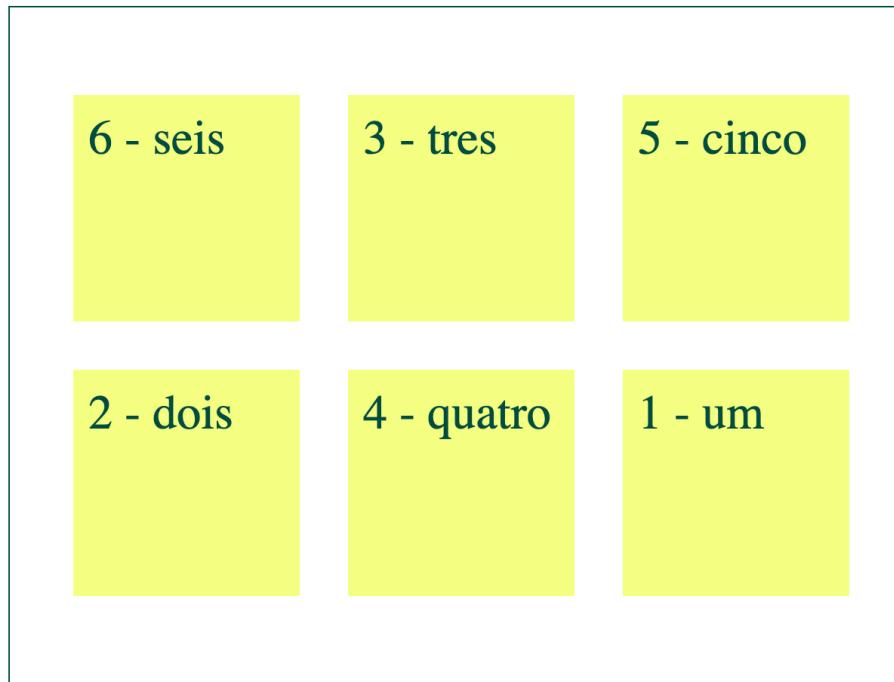
# flexbox-07.html

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8"/>
5   <title>Flexbox - 07</title>
6   <style>
7     *,:before,:after{box-sizing: border-box;}
8     body{margin: 3em;}
9     header h1{margin-bottom: 2em;}
10    .flex-container{
11      margin-left: auto; margin-right: auto;
12      width: 600px; height: 450px; border: 1px solid #004d40;
13      display: flex;justify-content: center; align-content: center;flex-flow: column wrap;
14    }
15    div{color: #004d40;font-size: 2em; padding: 10px;background-color: #f4ff81; width: 150px;
16      height: 150px; margin: 16px;}
17      main > :nth-child(6){order: -1;}
18      main > :nth-child(1){order: 1;}
19    </style>
20  </head>
21  <body>
22    <header><h1>Flexbox 07 - <code>order</code></h1></header>
23    <main class="flex-container">
24      <div>1 - um </div><div>2 - dois</div>
25      <div>3 - tres</div><div>4 - quatro</div>
26      <div>5 - cinco</div><div>6 - seis</div>
27    </main>
28  </body>
29 </html>
```



# flexbox-07.html

## Flexbox 07 - order



```
<!DOCTYPE html>
<html lang="pt-br">
  <head> ...
  <body>
    <main class="flex-container">
      <div>1 - um</div>
      <div>2 - dois</div>
      <div>3 - tres</div>
      <div>4 - quatro</div>
      <div>5 - cinco</div>
      <div>6 - seis</div>
    </main>
  </body>
</html>
```

html body main.flex-container

Styles Computed Layout Event Listeners >

Filter :hov .cls + ↻

```
element.style { }
.flex-container {
  margin-left: auto;
  margin-right: auto;
  width: 600px;
  height: 450px;
  border: 1px solid #004d40;
  display: flex;
  justify-content: center;
  align-content: center;
  flex-flow: column wrap;
}
```

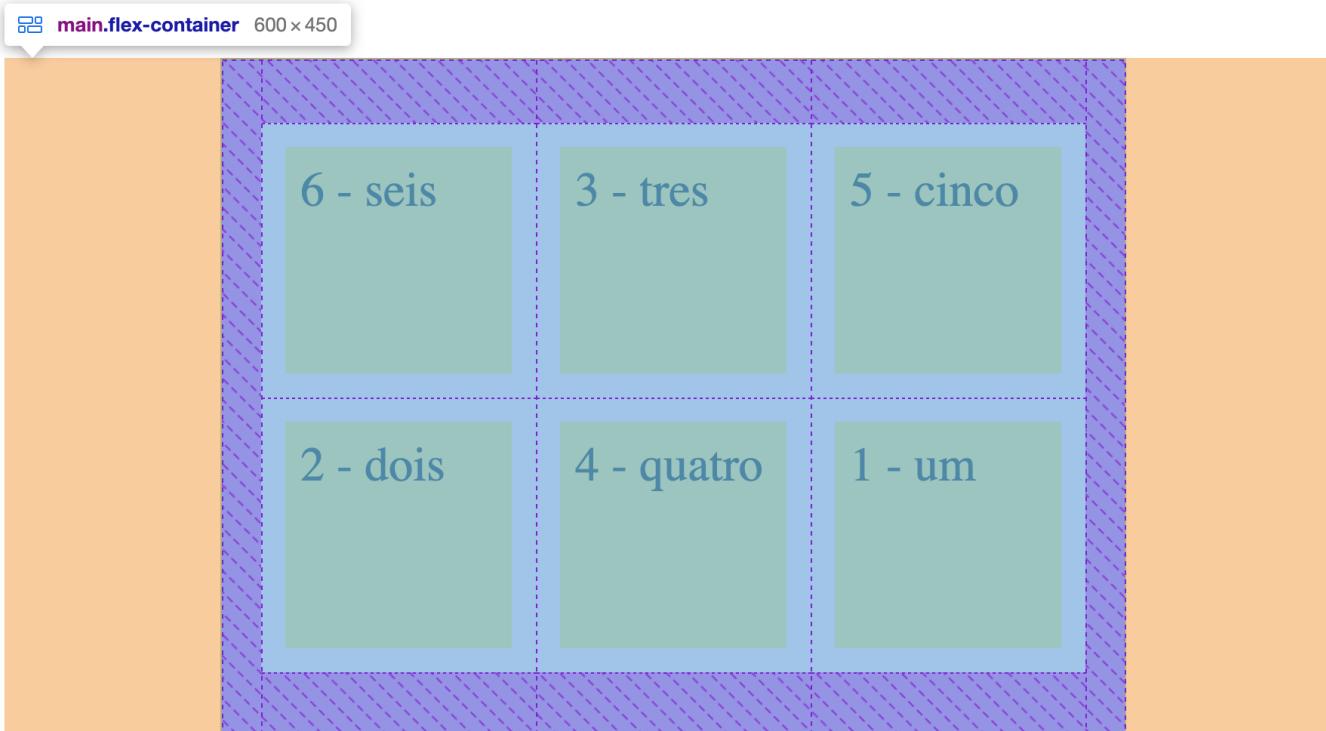
, \*::before, \*::after { }

flexbox-07.html:7



# flexbox-07.html

## Flexbox 07 - order



DevTools is now available in Portuguese!

Always match Chrome's language [Switch DevTools to Portuguese](#)

Don't show again

Elements Console Sources > | [Settings](#) | [More](#) | [X](#)

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>...</head>
  <body>
    <header>...</header>
    ... <main class="flex-container"> flex == $0
      <div>1 - um </div>
      <div>2 - dois</div>
      <div>3 - tres</div>
      <div>4 - quatro</div>
      <div>5 - cinco</div>
      <div>6 - seis</div>
    </main>
  </body>
</html>
```

html body main.flex-container

Styles Computed Layout Event Listeners >

Filter :hov .cls + ↻

```
element.style {  
}  
.flex-container {  
  margin-left: auto;  
  margin-right: auto;  
  width: 600px;  
  height: 450px;  
  border: 1px solid #004d40;  
  display: flex; grid  
  justify-content: center;  
  align-content: center;  
  flex-flow: column wrap;  
}  
*, *::before, *::after {  
  box-sizing: border-box;  
}
```

flexbox-07.html:10

flexbox-07.html:7



# Flex: items

Propriedades > align-self

Destinada a alinhar individualmente os *flex items* dentro do container segundo a direção e o sentido do eixo transversal (*cross axis*). A propriedade *align-content* alinha todos os *flex items* de uma só vez dentro do container.

Os valores possíveis são: *flex-start*; *center*; *flex-end*; *stretch* e *baseline*

```
.flex-container{ display: flex; }
.flex-container > :nth-child(3){ align-self: center; } /* o valor padrão é auto */
```



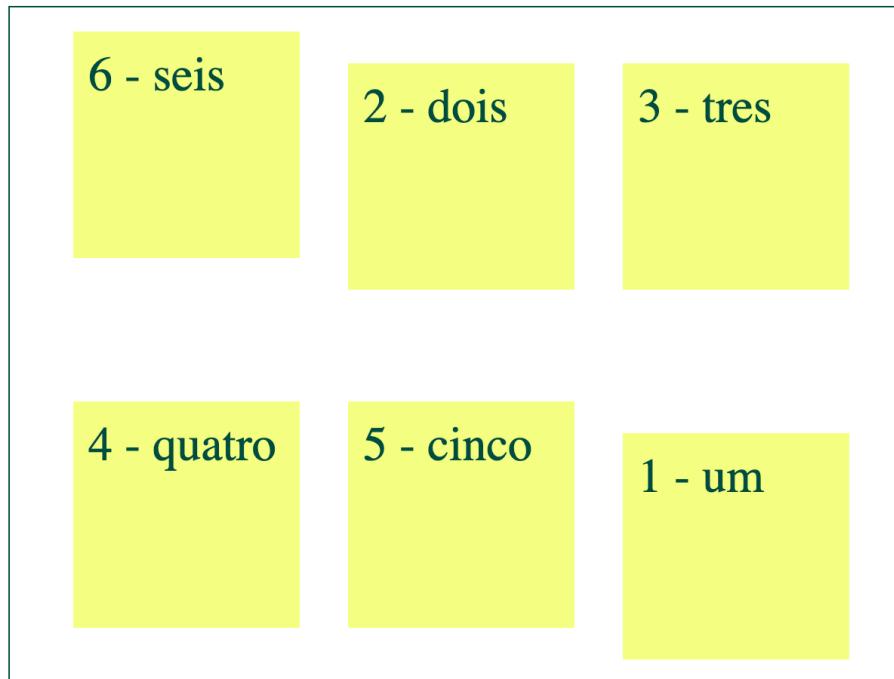
# flexbox-08.html

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8"/>
5      <title>Flexbox - 08</title>
6      <style>
7          *,:before,:after{box-sizing: border-box;}
8          body{margin: 3em;}
9          header h1{margin-bottom: 2em;}
10         .flex-container{
11             margin-left: auto; margin-right: auto;
12             width: 600px; height: 450px; border: 1px solid #004d40;
13             display: flex; justify-content: center; flex-flow: row wrap; align-items: center;
14         }
15         div{color: #004d40; font-size: 2em; padding: 10px; background-color: #f4ff81; width: 150px;
16             height: 150px; margin: 16px;}
17             main > :nth-child(6){order: -1; align-self: flex-start;}
18             main > :nth-child(1){order: 1; align-self: flex-end;}
19         </style>
20     </head>
21     <body>
22         <header><h1>Flexbox 08 - <code>align-self</code></h1></header>
23         <main class="flex-container">
24             <div>1 - um </div><div>2 - dois</div>
25             <div>3 - tres</div><div>4 - quatro</div>
26             <div>5 - cinco</div><div>6 - seis</div>
27         </main>
28     </body>
29 </html>
```



# flexbox-08.html

## Flexbox 08 - align-self



DevTools is now available in Portuguese!

Always match Chrome's language [Switch DevTools to Portuguese](#)

[Don't show again](#)

Elements Console Sources [»](#) [⚙️](#) [⋮](#) [×](#)

```
<!DOCTYPE html>
<html lang="pt-br">
  <head> ...
  </head>
  <body>
    <main class="flex-container">
      <div>1 - um</div>
      <div>2 - dois</div>
      <div>3 - tres</div>
      <div>4 - quatro</div>
      <div>5 - cinco</div>
      <div>6 - seis</div>
    </main>
  </body>
</html>
```

html body main.flex-container

Styles Computed Layout Event Listeners [»](#)

Filter :hov .cls + ↻

```
element.style { }
.flex-container {
  margin-left: auto;
  margin-right: auto;
  width: 600px;
  height: 450px;
  border: 1px solid #004d40;
  display: flex;
  justify-content: center;
  flex-flow: row wrap;
  align-items: center;
}
*, *::before, *::after {
  box-sizing: border-box;
```

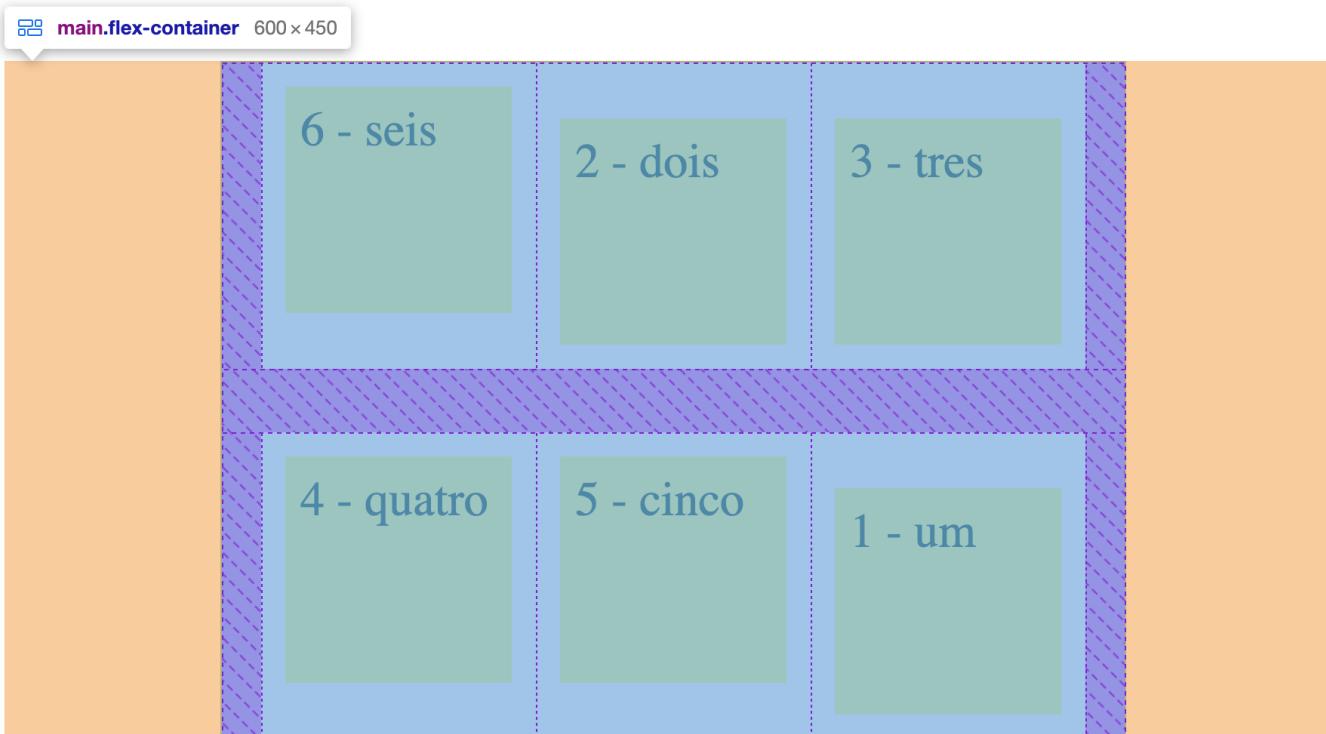
flexbox-08.html:10

flexbox-08.html:7



# flexbox-08.html

## Flexbox 08 - align-self



The screenshot shows the Chrome DevTools Elements tab. The HTML structure is as follows:

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>...</head>
  <body>
    <header>...</header>
    <main class="flex-container"> ... </main>
  </body>
</html>
```

The main element has a class of "flex-container". The CSS styles for ".flex-container" are:

```
.flex-container {
  margin-left: auto;
  margin-right: auto;
  width: 600px;
  height: 450px;
  border: 1px solid #004d40;
  display: flex;
  justify-content: center;
  flex-flow: row wrap;
  align-items: center;
}
```

The DevTools also shows a message: "DevTools is now available in Portuguese!" and buttons for "Always match Chrome's language" and "Switch DevTools to Portuguese".



# Flex: items

Propriedades > flex-grow

Destinada a definir o fator de crescimento do *flex items* dentro do container quando nele há espaço disponível.

Os valores possíveis são: 0 (zero) ou um número inteiro positivo.  
O valor inicial é 0 (zero).



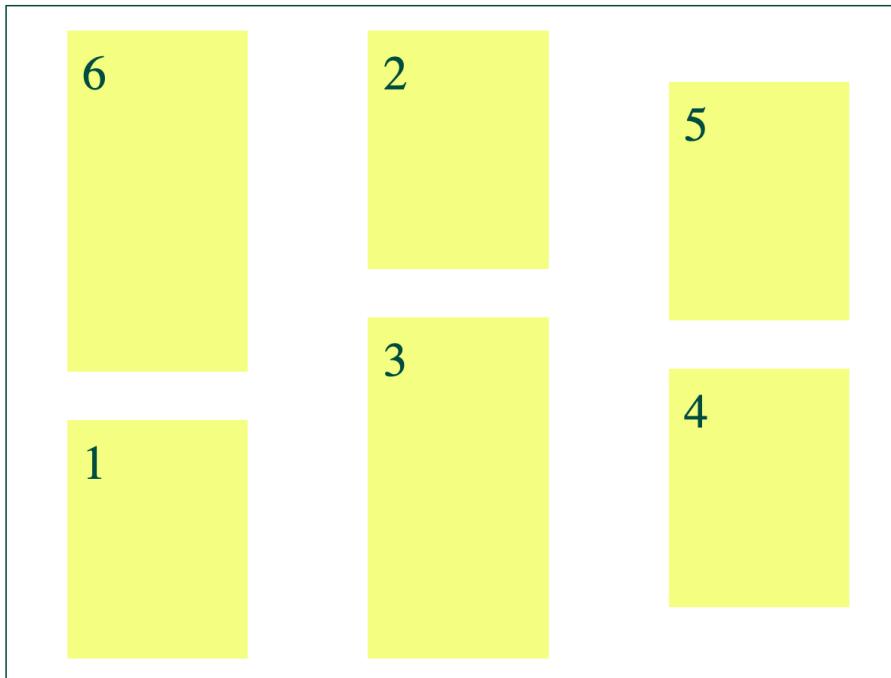
# flexbox-09.html

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8"/>
5      <title>Flexbox - 09</title>
6      <style>
7          *,:*,:before,*,:after{box-sizing: border-box;}
8          body{margin: 3em;}
9          header h1{margin-bottom: 2em;}
10         .flex-container{
11             margin-left: auto; margin-right: auto;
12             width: 600px; height: 450px; border: 1px solid #004d40;
13             display: flex; justify-content: center; flex-flow: column wrap;align-items: center;
14         }
15         div{color: #004d40;font-size: 2em; padding: 10px; background-color: #f4ff81; width: 20%;
16             height: 20vh; margin: 16px;}
17             main > :nth-child(6){order: -1; flex-grow: 1;}
18             main > :nth-child(4){order: 1;}
19             main > :nth-child(3){flex-grow: 1;}
20     </style>
21  </head>
22  <body>
23      <header><h1>Flexbox 09 - <code>flex-grow</code></h1></header>
24      <main class="flex-container">
25          <div>1</div><div>2</div><div>3</div>
26          <div>4</div><div>5</div><div>6</div>
27      </main>
28  </body>
29 </html>
```



# flexbox-09.html

## Flexbox 09 - flex-grow



The screenshot shows the DevTools interface with the 'Elements' tab selected. The DOM tree is displayed, starting with the <html> tag. A blue highlight is applied to the main element, which has a class of 'flex-container'. The 'flex' property is listed under its inline styles. The element itself contains six child div elements, each with a number from 1 to 6.

```
<!DOCTYPE html>
<html lang="pt-br">
  <head> ...
  <body>
    <main class="flex-container">
      <div>1</div>
      <div>2</div>
      <div>3</div>
      <div>4</div>
      <div>5</div>
      <div>6</div>
    </main>
  </body>
</html>
```

At the bottom of the DevTools, the 'flex-container' class is selected in the 'Elements' panel, and its styles are shown in the 'Styles' tab:

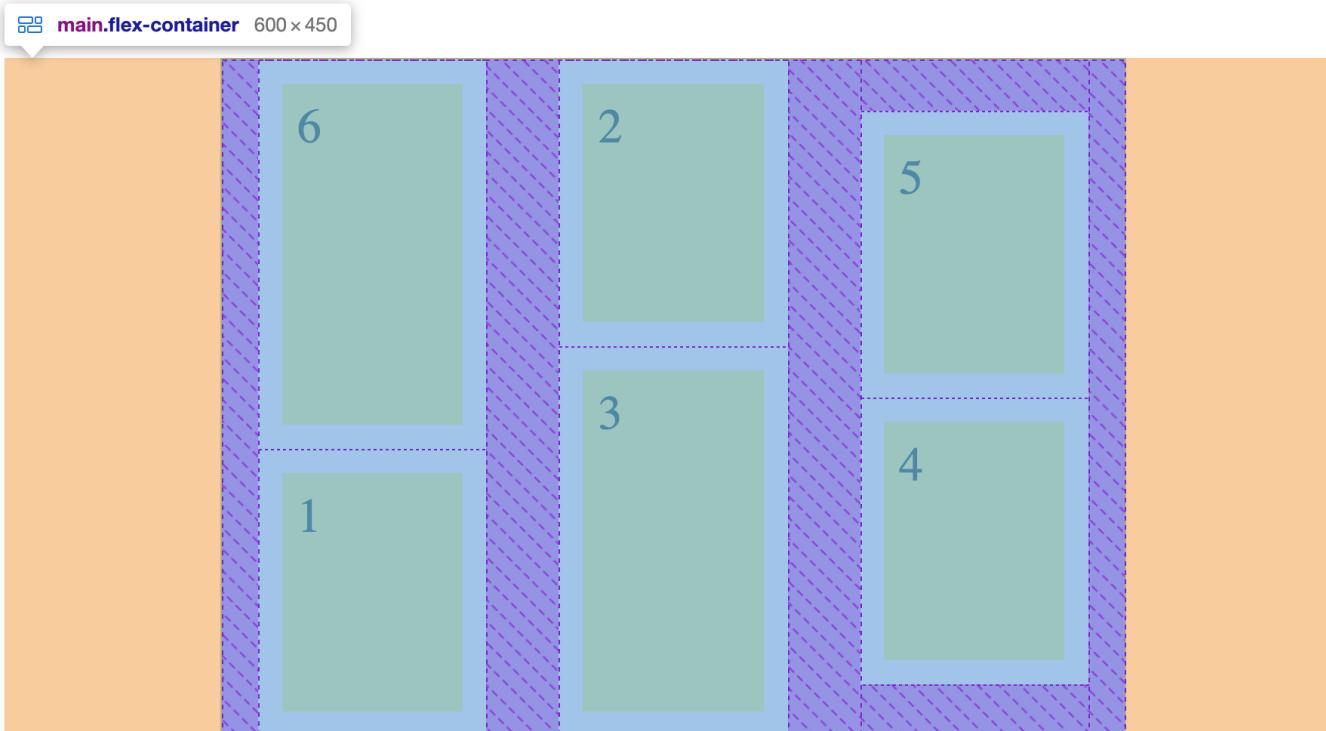
```
.flex-container {
  margin-left: auto;
  margin-right: auto;
  width: 600px;
  height: 450px;
  border: 1px solid #004d40;
  display: flex;
  justify-content: center;
  flex-flow: column wrap;
  align-items: center;
}
```

Other tabs visible in the DevTools include 'Computed', 'Layout', 'Event Listeners', and 'Filter'.



# flexbox-09.html

## Flexbox 09 - flex-grow



DevTools is now available in Portuguese!

Always match Chrome's language [Switch DevTools to Portuguese](#)

Don't show again

Elements Console Sources > | [Settings](#) : X

```
<!DOCTYPE html>
<html lang="pt-br">
  <head> ... </head>
  <body>
    <header> ... </header>
    ... <main class="flex-container"> flex == $0
      <div>1</div>
      <div>2</div>
      <div>3</div>
      <div>4</div>
      <div>5</div>
      <div>6</div>
    </main>
  </body>
</html>
```

html body main.flex-container

Styles Computed Layout Event Listeners >

Filter :hov .cls + ↻

```
element.style {  
}  
.flex-container {  
  margin-left: auto;  
  margin-right: auto;  
  width: 600px;  
  height: 450px;  
  border: 1px solid #004d40;  
  display: flex; grid  
  justify-content: center;  
  flex-flow: column wrap;  
  align-items: center;  
}  
*, *::before, *::after {  
  box-sizing: border-box;  
}
```

flexbox-09.html:10

flexbox-09.html:7



# Flex: items

Propriedades > flex-shrink

Destinada a definir o fator de encolhimento do *flex items* dentro do container quando a largura do container for menor do que a soma das larguras dos *flex items*.

Os valores possíveis são: 0 (zero) ou um número inteiro positivo. O valor inicial é 0 (zero).

Se a *viewport* for redimensionada de modo que o container diminua de largura, por padrão, os *flex items* contraem e vão se acomodando à largura disponível do container.



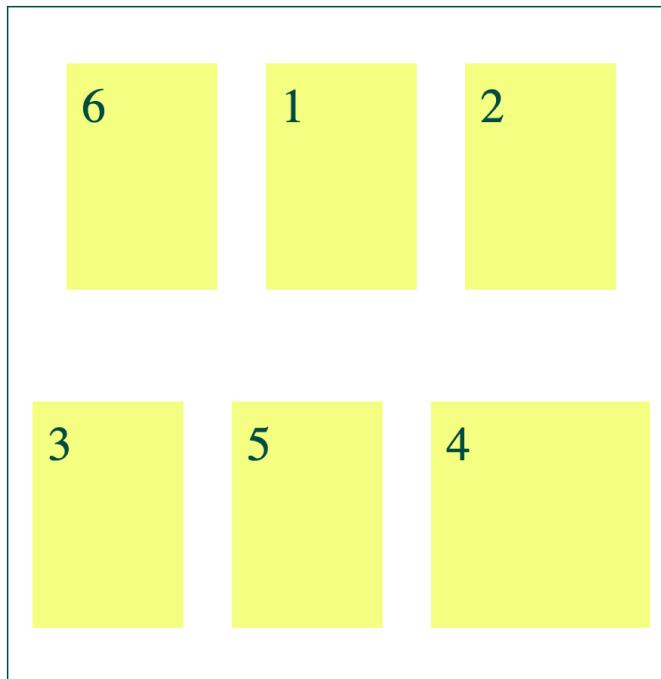
# flexbox-10.html

```
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8"/>
5   <title>Flexbox - 10</title>
6   <style>
7     *,:before,:after{box-sizing: border-box;}
8     body{margin: 3em;}
9     header h1{margin-bottom: 2em;}
10    .flex-container{
11      margin-left: auto; margin-right: auto;
12      width: 50%; height: 450px; border: 1px solid #004d40;
13      display: flex; justify-content: center; flex-flow: row wrap; align-items: center;
14      flex-shrink: 1;
15    }
16    div{color: #004d40; font-size: 2em; padding: 10px; background-color: #f4ff81; width: 100px;
17      height: 150px; margin: 16px;}
18      main > :nth-child(6){order: -1; flex-shrink: 1;}
19      main > :nth-child(4){order: 1; flex-grow: 1;}
20      main > :nth-child(3){ flex-shrink: 1;}
21    </style>
22  </head>
23  <body>
24    <header><h1>Flexbox 10 - <code>flex-shrink</code></h1></header>
25    <main class="flex-container">
26      <div>1</div><div>2</div><div>3</div>
27      <div>4</div><div>5</div><div>6</div>
28    </main>
29  </body>
29 </html>
```



# flexbox-10.html

## Flexbox 10 - flex-shrink



DevTools is now available in Portuguese!

Always match Chrome's language [Switch DevTools to Portuguese](#)

[Don't show again](#)

Elements Console Sources > | [Settings](#) : [Help](#) X

```
<!DOCTYPE html>
<html lang="pt-br">
  <head> ...
  </head>
  <body>
    <header> ...
    </header>
    ... <main class="flex-container"> (flex) == $0
      <div>1</div>
      <div>2</div>
      <div>3</div>
      <div>4</div>
      <div>5</div>
      <div>6</div>
    </main>
  </body>
</html>
```

html body main.flex-container

Styles Computed Layout Event Listeners >

Filter :hov .cls + ↻

```
element.style { }
.flex-container {
  margin-left: auto;
  margin-right: auto;
  width: 50%;
  height: 450px;
  border: 1px solid #004d40;
  display: flex;
  justify-content: center;
  flex-flow: row wrap;
  align-items: center;
  flex-shrink: 1; ⓘ
}
*, *::before, *::after { }
```

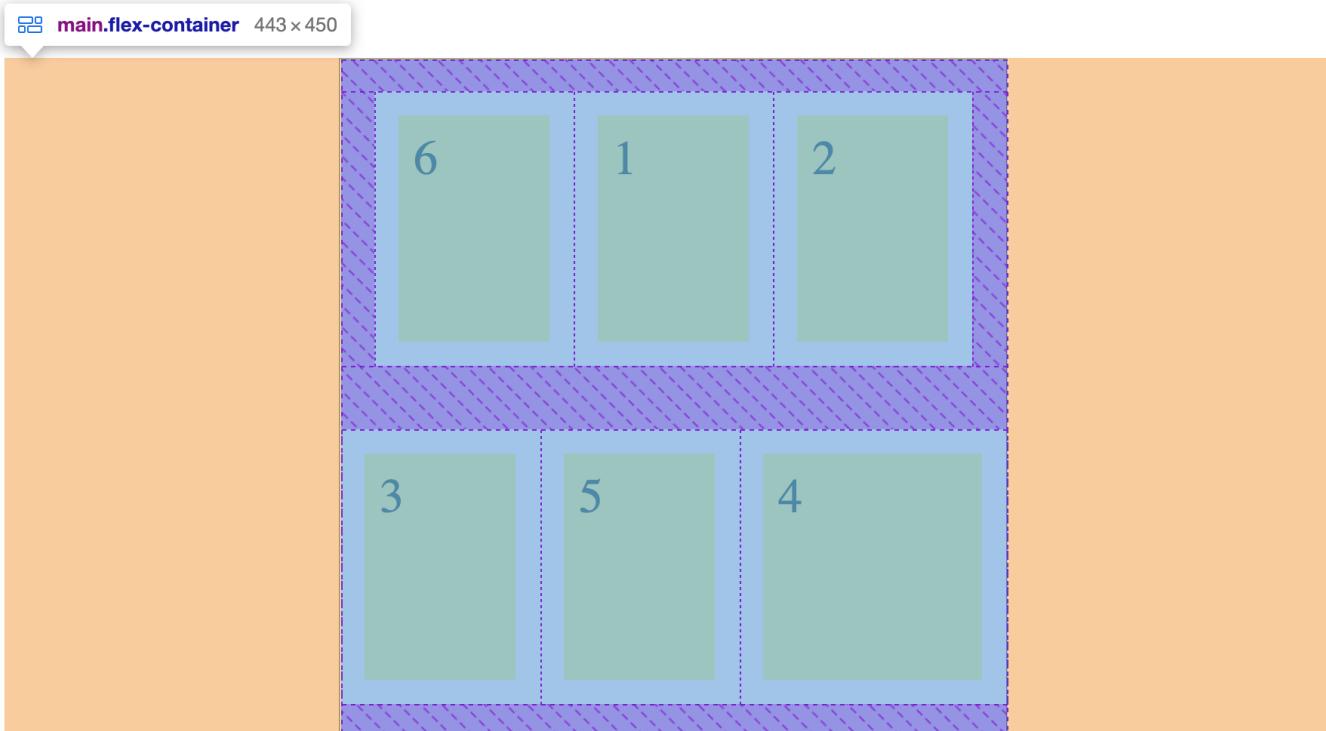
flexbox-10.html:10

flexbox-10.html:7



# flexbox-10.html

## Flexbox 10 - flex-shrink



DevTools is now available in Portuguese!

Always match Chrome's language [Switch DevTools to Portuguese](#)

Don't show again

Elements Console Sources > | [Settings](#) : X

```
<!DOCTYPE html>
<html lang="pt-br">
  <head> ... </head>
  <body>
    <header> ... </header>
    ... <main class="flex-container"> flex == $0
      <div>1</div>
      <div>2</div>
      <div>3</div>
      <div>4</div>
      <div>5</div>
      <div>6</div>
    </main>
  </body>
</html>
```

html body main.flex-container

Styles Computed Layout Event Listeners >

Filter :hov .cls + ↻

```
element.style { }
.flex-container {
  margin-left: auto;
  margin-right: auto;
  width: 50%;
  height: 450px;
  border: 1px solid #004d40;
  display: flex;
  justify-content: center;
  flex-flow: row wrap;
  align-items: center;
  flex-shrink: 1; ⓘ
}
*, *::before, *::after { }
```

flexbox-10.html:10

flexbox-10.html:7



# Flex: items

Propriedades > flex-basis

Destinada a definir a dimensão inicial do *flex items* dentro do container no sentido do eixo principal (*main axis*) largura ou altura.

Os valores possíveis são: a palavra-chave *content* ou qualquer valor válido para definir a propriedade *width* ou *height* CSS, tais como 100px, 40em, 70% ou 40vw.

O valor inicial é *auto*.



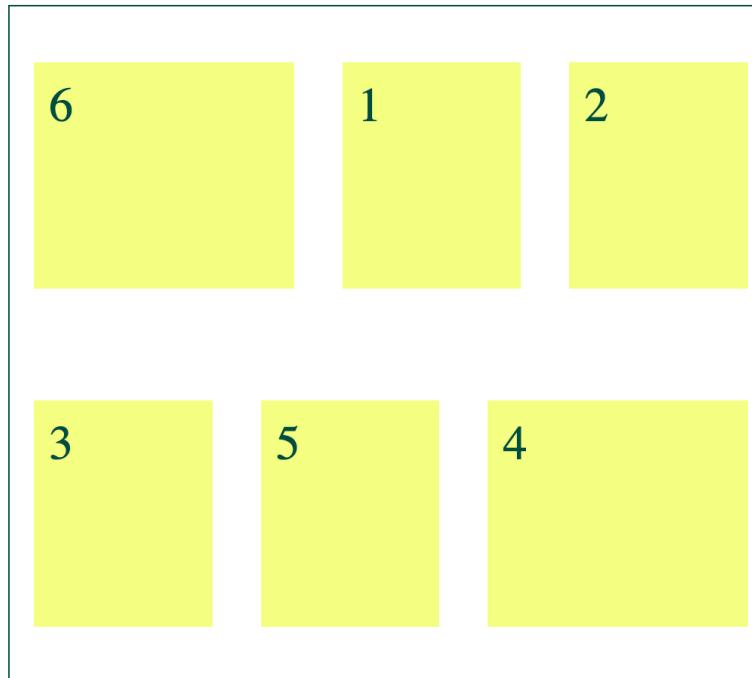
# flexbox-11.html

```
3 <head>
4   <meta charset="UTF-8"/>
5   <title>Flexbox - 11</title>
6   <style>
7     *,:before,:after{box-sizing: border-box;}
8     body{margin: 3em;}
9     header h1{margin-bottom: 2em;}
10    .flex-container{
11      margin-left: auto; margin-right: auto;
12      width: 50%; height: 450px; border: 1px solid #004d40;
13      display: flex; justify-content: center; flex-flow: row wrap; align-items: center;
14    }
15    div{
16      color: #004d40; font-size: 2em; padding: 10px; background-color: #f4ff81;
17      width: 150px; height: 150px; margin: 16px; flex-basis: 15vh;
18    }
19    main > :nth-child(6){order: -1; flex-grow: 1;}
20    main > :nth-child(4){order: 1; flex-grow: 1;}
21  </style>
22 </head>
23 <body>
24   <header><h1>Flexbox 11 - <code>flex-basis</code></h1></header>
25   <main class="flex-container">
26     <div>1</div><div>2</div><div>3</div>
27     <div>4</div><div>5</div><div>6</div>
28   </main>
29 </body>
30 </html>
```



# flexbox-11.html

## Flexbox 11 - flex-basis



i DevTools is now available in Portuguese!

Always match Chrome's language

Switch DevTools to Portuguese

Don't show again

Elements

```
<!DOCTYPE html>
<html lang="pt-br">
  <head> ... </head>
  <body>
    <header> ... </header>
    ... <main class="flex-container"> flex
      == $0
      <div>1</div>
      <div>2</div>
      <div>3</div>
      <div>4</div>
      <div>5</div>
      <div>6</div>
    </main>
  </body>
</html>
```

html body main.flex-container

Styles Computed Layout >>

Filter :hov .cls + ↻

element.style {  
}

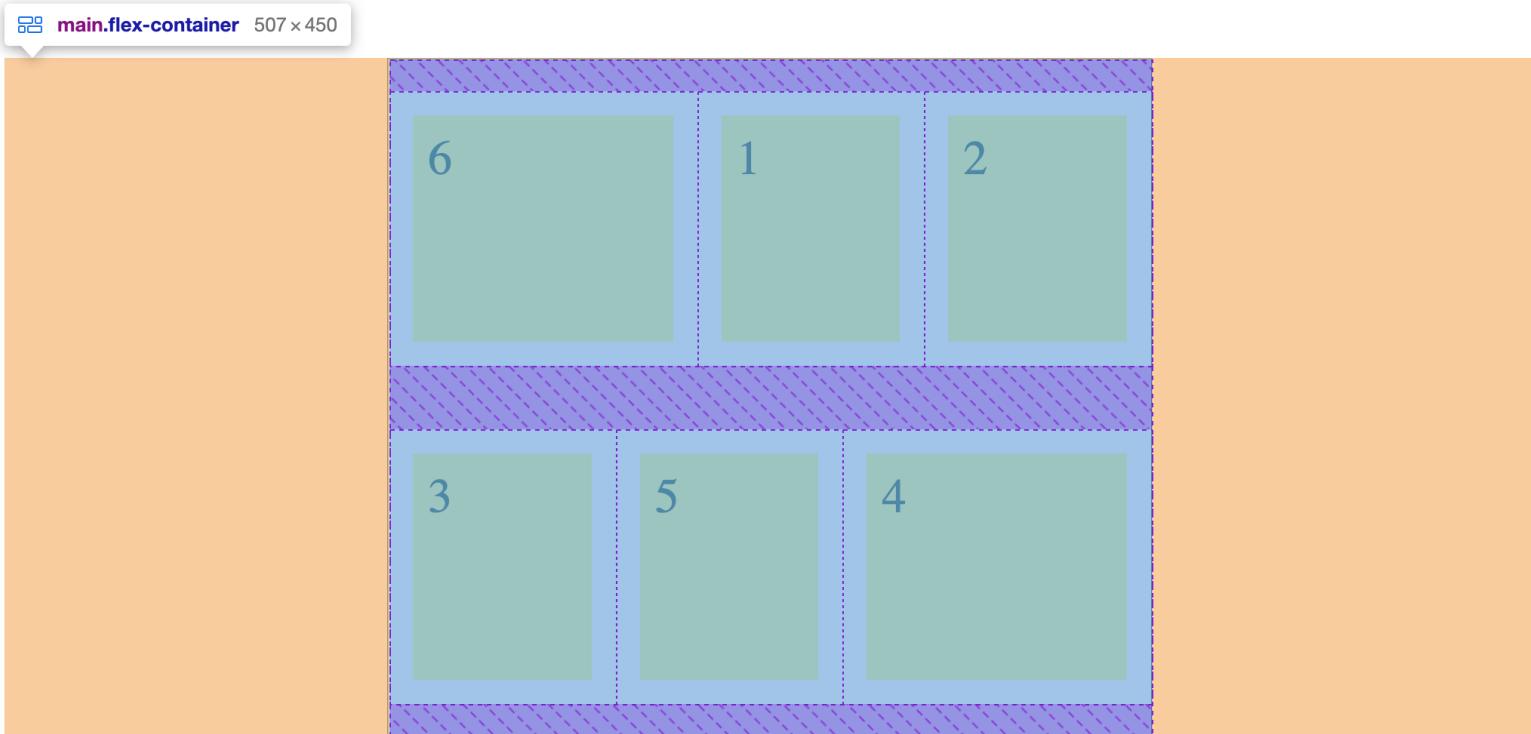
.flex-container { flexbox-11.html:10  
 margin-left: auto;  
 margin-right: auto;  
 width: 50%;  
 height: 450px;  
 border: 1px solid #004d40;  
 display: flex; ⚡  
 justify-content: center;  
 flex-flow: row wrap;  
 align-items: center;  
}

\*, \*::before, \*::after { flexbox-11.html:7  
 box-sizing: border-box;  
}



# flexbox-11.html

## Flexbox 11 - flex-basis



DevTools is now available in Portuguese!

Always match Chrome's language

Switch DevTools to Portuguese

Don't show again

Elements

```
<!DOCTYPE html>
<html lang="pt-br">
  <head> ... </head>
  <body>
    <header> ... </header>
    ... <main class="flex-container"> flex
      == $0
      <div>1</div>
      <div>2</div>
      <div>3</div>
      <div>4</div>
      <div>5</div>
      <div>6</div>
    </main>
  </body>
</html>
```

html body main.flex-container

Styles Computed Layout >>

Filter :hov .cls + ↻

element.style {  
}

.flex-container { flexbox-11.html:10  
 margin-left: auto;  
 margin-right: auto;  
 width: 50%;  
 height: 450px;  
 border: 1px solid #004d40;  
 display: flex; ⚡  
 justify-content: center;  
 flex-flow: row wrap;  
 align-items: center;  
}  
\*, \*::before, \*::after { flexbox-11.html:7  
 box-sizing: border-box;  
}



# Flex: items

## Propriedades > flex

Essa propriedade é a forma abreviada (*shorthand*) de se declarar as propriedades *flex-grow*, *flex-shrink* e *flex-basis* de uma só vez.

O único valor obrigatório dessa declaração é o *flex-basis*.

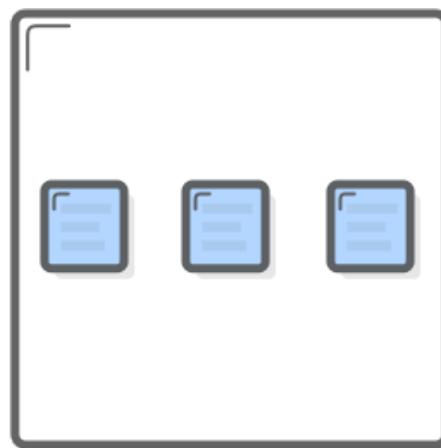
O valor inicial é: 1 0 *auto*. Valores válidos:

- *initial*: 0 1 *auto*
- *auto*: 1 1 *auto*
- *none*: 0 0 *auto*

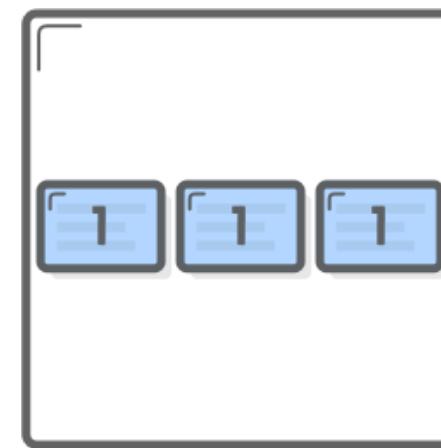


# Flex: items

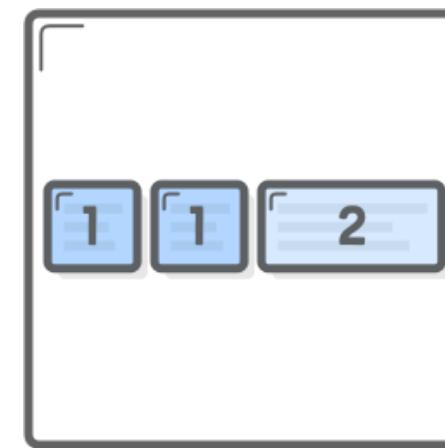
Propriedades > flex



**NO FLEX**



**EQUAL FLEX**

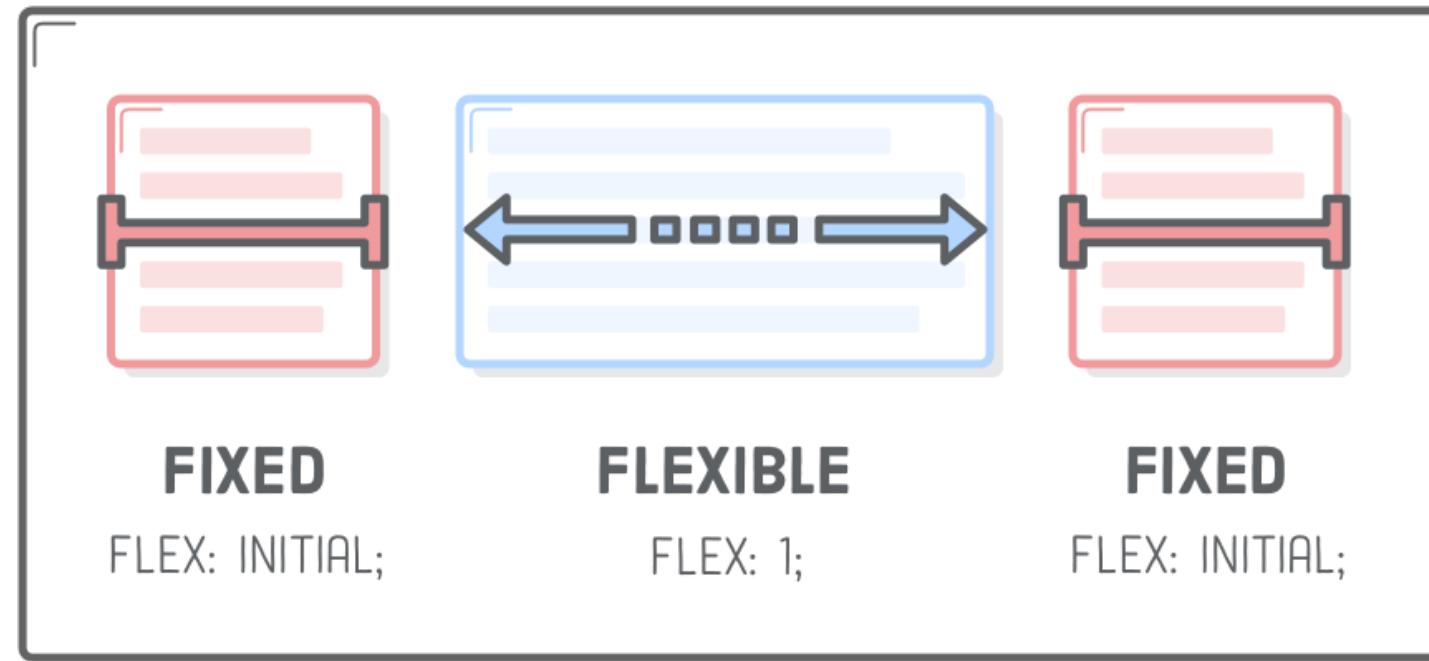


**UNEQUAL FLEX**



# Flex: items

Propriedades > flex





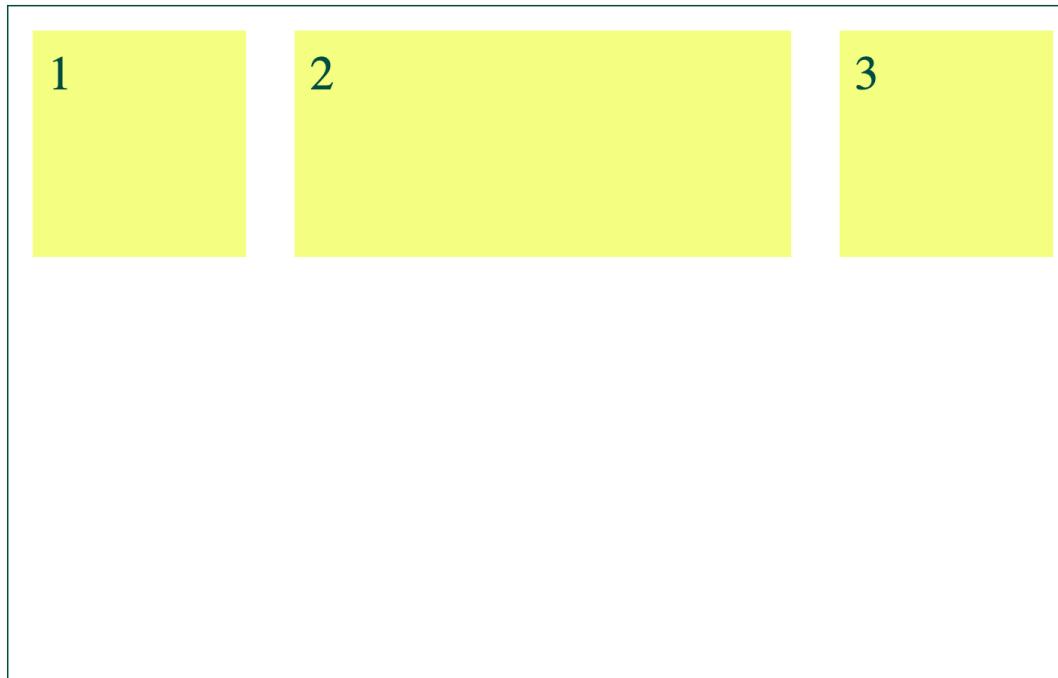
# flexbox-12.html

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4     <meta charset="UTF-8"/>
5     <title>Flexbox - 12</title>
6     <style>
7         *,:*,:before,*,:after{box-sizing: border-box;}
8         body{margin: 3em;}
9         header h1{margin-bottom: 2em;}
10        .flex-container{
11            margin-left: auto; margin-right: auto;
12            width: 70%; height: 450px; border: 1px solid #004d40;
13            display: flex; flex-flow: row wrap;
14        }
15        div{color: #004d40;font-size: 2em; padding: 10px; background-color: #f4ff81;
16            width: 150px; height: 150px; margin: 16px;}
17        main > :nth-child(1),main > :nth-child(3){flex: initial; width: 20%}
18        main > :nth-child(2){flex:1;}
19    </style>
20 </head>
21 <body>
22     <header><h1>Flexbox 12 - <code>flex</code></h1></header>
23     <main class="flex-container">
24         <div>1</div><div>2</div><div>3</div>
25     </main>
26 </body>
27 </html>
```



# flexbox-12.html

## Flexbox 12 - flex



i DevTools is now available in Portuguese!

Always match Chrome's language

Switch DevTools to Portuguese

Don't show again

Elements

```
<!DOCTYPE html>
<html lang="pt-br">
  <head> ... </head>
  <body>
    <header> ... </header>
    ... <main class="flex-container" flex="1">
      <div>1</div>
      <div>2</div>
      <div>3</div>
    </main>
  </body>
</html>
```

html body main.flex-container

Styles Computed Layout >>

Filter :hov .cls + ↻

```
element.style { }
.flex-container { flex-grow: 1; margin-left: auto; margin-right: auto; width: 70%; height: 450px; border: 1px solid #004d40; display: flex; flex-flow: row wrap; }
```

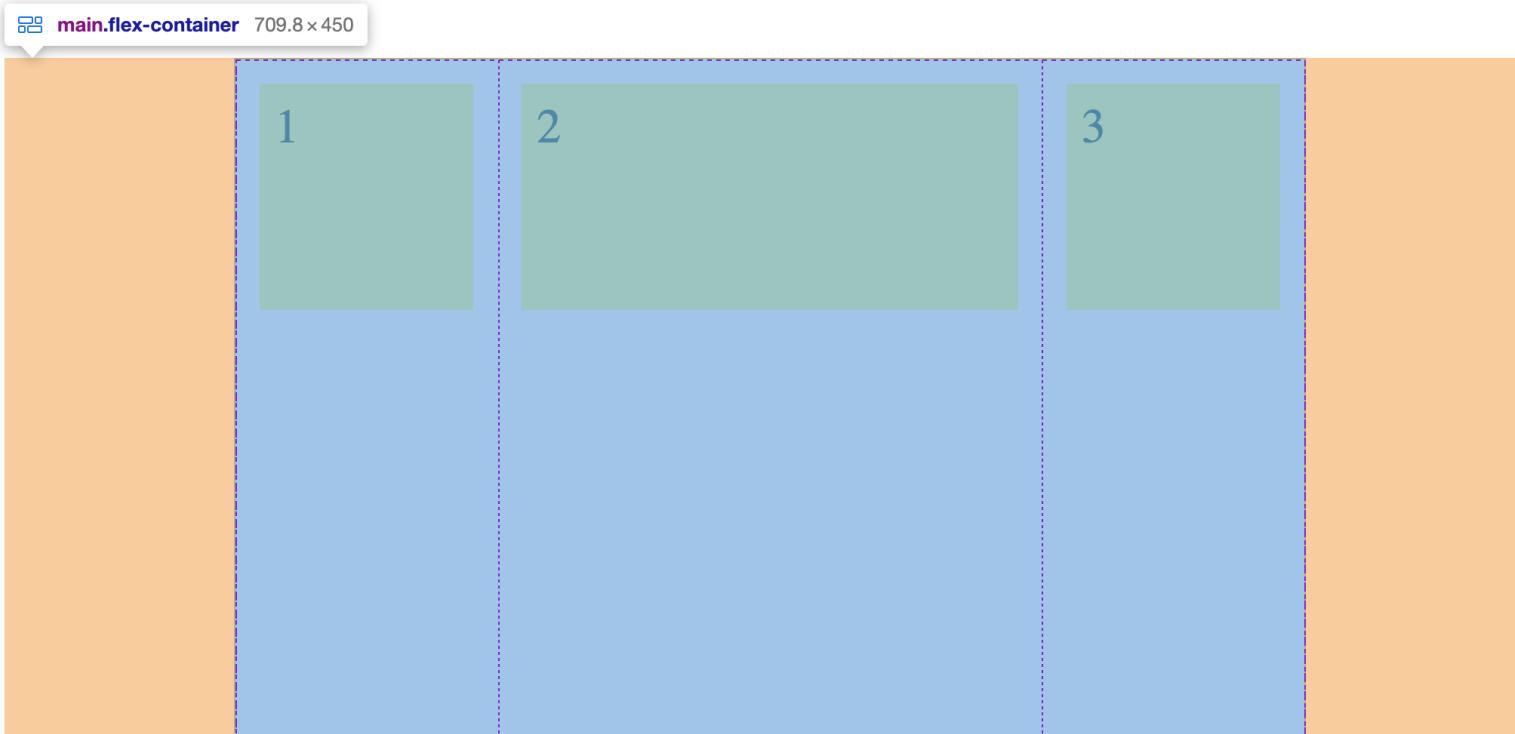
\*, \*::before, \*::after { box-sizing: border-box; }

main { user agent stylesheet }



# flexbox-12.html

## Flexbox 12 - flex



DevTools is now available in Portuguese!

Always match Chrome's language

Switch DevTools to Portuguese

Don't show again

Elements

```
<!DOCTYPE html>
<html lang="pt-br">
  <head> ... </head>
  <body>
    <header> ... </header>
    ... <main class="flex-container" flex
          == $0
          <div>1</div>
          <div>2</div>
          <div>3</div>
        </main>
      </body>
    </html>
```

html body main.flex-container

Styles Computed Layout >>

Filter :hov .cls + ↻

```
element.style { } .flex-container { flexbox-12.html:10 margin-left: auto; margin-right: auto; width: 70%; height: 450px; border: 1px solid #004d40; display: flex; flex-flow: row wrap; }
```

```
* , *::before , *::after { flexbox-12.html:7 box-sizing: border-box; }
```

```
main { user agent stylesheet }
```



# Sumário

A propriedade Flexbox nos deu uma tonelada de incríveis novas possibilidades para criar uma página da web.

Comparando essas técnicas com o uso de *floats*, fica claro que *flexbox* é uma opção mais limpa para a diagramação de sites modernos.



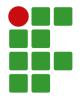
# Sumário

- Use *display: flex* para criar um flex container.
- Use *justify-content* para definir o alinhamento horizontal de itens.
- Use *align-items* para definir o alinhamento vertical de itens.
- Use *flex-direction* se você necessita aninhar colunas (*columns*) dentro de uma linha (*row*).



# Sumário

- Use *row-reverse* ou *column-reverse* para inverter a ordem dos itens.
- Use *order* para customizar/alterar a ordem individual dos elementos HTML.
- Use *align-self* para alinhar individualmente um item.
- Use *flex* para criar um recipiente com itens flexíveis que esticar (*stretch*) ou encolher (*shrink*).



# Dúvidas?





# Bibliografia

- SILVA, Maurício Samy. CSS Grid Layout: Criando layouts CSS profissionais. São Paulo. Novatec, 2017.
- <https://developer.mozilla.org/en-US/docs/Glossary/Flexbox>
- <https://internetingishard.com/html-and-css/flexbox/>
- <http://desenvolvimentoparaweb.com/css/flexbox/>
- <https://www.devmedia.com.br/css3-flexbox-funcionamento-e-propriedades/29532>
- [http://devdocs.io/css/css\\_flexible\\_box\\_layout/basic\\_concepts\\_of\\_flexbox](http://devdocs.io/css/css_flexible_box_layout/basic_concepts_of_flexbox)
- <http://devdocs.io/>
- <https://caniuse.com/#feat=flexbox>