



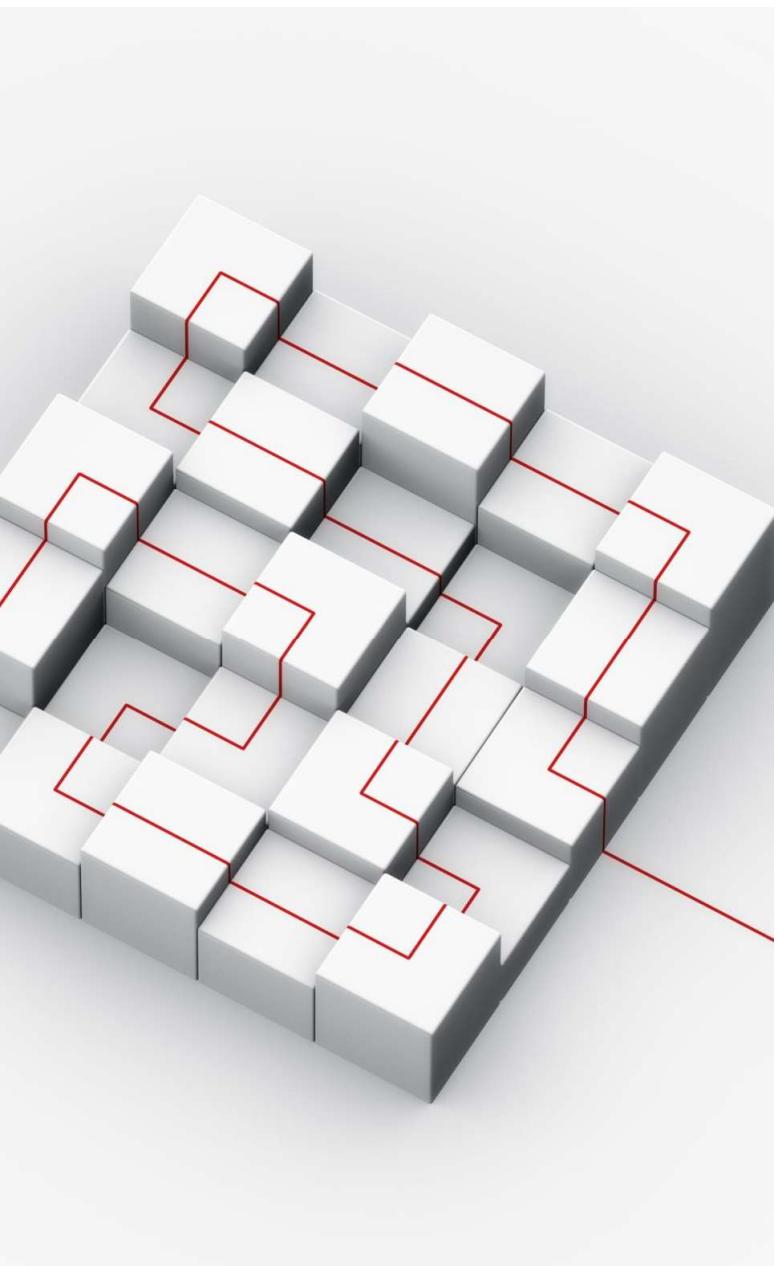
# Flexbox

---

Flexible Box Layout

# Flexbox

- O Flexbox provê ferramentas para criação rápida de layouts complexos e flexíveis, e características que se mostraram historicamente difíceis com CSS;
- É um modelo de layout que torna mais fácil o design de layouts complexos e a reorganização de elementos em diferentes tamanhos de tela;
- É útil quando se pretende criar interfaces responsivas e distribuir espaço de forma eficiente;
- A ideia por trás do flexbox é dar ao container a habilidade de expandir e encolher os elementos para o melhor aproveitamento do espaço disponível.



## Técnicas antigas para criar designs de uma página da web

- Tabelas;
- Inline-block;
- Posicionamento Absolute;
- Float.

# Flexbox

Cheese and wine  
mascarpone halloumi.  
Boursin danish  
fontina cheeseburger  
goat when the cheese  
comes out  
everybody's happy the  
big cheese cut the  
cheese cheese on  
toast. Brie hard  
cheese cheese  
triangles bavarian  
bergkase caerphilly  
chalk and cheese  
dolcelatte boursin.  
Parmesan when the  
cheese comes out  
everybody's happy.

Taleggio cut the  
cheese everyone  
loves. Babybel  
cheesecake brie  
roquefort parmesan  
paneer cauliflower  
cheese paneer.  
Cheese slices who  
moved my cheese  
smelly cheese  
halloumi cheese and  
biscuits monterey  
jack cheese and wine  
when the cheese  
comes out  
everybody's happy.  
Blue castello st. agur  
blue cheese.

Emmental say cheese  
emmental. Port-salut  
cream cheese boursin  
cheeseburger when the  
cheese comes out  
everybody's happy cheese  
and wine cheese on toast  
mascarpone. The big  
cheese feta melted cheese  
hard cheese brie cheese on  
toast pepper jack  
mascarpone. Caerphilly  
taleggio boursin  
mascarpone camembert de  
normandie cheesy feet  
stilton cheeseburger.  
Camembert de normandie  
edam.

Cross axis

Main axis

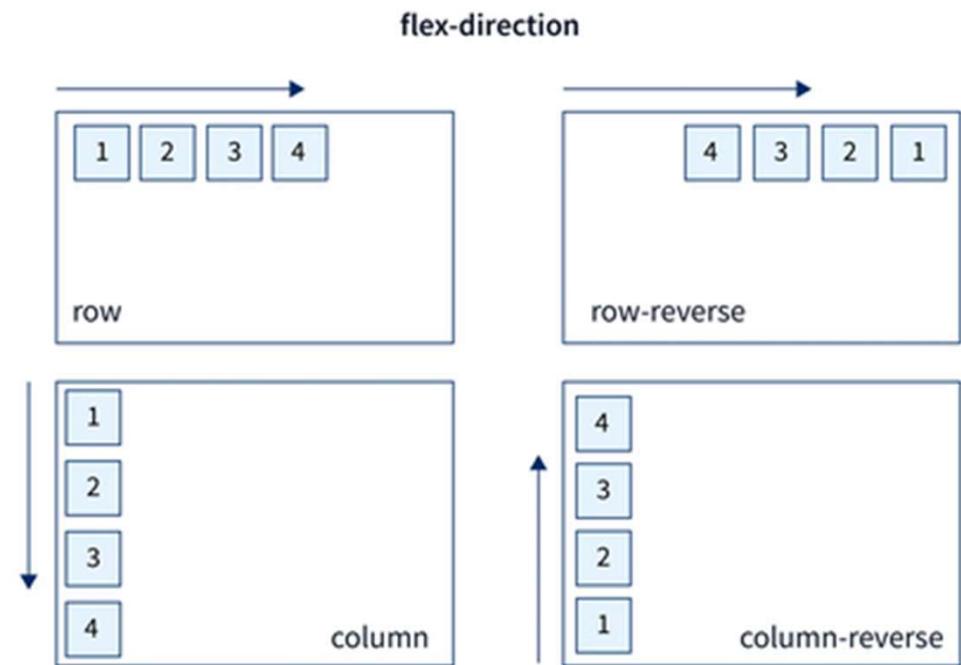
```
.container {  
  display: flex;  
  gap: 30px;  
}
```

```
<body>  
  <div class="container">  
    <div class="A">Cheese and wine mascarpone  
    <div class="B">Taleggio cut the cheese ev  
    <div class="C">Emmental say cheese emment  
  </div>  
</body>
```

# Visão geral Flexbox Property

Container	Item
flex-direction: <b>row</b>   row-reverse   column   column-reverse	align-self: <b>auto</b>   stretch   flex-start   flex-end   center   baseline
flex-wrap: <b>nowrap</b>   wrap   wrap-reverse	order: 0   <integer>
justify-content: <b>flex-start</b>   flex-end   center   space-between   space-around   space-evenly	flex-grow: 0   <integer>
stretch   flex-start   flex-end   center   baseline	flex-shrink: 1   <integer>
stretch   flex-start   flex-end   center   space-between   space-around	flex-basis: <b>auto</b>   <length>

Flex: 0 1 auto



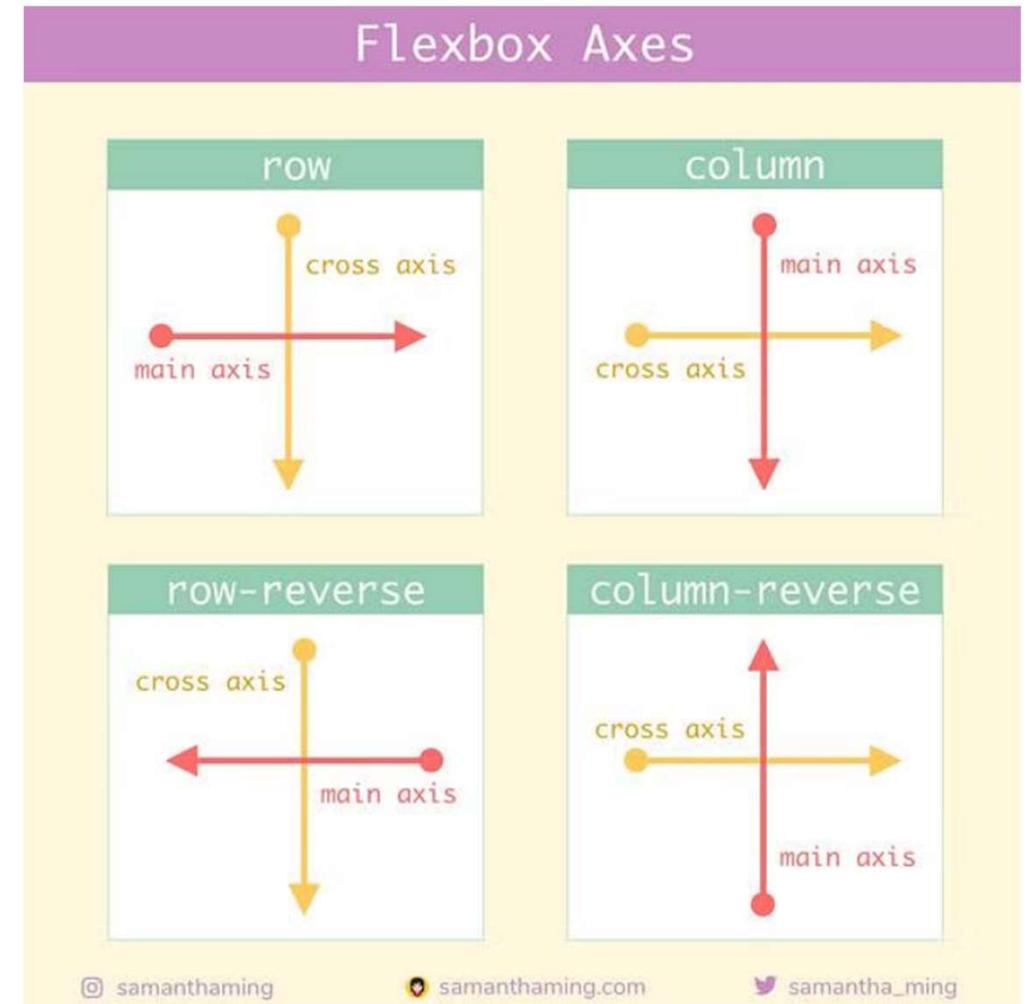
```
.container {  
  display: flex;  
  flex-direction: column;  
  gap: 10px;  
}
```

parent property

# Flex Direction



# Flex Direction



# Flex Direction

The diagram shows a horizontal blue bar representing a flex container. Five orange rectangular boxes are arranged horizontally from left to right, labeled "Coolers", "Processadores", "SSD", "Placas-mãe", and "Placa de Vídeo". A thick yellow arrow points from left to right beneath the first four boxes, with the text "Current main axis" written in yellow below the arrow. Above the diagram, a code editor window is visible, showing the HTML and CSS code for creating this layout.

```
HTML
1 <div class="container">
2   <div class="A">Coolers</div>
3   <div class="B">Processadores</div>
4   <div class="C">SSD</div>
5   <div class="D">Placas-mãe</div>
6   <div class="E">Placa de Vídeo</div>
7 </div>

CSS
1 * {
2   margin: 0;
3   padding: 0;
4   box-sizing: border-box;
5 }
6
7 .container {
8   background-color:#427D9D;
9   display:flex;
10  padding: 10px;
11  flex-direction: row;
12 }
13
14 .A {
```

# Flex Direction

The screenshot shows a code editor with two tabs: "HTML" and "CSS".

**HTML:**

```
1 <div class="container">
2   <div class="A">Coolers</div>
3   <div class="B">Processadores</div>
4   <div class="C">SSD</div>
5   <div class="D">Placas-mãe</div>
6   <div class="E">Placa de Vídeo</div>
7 </div>
```

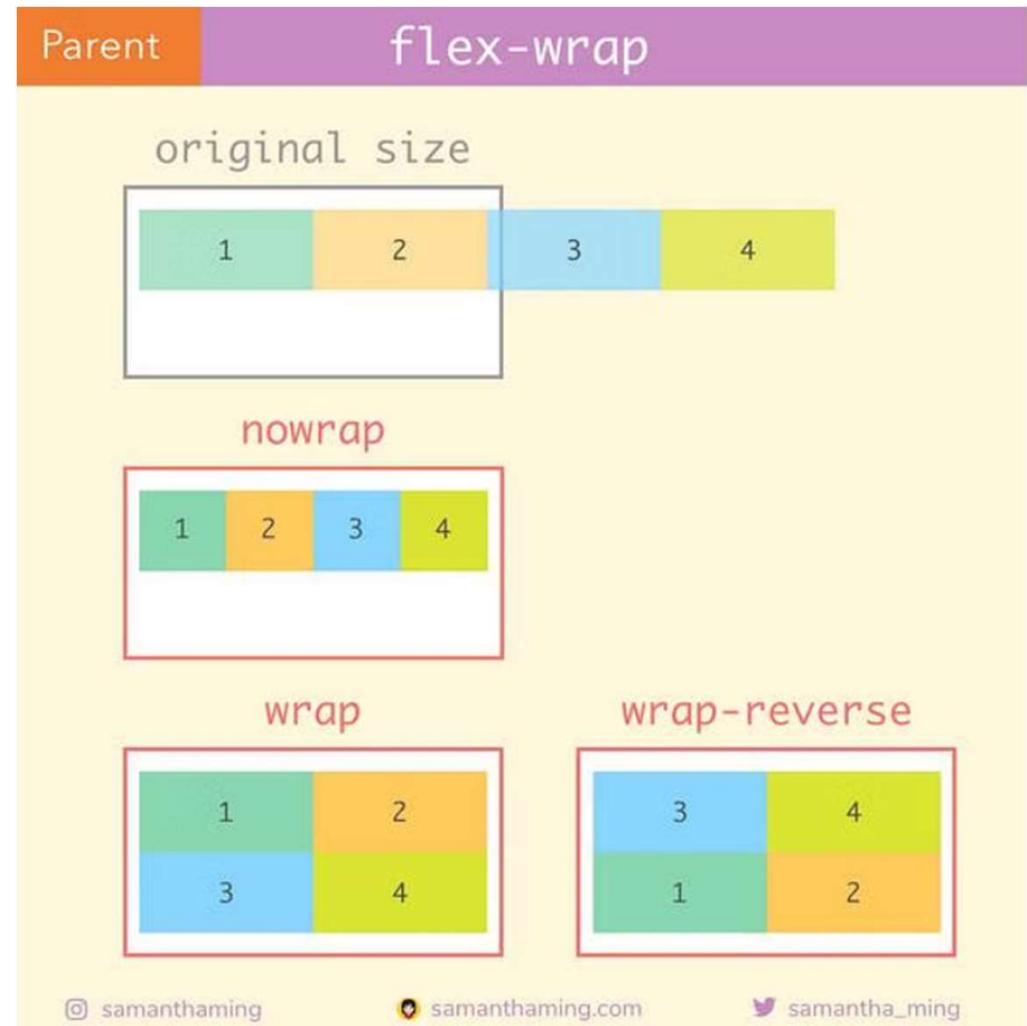
**CSS:**

```
* { margin: 0; padding: 0; box-sizing: border-box; }
.container { background-color: #427D9D; display: flex; padding: 10px; flex-direction: column; }
.A { }
```

Below the code editor, there is a visual representation of the flex items:

- A vertical yellow arrow points downwards, labeled "Current main axis".
- The first item is a blue bar labeled "Coolers".
- The second item is an orange bar labeled "Processadores".
- The third item is an orange bar labeled "SSD".

# Flex-wrap



parent property

# flex-wrap: wrap;

The image shows a code editor interface with two tabs: 'HTML' and 'CSS'. The 'HTML' tab contains the following code:

```
1 <div class="container">
2   <div class="A">1</div>
3   <div class="B">2</div>
4   <div class="C">3</div>
5   <div class="D">4</div>
6   <div class="E">5</div>
7   <div class="F">6</div>
8   <div class="A">7</div>
9   <div class="B">8</div>
10  <div class="C">9</div>
11  <div class="D">10</div>
12  <div class="E">11</div>
13  <div class="F">12</div>
14 </div>
```

The 'CSS' tab contains the following code:

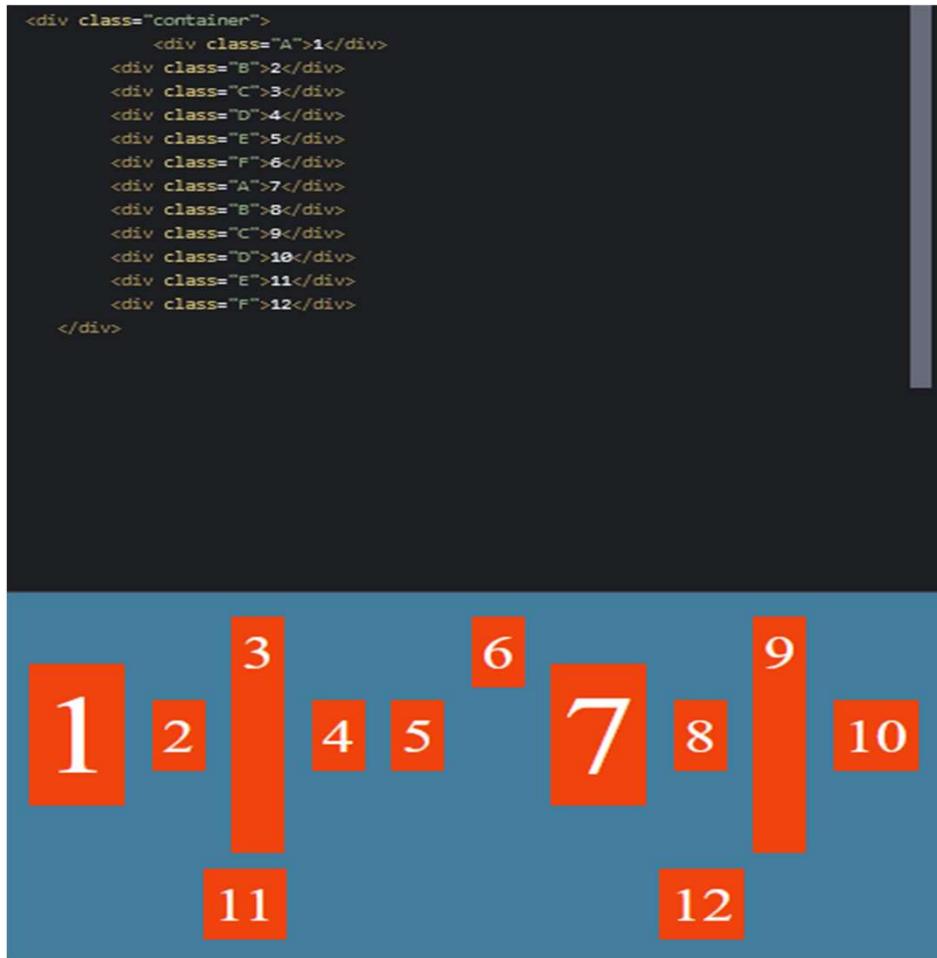
```
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

.container {
  background-color: #427D9D;
  display: flex;
  padding: 10px;
  justify-content: space-around;
  flex-direction: row;
  align-items: center;
  flex-wrap: wrap;
}
```

Below the code editor, there is a visual representation of the flex container's layout. It consists of a blue rectangular background containing twelve orange-red squares, each containing a white number from 1 to 12. The squares are arranged in three rows: the first row has squares 1, 2, and 3; the second row has squares 4, 5, and 6; and the third row has squares 7, 8, 9, 10, 11, and 12. This visual demonstrates how the 'flex-wrap: wrap;' property creates a new line for items after the fourth item.

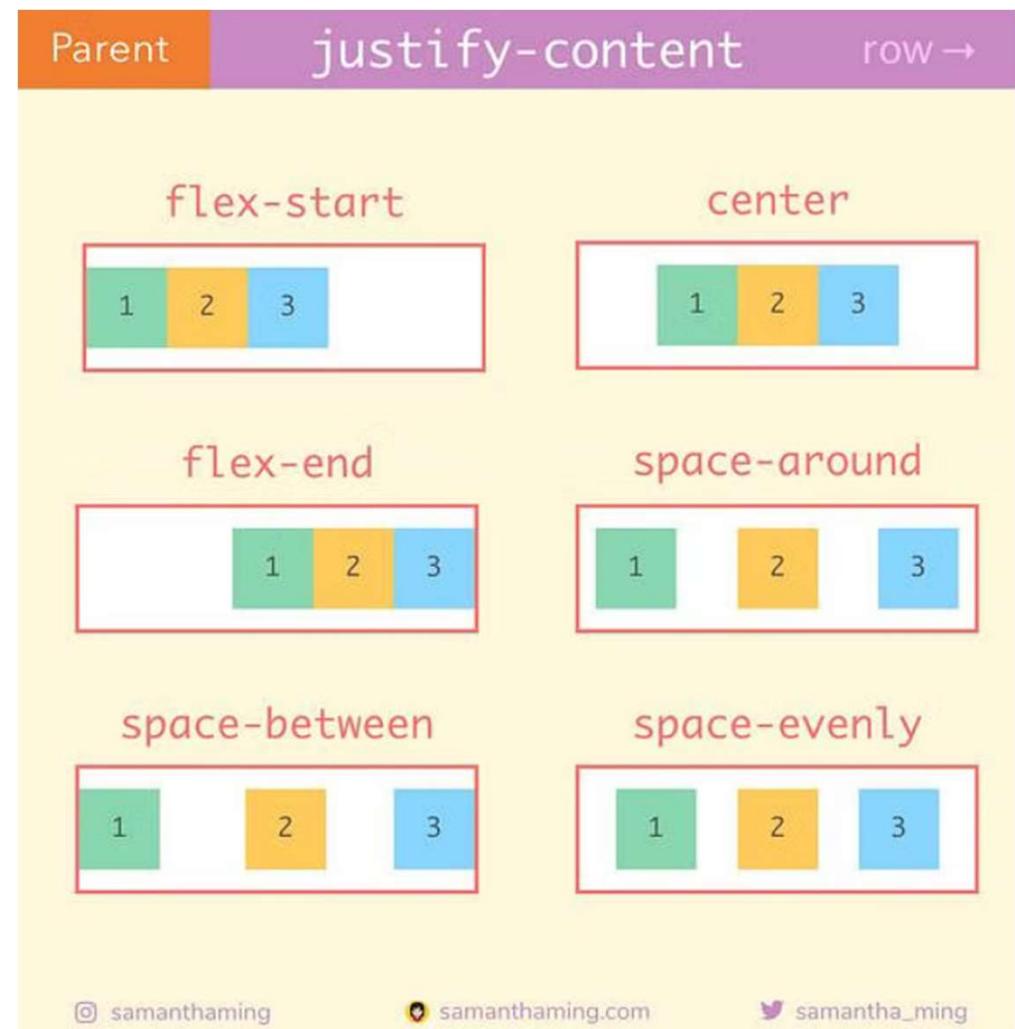
## flex-wrap: wrap;

```
<div class="container">
  <div class="A">1</div>
  <div class="B">2</div>
  <div class="C">3</div>
  <div class="D">4</div>
  <div class="E">5</div>
  <div class="F">6</div>
  <div class="A">7</div>
  <div class="B">8</div>
  <div class="C">9</div>
  <div class="D">10</div>
  <div class="E">11</div>
  <div class="F">12</div>
</div>
```



Viewport diminuiu e os quadros  
11 e 12 pularam (wrap: acondicionou) de linha

# Justify-content



parent property

# justify-content: center;

The image shows a code editor interface with two tabs: 'HTML' and 'CSS'. The 'HTML' tab contains the following code:

```
1 <div class="container">
2   <div class="A">1</div>
3   <div class="B">2</div>
4   <div class="C">3</div>
5   <div class="D">4</div>
6   <div class="E">5</div>
7   <div class="F">6</div>
8 </div>
```

The 'CSS' tab contains the following code:

```
6
7 .container {
8   background-color:#427D9D;
9   display:flex;
10  padding: 10px;
11  flex-direction: row;
12  justify-content: center;
13 }
14
15 .A {
16  padding: 30px;
17  margin: 30px;
18  background-color: #f1430D;
19  color: #FFF;
```

Below the code editor, there is a visual representation of six orange squares arranged horizontally. Each square contains a white number from 1 to 6, corresponding to the div elements in the HTML code.

# justify-content: space-between;

The image shows a code editor interface with two tabs: 'HTML' and 'CSS'. The 'HTML' tab contains the following code:

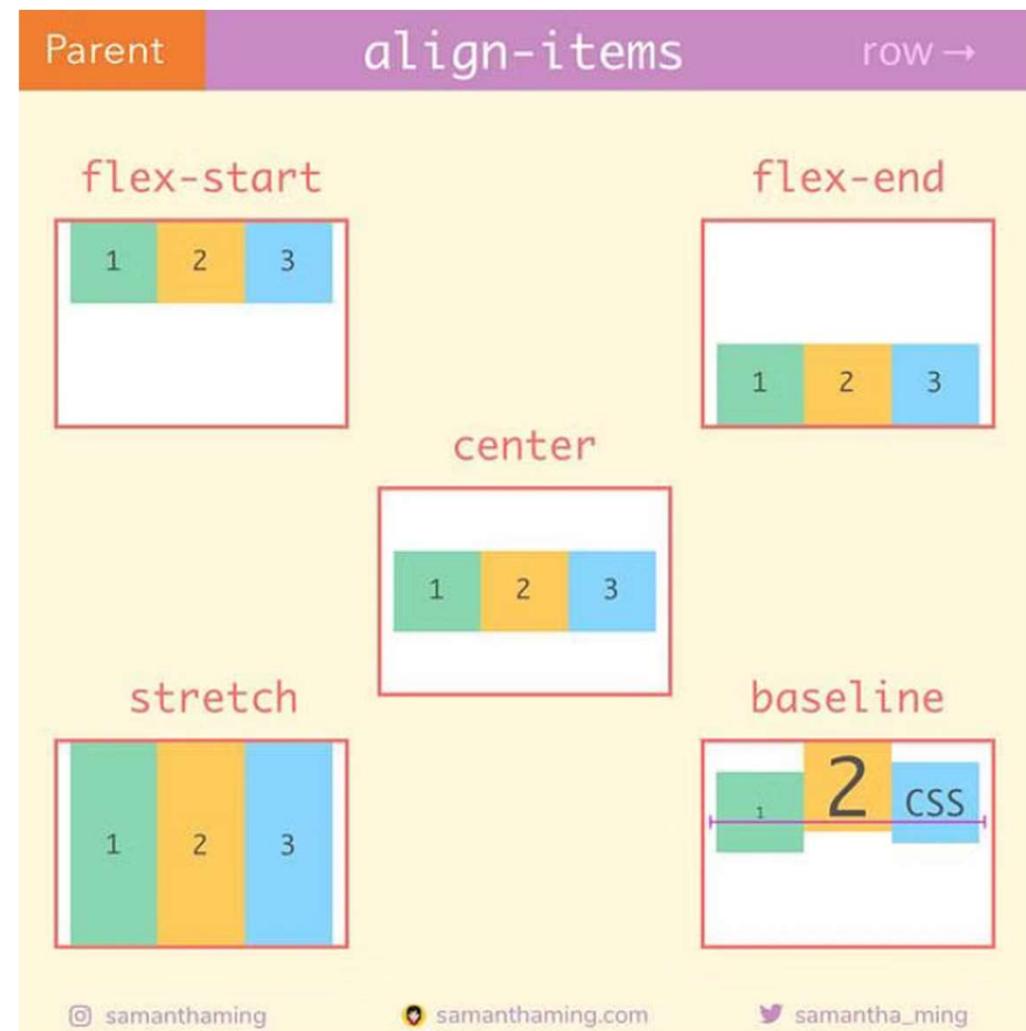
```
1 <div class="container">
2   <div class="A">1</div>
3   <div class="B">2</div>
4   <div class="C">3</div>
5   <div class="D">4</div>
6   <div class="E">5</div>
7   <div class="F">6</div>
8 </div>
```

The 'CSS' tab contains the following code:

```
6
7 .container {
8   background-color:#427D9D;
9   display:flex;
10  padding: 10px;
11  flex-direction: row;
12  justify-content: space-between;
13 }
14
15 .A {
16  padding: 30px;
17  margin: 30px;
18  background-color: #f1430D;
19  color: #FFF;
```

Below the code editor, there is a visual representation of six orange squares arranged horizontally on a blue background. Each square contains a white number from 1 to 6, corresponding to the div elements in the HTML code.

# Align-items



parent property

# align-items: flex-start;

The image shows a code editor interface with two tabs: "HTML" and "CSS".

**HTML Tab:**

```
<div class="container">
  <div class="A">1</div>
  <div class="B">2</div>
  <div class="C">3</div>
  <div class="D">4</div>
  <div class="E">5</div>
  <div class="F">6</div>
</div>
```

**CSS Tab:**

```
.container {
  background-color:#427D9D;
  display:flex;
  padding: 10px;
  flex-direction: row;
  justify-content: center;
  align-items: flex-start;
}

.A {
  padding: 30px;
  margin: 30px;
  background-color: #f1430D;
  color: #FFF;
  font-size: 30px;
}
```

The preview area below the code editor shows a blue rectangular container holding six orange rectangular boxes labeled 1 through 6. Box 3 is significantly taller than the others. The boxes are arranged horizontally in the center of the container.

## Align-items: flex-end;

The image shows a code editor interface with two tabs: "HTML" and "CSS".

**HTML:**

```
1 <div class="container">
2   <div class="A">1</div>
3   <div class="B">2</div>
4   <div class="C">3</div>
5   <div class="D">4</div>
6   <div class="E">5</div>
7   <div class="F">6</div>
8 </div>
```

**CSS:**

```
1 .container {
2   background-color:#427D9D;
3   display:flex;
4   padding: 10px;
5   flex-direction: row;
6   justify-content: center;
7   align-items: flex-end;
8 }
9
10 .A {
11   height: 250px;
12   padding: 30px;
13   margin: 30px;
14   background-color: #f1430D;
15   color: #FFF;
16   font-size: 30px;
17 }
18
19 .B {
20   height: 250px;
21   padding: 30px;
22   margin: 30px;
23   background-color: #f1430D;
24   color: #FFF;
25   font-size: 30px;
26 }
```

The preview area below the code editor shows a blue rectangular container holding six orange rectangular boxes labeled 1 through 6. Box 3 is significantly taller than the others. Boxes 1, 2, 4, 5, and 6 are evenly spaced along the bottom edge of the container, while box 3 is positioned at the bottom right corner, demonstrating the effect of `align-items: flex-end;`.

# align-items: stretch;

The image shows a code editor interface with two tabs: "HTML" and "CSS".

**HTML:**

```
1 <div class="container">
2   <div class="A">1</div>
3   <div class="B">2</div>
4   <div class="C">3</div>
5   <div class="D">4</div>
6   <div class="E">5</div>
7   <div class="F">6</div>
8 </div>
```

**CSS:**

```
1 * {
2   margin: 0;
3   padding: 0;
4   box-sizing: border-box;
5 }
6
7 .container {
8   background-color:#427D9D;
9   display:flex;
10  padding: 10px;
11  flex-direction: row;
12  justify-content: center;
13  align-items: stretch;
14 }
15
16 .A {
17  padding: 30px;
18  margin: 30px;
```

The CSS code includes a global style for all elements and a specific style for the ".container" class. The ".container" class uses a flex layout with "flex-direction: row", "justify-content: center", and "align-items: stretch". It also has a background color of #427D9D, padding of 10px, and a white font color (#FFF). The individual items ("A" through "F") have a padding of 30px and a margin of 30px.

The resulting visual output is a blue horizontal bar containing six orange rectangular boxes labeled 1 through 6. Boxes 1-5 are evenly spaced, while box 6 is positioned at the end of the row.

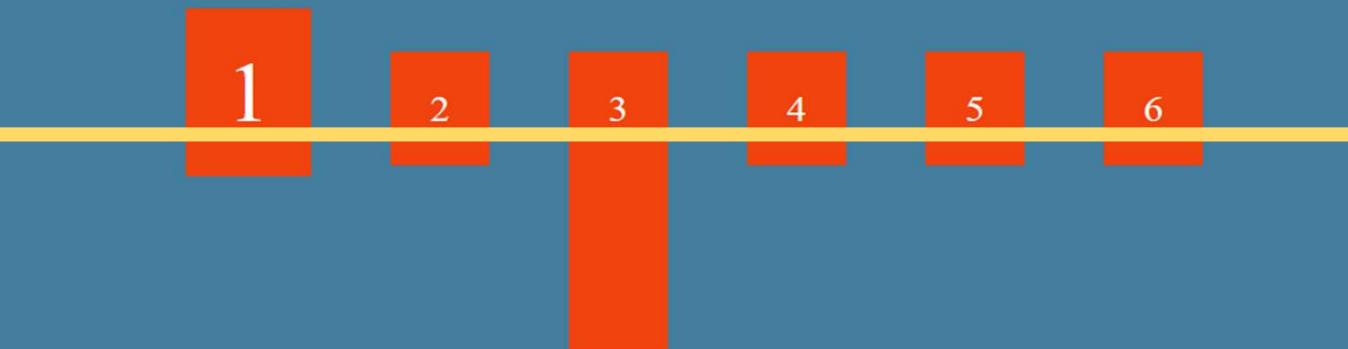
# align-items: baseline;

HTML

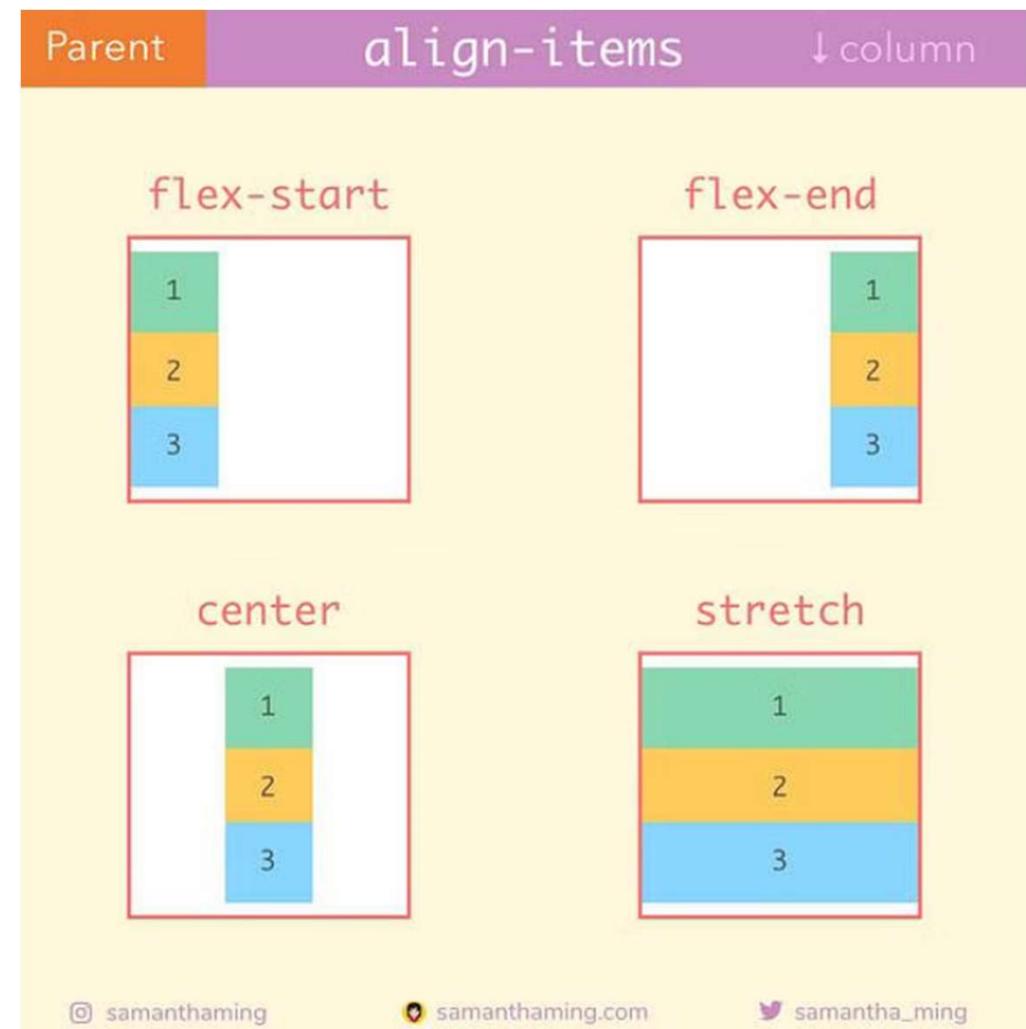
```
1 <div class="container">
2   <div class="A">1</div>
3   <div class="B">2</div>
4   <div class="C">3</div>
5   <div class="D">4</div>
6   <div class="E">5</div>
7   <div class="F">6</div>
8 </div>
```

CSS

```
6
7 .container {
8   background-color:#427D9D;
9   display:flex;
10  padding: 10px;
11  flex-direction: row;
12  justify-content: center;
13  align-items: baseline;
14 }
15
16 .A {
17   padding: 30px;
18   margin: 30px;
19   background-color: #f1430D;
20   color: #FFF;
21   font-size: 70px;
22 }
```



# Align-items



parent property

@ samanthaming

@samanthaming.com

🐦 samantha\_ming

# Flex-Direction: column; align-items: center;

The image shows a code editor interface with two tabs: "HTML" and "CSS".

**HTML:**

```
1 <div class="container">
2   <div class="A">1</div>
3   <div class="B">2</div>
4   <div class="C">3</div>
5   <div class="D">4</div>
6   <div class="E">5</div>
7   <div class="F">6</div>
8 </div>
```

**CSS:**

```
1 .container {
2   background-color:#427D9D;
3   display:flex;
4   padding: 10px;
5   justify-content: center;
6   flex-direction: column;
7   align-items: center;
8 }
```

The preview area below the code editor shows a blue rectangular container with three orange rectangular items labeled 1, 2, and 3. Item 1 is at the top, item 2 is in the middle, and item 3 is at the bottom. A yellow arrow points horizontally from left to right, labeled "Current cross axis".

## Flex-Direction: column; align-items: flex-start;

The image shows a code editor interface with two tabs: 'HTML' and 'CSS'. The 'HTML' tab contains the following code:

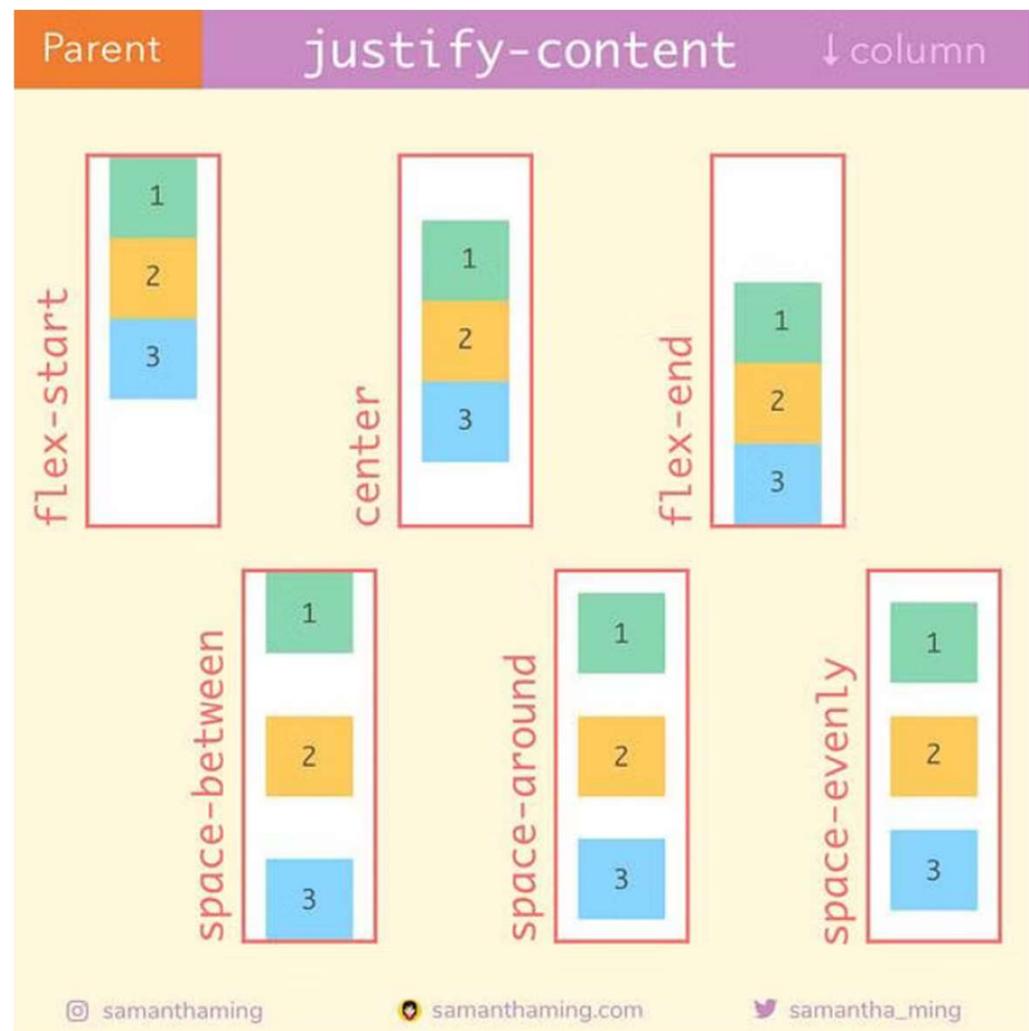
```
<div class="container">
  <div class="A">1</div>
  <div class="B">2</div>
  <div class="C">3</div>
  <div class="D">4</div>
  <div class="E">5</div>
  <div class="F">6</div>
</div>
```

The 'CSS' tab contains the following code:

```
.container {
  background-color:#427D9D;
  display:flex;
  padding: 10px;
  justify-content: center;
  flex-direction: column;
  align-items: flex-start;
}
```

Below the code editor, there is a visual representation of the flex container. It consists of a blue rectangular area containing three orange squares labeled '1', '2', and '3' from top to bottom. A yellow arrow points horizontally to the right, labeled 'Current cross axis'.

# Justify-content



parent property

# flex-direction:column; justify-content: flex-start

The screenshot shows a code editor interface with two tabs: "HTML" and "CSS".

**HTML Tab:**

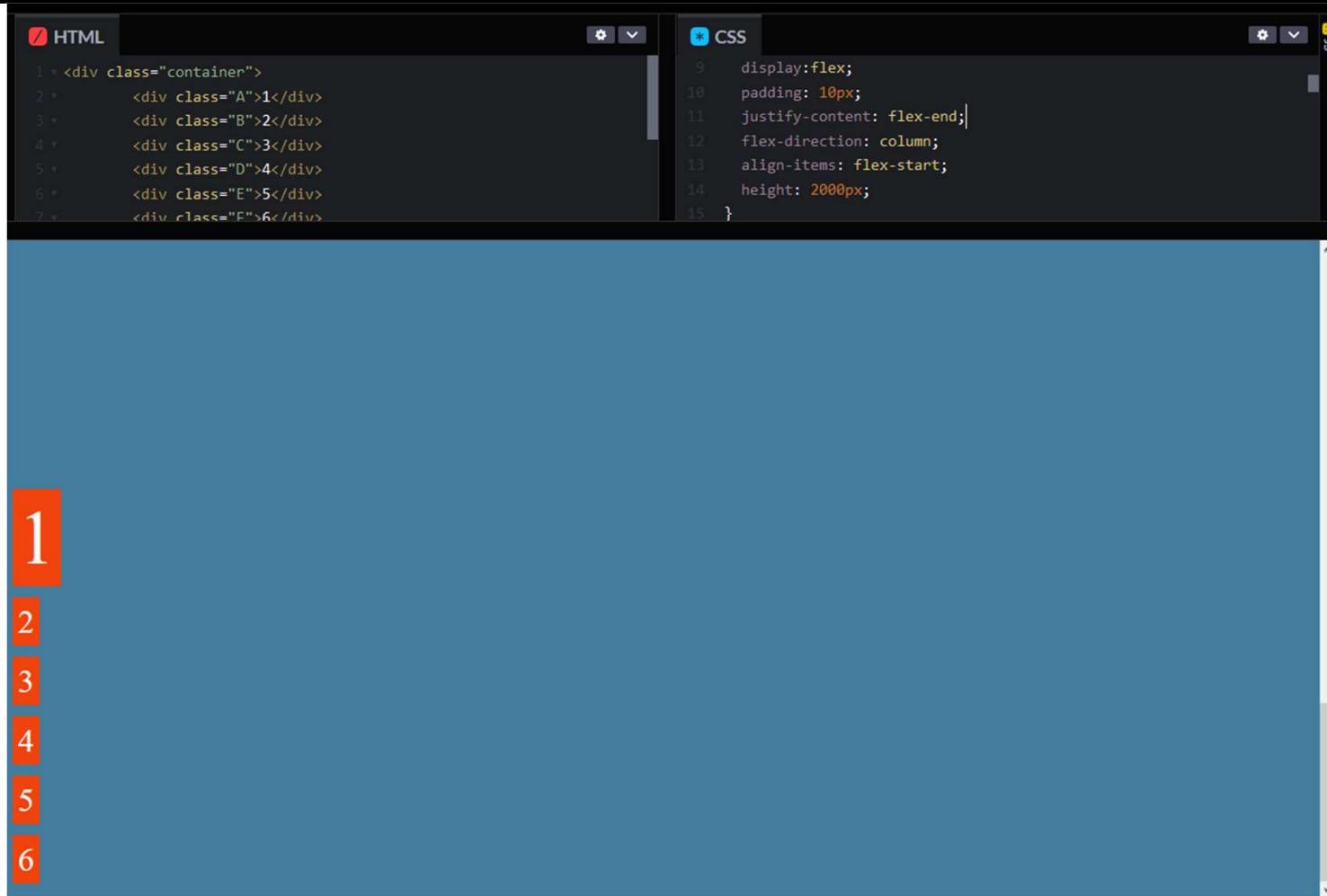
```
<div class="container">
  <div class="A">1</div>
  <div class="B">2</div>
  <div class="C">3</div>
  <div class="D">4</div>
  <div class="E">5</div>
  <div class="F">6</div>
```

**CSS Tab:**

```
display:flex;
padding: 10px;
justify-content: flex-start;
flex-direction: column;
align-items: flex-start;
height: 2000px;
```

The preview area below the tabs shows a blue rectangular container with six red vertical boxes labeled 1 through 6 from top to bottom. This visual representation demonstrates that the items are arranged vertically (flex-direction: column) and positioned at the top of the container (justify-content: flex-start).

flex-direction:column; justify-content: flex-end;



```
HTML
1 <div class="container">
2   <div class="A">1</div>
3   <div class="B">2</div>
4   <div class="C">3</div>
5   <div class="D">4</div>
6   <div class="E">5</div>
7   <div class="F">6</div>

CSS
9  display:flex;
10 padding: 10px;
11 justify-content: flex-end;
12 flex-direction: column;
13 align-items: flex-start;
14 height: 2000px;
15 }
```

The screenshot shows a code editor interface with two tabs: "HTML" and "CSS". The "HTML" tab contains the following code:

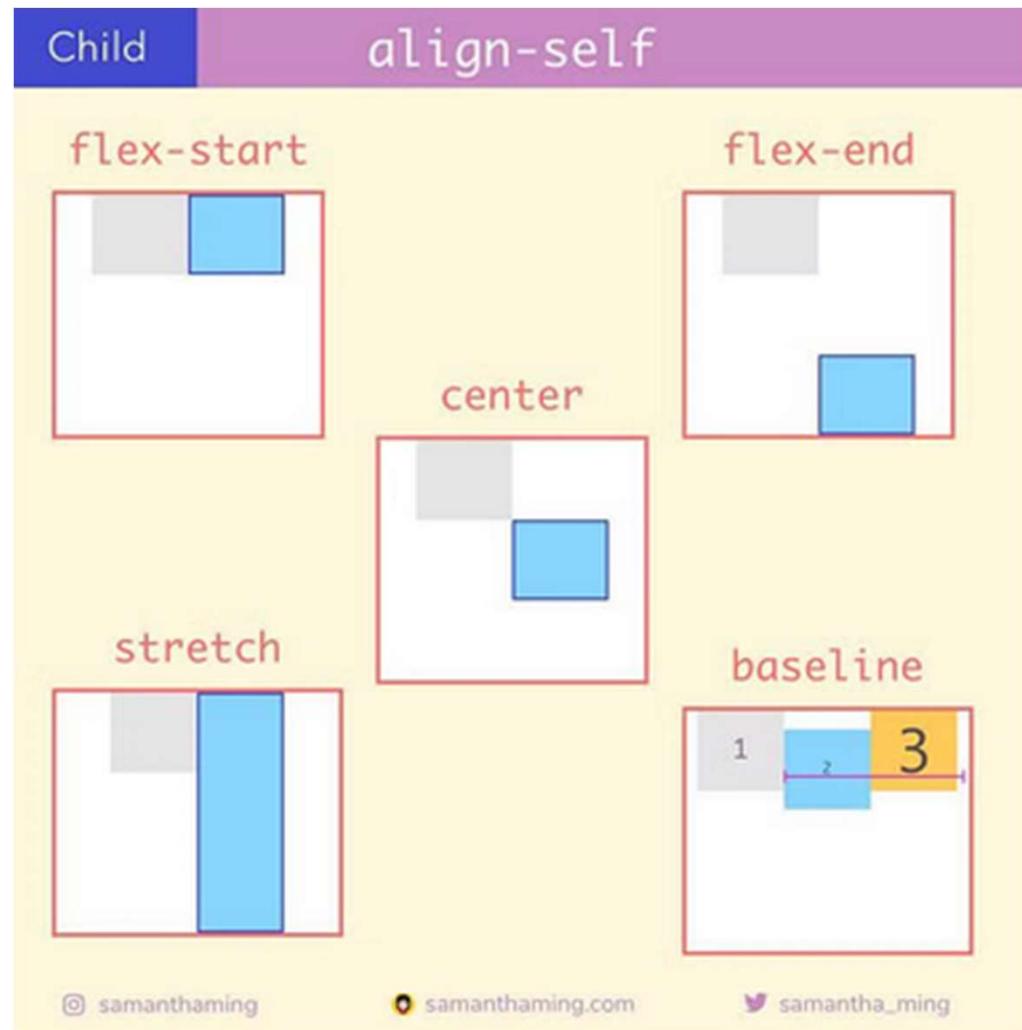
```
<div class="container">
  <div class="A">1</div>
  <div class="B">2</div>
  <div class="C">3</div>
  <div class="D">4</div>
  <div class="E">5</div>
  <div class="F">6</div>
```

The "CSS" tab contains the following code:

```
display:flex;
padding: 10px;
justify-content: flex-end;
flex-direction: column;
align-items: flex-start;
height: 2000px;
```

The resulting visual output is a blue rectangular container with six red vertical boxes labeled 1 through 6. Box 1 is at the top, and box 6 is at the bottom, demonstrating that the items are positioned at the end of the flex-direction axis.

# Align-self



# 

The screenshot shows a code editor interface with two panels: HTML on the left and CSS on the right.

**HTML Panel:**

```
1 <div class="container">
2   <div class="A">1</div>
3   <div class="B">2</div>
4   <div class="C">3</div>
5   <div class="D">4</div>
6   <div class="E">5</div>
7   <div class="F">6</div>
8 </div>
```

**CSS Panel:**

```
1 color: #FFF;
2 font-size: 30px;
3 }
4
5
6
7
8 .F {
9   padding: 5px;
10  margin: 5px;
11  background-color: #f1430D;
12  color: #FFF;
13  font-size: 30px;
14  align-self: flex-start;
15 }
```

Below the code editor, there is a visual representation of the flex container. It consists of six orange rectangular boxes arranged horizontally on a blue background. The boxes are labeled with numbers: 1, 2, 3, 4, 5, and 6. Box 3 is significantly taller than the others, extending vertically across the entire height of the container. Boxes 1, 2, 4, 5, and 6 are aligned vertically at the top of the container, demonstrating the effect of the CSS rule `align-self: flex-start;` applied to the class ".F".

# 

The screenshot shows a code editor interface with two tabs: "HTML" and "CSS".

**HTML:**

```
1 <div class="container">
2   <div class="A">1</div>
3   <div class="B">2</div>
4   <div class="C">3</div>
5   <div class="D">4</div>
6   <div class="E">5</div>
7   <div class="F">6</div>
8 </div>
```

**CSS:**

```
1 .A {
2   font-size: 30px;
3 }
4
5
6
7
8 .B {
9   font-size: 30px;
10 }
11
12
13
14 .C {
15   font-size: 30px;
16 }
17
18
19
20 .D {
21   font-size: 30px;
22 }
23
24
25
26 .E {
27   font-size: 30px;
28 }
29
30
31
32 .F {
33   padding: 5px;
34   margin: 5px;
35   background-color: #f1430D;
36   color: #FFF;
37   font-size: 30px;
38   align-self: flex-end;
39 }
```

The visual output below the editor shows six red rectangular boxes containing the numbers 1 through 6. Box 1 is on the left, box 2 is to its right, box 3 is a tall vertical rectangle further to the right, box 4 is to the right of box 3, box 5 is to the right of box 4, and box 6 is to the right of box 5. This demonstrates that box 6 is aligned at the bottom of the container.

order:  
<integer>

Child	order
1	order:0
2	order:0
3	order:0
4	order:0
1	order:0
3	order:0
4	order:0
2	order:1
2	order:-1
1	order:0
3	order:0
4	order:0

# order: <integer>

The image shows a code editor interface with two tabs: "HTML" and "CSS".

**HTML:**

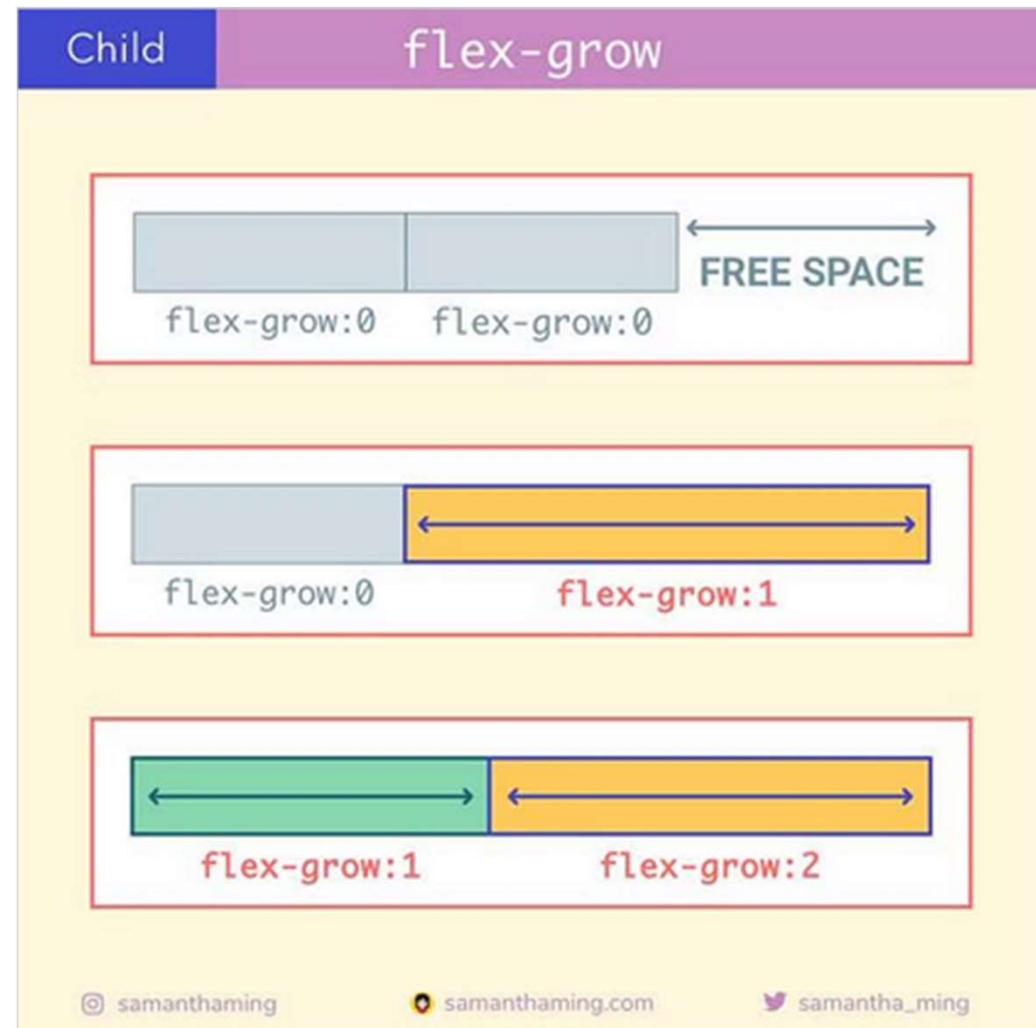
```
1 <div class="container">
2   <div class="A">1</div>
3   <div class="B">2</div>
4   <div class="C">3</div>
5   <div class="D">4</div>
6   <div class="E">5</div>
7   <div class="F">6</div>
8 </div>
```

**CSS:**

```
*, *:before, *:after {
  font-size: 30px;
}
.F {
  padding: 5px;
  margin: 5px;
  background-color: #f1430d;
  color: #FFF;
  font-size: 30px;
  align-self: flex-start;
  order: -1;
}
```

The visual representation below the code editor shows six items labeled 1 through 6. Items 1, 2, 4, and 5 are white rectangles with black outlines. Item 3 is a tall, narrow red rectangle. Item 6 is a small red square.

flex-grow:  
<integer>



# flex-grow: 0;

The image shows a code editor interface with two panels: HTML on the left and CSS on the right. The HTML panel contains the following code:

```
1 <div class="container">
2   <div class="A">1</div>
3   <div class="B">2</div>
4   <div class="C">3</div>
5   <div class="D">4</div>
6   <div class="E">5</div>
7   <div class="F">6</div>
8 </div>
```

The CSS panel contains the following code:

```
7 * .container {
8   background-color:#427D9D;
9   display:flex;
10  padding: 10px;
11  justify-content: space-around;
12  flex-direction: row;
13  align-items: center;
14 }
15
16 * .container>div{
17   flex-grow: 0;
18 }
19
20 * .A {
21   padding: 5px;
22   margin: 5px;
23   background_color: #E1A30D;
24 }
```

Below the code editor, there is a visual representation of the flex container. It consists of a blue rectangular background containing six red rectangular boxes labeled 1 through 6. Box 1 is on the far left, box 6 is on the far right, and boxes 2, 3, 4, and 5 are in the middle, evenly spaced apart.

# flex-grow: 1;

The image shows a code editor interface with two tabs: "HTML" and "CSS".

**HTML Tab:**

```
1 <div class="container">
2   <div class="A">1</div>
3   <div class="B">2</div>
4   <div class="C">3</div>
5   <div class="D">4</div>
6   <div class="E">5</div>
7   <div class="F">6</div>
8 </div>
```

**CSS Tab:**

```
* { margin: 0; padding: 0; }
5 }
6
7 .container {
8   background-color: #427D9D;
9   display: flex;
10  padding: 10px;
11  justify-content: space-around;
12  flex-direction: row;
13  align-items: center;
14 }
15
16 .container>div{
17   flex-grow: 1;
18 }
19
20 .A {
```

Below the code editor, there is a visual representation of the flex container. It consists of six orange rectangular boxes arranged horizontally on a blue background. The boxes are labeled with numbers: 6, 1, 2, 3, 4, and 5 from left to right. Boxes 1, 2, 4, and 5 are significantly wider than boxes 3 and 6, demonstrating how the "flex-grow: 1;" property distributes extra space.

# flex-grow: 2; flex-grow:1;

The image shows a code editor interface with two tabs: "HTML" and "CSS".

**HTML:**

```
1 <div class="container">
2   <div class="A">1</div>
3   <div class="B">2</div>
4   <div class="C">3</div>
5   <div class="D">4</div>
6   <div class="E">5</div>
7   <div class="F">6</div>
8 </div>
```

**CSS:**

```
* { margin: 0; padding: 0; }
*, *:before, *:after { box-sizing: border-box; }

13 .container {
14   display: flex;
15   justify-content: space-around;
16   align-items: center;
17 }
18
19 .A {
20   padding: 5px;
21   margin: 5px;
22   background-color: #f1430d;
23   color: #FFF;
24   font-size: 70px;
25   flex-grow: 2;
26 }
27
28 .B {
29   flex-grow: 1;
30   padding: 5px;
31   margin: 5px;
32 }
```

Below the code editor, there is a visual representation of the flex container. It consists of a blue horizontal bar divided into four orange segments. The first segment contains the number "1", the second contains "2", and the third contains "3", "4", and "5". Red numbers "6" and "3" are positioned at the ends of the first and third segments respectively.

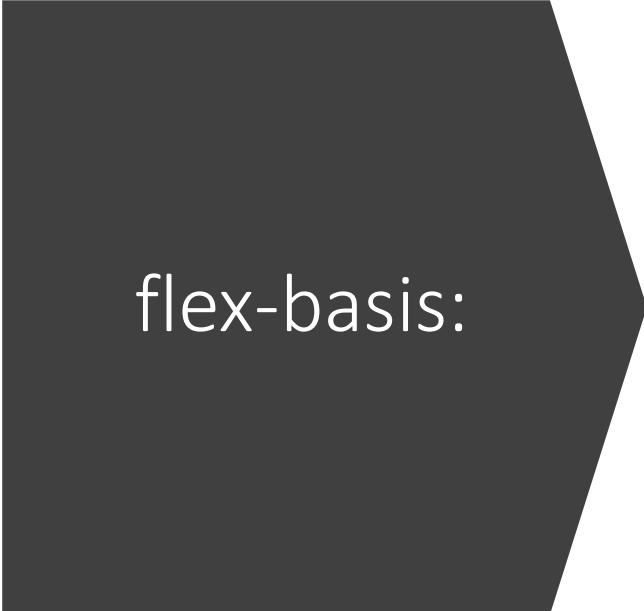
# Flex-basis

- A propriedade flex-basis especifica o comprimento inicial de um item flexível.

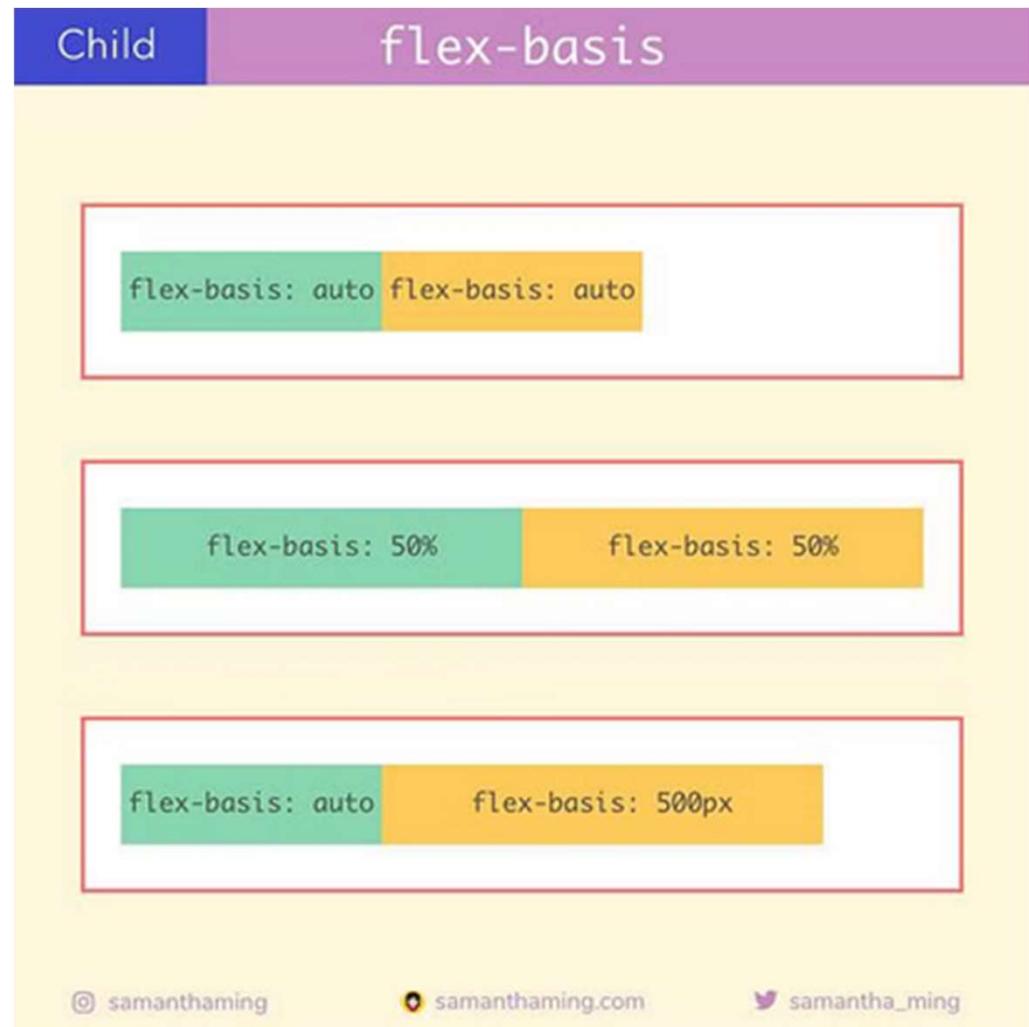
```
.A{  
    background-color: #D0A2F7;  
    text-align: center;  
}  
  
.B{  
    background-color: #DCBFFF;  
    text-align: center;  
}  
  
.C{  
    background-color: #E5D4FF;  
    text-align: center;  
    flex-basis: 150px;  
}  
  
.D{  
    background-color: #F1EAFF;  
    text-align: center;  
}
```

Serviços Graduação Pós-Graduação Extensão

child property



flex-basis:



flex-basis: 30%;

The screenshot shows a code editor interface with two tabs: 'HTML' and 'CSS'. The 'HTML' tab contains the following code:

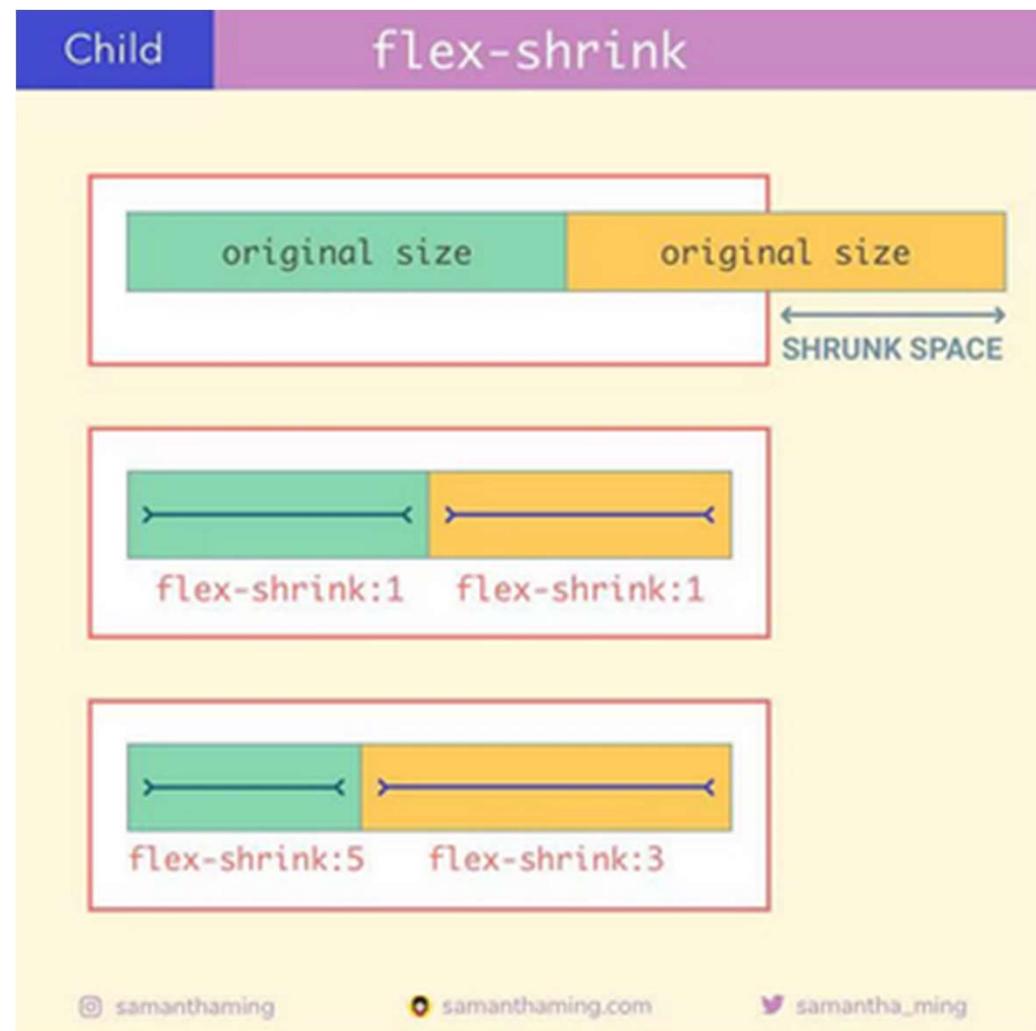
```
1 <div class="container">
2   <div class="A">1</div>
3   <div class="B">2</div>
4   <div class="C">3</div>
5   <div class="D">4</div>
6   <div class="E">5</div>
7   <div class="F">6</div>
8 </div>
```

The 'CSS' tab contains the following code:

```
12 flex-direction: row,
13 align-items: center;
14 }
15
16 .A {
17   padding: 5px;
18   margin: 5px;
19   background-color: #f1430D;
20   color: #FFF;
21   font-size: 70px;
22   flex-grow: 0;
23 }
24
25 .B {
26   flex-basis: 30%; // This line is highlighted in yellow
27   padding: 5px;
28   margin: 5px;
```

Below the code editor, there is a visual representation of the flexbox layout. It consists of six orange rectangular boxes on a blue background, labeled 1 through 6. Box 1 is the widest, followed by box 2, then box 3, and so on. Box 2 has a yellow border around its text, indicating it is the active element.

# flex-shrink



# flex-shrink: 0;

The screenshot shows a code editor interface with two tabs: "HTML" and "CSS".

**HTML:**

```
1 <div class="container">
2   <div class="A">1</div>
3   <div class="B">2</div>
4   <div class="C">3</div>
5   <div class="D">4</div>
6   <div class="E">5</div>
7   <div class="F">6</div>
8 </div>
```

**CSS:**

```
1 .A {
2   margin: 5px;
3   background-color: #f1430D;
4   color: #FFF;
5   font-size: 70px;
6   flex-grow:0;
7 }
8
9 .B {
10   flex-basis: 400px;
11   flex-shrink:0;
12   padding: 5px;
13   margin: 5px;
14   background-color: #f1430D;
15   color: #FFF;
16   font-size: 30px;
17 }
```

The visual representation below the code editor shows six items labeled 1 through 6. Item 2 is a long orange rectangle, item 3 is a tall orange rectangle, and items 1, 4, 5, and 6 are smaller orange squares.

- Item 1: A small orange square.
- Item 2: A very long orange rectangle, representing the element with `flex-basis: 400px`.
- Item 3: A tall orange rectangle, representing the element with `flex-grow: 0`.
- Item 4: A small orange square.
- Item 5: A small orange square.
- Item 6: A small orange square.

## Flex-shrink: 0;

The screenshot shows a code editor with two tabs: 'HTML' and 'CSS'. The 'HTML' tab contains the following code:

```
1 <div class="container">
2   <div class="A">1</div>
3   <div class="B">2</div>
4   <div class="C">3</div>
5   <div class="D">4</div>
6   <div class="E">5</div>
7   <div class="F">6</div>
8 </div>
```

The 'CSS' tab contains the following code:

```
18 margin: 5px;
19 background-color: #f1430D;
20 color: #FFF;
21 font-size: 70px;
22 flex-grow:0;
23 }
24
25 .B {
26   flex-basis: 400px;
27   flex-shrink:0;
28   padding: 5px;
29   margin: 5px;
30   background-color: #f1430D;
31   color: #FFF;
32   font-size: 30px;
33 }
```

Below the code editor, there is a visual representation of the flex container. It consists of a blue rectangular area containing six numbered boxes: '1' (orange), '2' (orange), '3' (orange), '4' (orange), '5' (orange), and '6' (orange). Box '2' is significantly larger than the others. The text to the right of the visualization states: "Viewport diminuiu e o quadro 2 não encolheu".

Viewport diminuiu e o quadro 2 não encolheu

# flex-shrink: 1; flex-shrink: 5;

```
HTML
1 <div class="container">
2   <div class="A">1</div>
3   <div class="B">2</div>
4   <div class="C">3</div>
5   <div class="D">4</div>
6   <div class="E">5</div>
7   <div class="F">6</div>
8 </div>

CSS
24
25 .B {
26   flex-basis: 400px;
27   flex-shrink:1;
28   padding: 5px;
29   margin: 5px;
30   background-color: #f1430D;
31   color: #FFF;
32   font-size: 30px;
33 }
34
35 .C {
36   padding: 5px;
37   flex-basis: 400px;
38   flex-shrink:5;
39   margin: 5px;
40   background-color: #f1430D;
```

6

1

2

3

4

5

# flex-shrink: 1; flex-shrink: 5;

HTML    CSS    JS    Result

```
<div class="container">
  <div class="A">1</div>
  <div class="B">2</div>
  <div class="C">3</div>
  <div class="D">4</div>
  <div class="E">5</div>
  <div class="F">6</div>
</div>
```

Viewport diminuiu e o  
quadro 3 (flex-shrink: 5) encolheu  
bem mais que o quadro 2 (flex-shrink: 1)

# Align-content

Parent	align-content
	flex-start
	center
	flex-end
	space-between
	space-around
	stretch

Diagram illustrating the align-content property in CSS Flexbox. The parent container holds six items (1-6) in a row. The align-content property determines how these items are positioned vertically within the container.

- flex-start:** Items 1, 2, and 3 are aligned to the top of the container, while items 4, 5, and 6 are aligned to the bottom.
- center:** Items 1, 2, and 3 are centered vertically within the container, while items 4, 5, and 6 are aligned to the bottom.
- flex-end:** Items 1, 2, and 3 are aligned to the bottom of the container, while items 4, 5, and 6 are aligned to the top.
- space-between:** Item 1 is aligned to the top, item 4 is in the middle, and item 6 is aligned to the bottom. Items 2 and 3 are placed in the space between item 1 and item 4, and between item 4 and item 6 respectively.
- space-around:** Item 1 is aligned to the top, item 4 is in the middle, and item 6 is aligned to the bottom. Item 2 is to the right of item 1, and item 5 is to the right of item 4. Item 3 is centered between item 2 and item 5.
- stretch:** Item 1 is aligned to the top, item 4 is in the middle, and item 6 is aligned to the bottom. Item 2 is stretched to the width of item 3, and item 5 is stretched to the width of item 3.

align-content: stretch; align-content: flex-start; align-content: flex-end;

HTML

```
<div>
  padding: 0px;
  box-sizing: border-box;
}

.container {
  background-color:#427090;
  display:flex;
  padding: 10px;
  justify-content: space-around;
  flex-direction: row;
  align-items: center;
  flex-wrap: wrap;
  height:500px;
  align-content: stretch;
}

.A {
  padding: 5px;
  margin: 5px;
  background-color: #f14380;
  color: #FFF;
  font-size: 20px;
}
```

CSS

JS

Result

The screenshot shows a code editor interface with tabs for HTML, CSS, JS, and Result. The CSS tab contains the following code:

```
.container {
  background-color:#427090;
  display:flex;
  padding: 10px;
  justify-content: space-around;
  flex-direction: row;
  align-items: center;
  flex-wrap: wrap;
  height:500px;
  align-content: stretch;
}

.A {
  padding: 5px;
  margin: 5px;
  background-color: #f14380;
  color: #FFF;
  font-size: 20px;
}
```

The Result tab shows a blue background with 12 red rectangular boxes arranged in two rows. The first row contains items 1 through 10. The second row contains items 11 and 12. Each box contains a white number corresponding to its index.

HTML

```
<div>
  padding: 0px;
  box-sizing: border-box;
}

.container {
  background-color:#427090;
  display:flex;
  padding: 10px;
  justify-content: space-around;
  flex-direction: row;
  align-items: center;
  flex-wrap: wrap;
  height:500px;
  align-content: flex-start;
}

.A {
  padding: 5px;
  margin: 5px;
  background-color: #f14380;
  color: #FFF;
  font-size: 20px;
}
```

CSS

JS

Result

The screenshot shows a code editor interface with tabs for HTML, CSS, JS, and Result. The CSS tab contains the following code:

```
.container {
  background-color:#427090;
  display:flex;
  padding: 10px;
  justify-content: space-around;
  flex-direction: row;
  align-items: center;
  flex-wrap: wrap;
  height:500px;
  align-content: flex-start;
}

.A {
  padding: 5px;
  margin: 5px;
  background-color: #f14380;
  color: #FFF;
  font-size: 20px;
}
```

The Result tab shows a blue background with 12 red rectangular boxes arranged in two rows. The first row contains items 1 through 10. The second row contains items 11 and 12. Each box contains a white number corresponding to its index.

HTML

```
<div>
  padding: 0px;
  box-sizing: border-box;
}

.container {
  background-color:#427090;
  display:flex;
  padding: 10px;
  justify-content: space-around;
  flex-direction: row;
  align-items: center;
  flex-wrap: wrap;
  height:500px;
  align-content: flex-end;
}

.A {
  padding: 5px;
  margin: 5px;
  background-color: #f14380;
  color: #FFF;
  font-size: 20px;
}
```

CSS

JS

Result

The screenshot shows a code editor interface with tabs for HTML, CSS, JS, and Result. The CSS tab contains the following code:

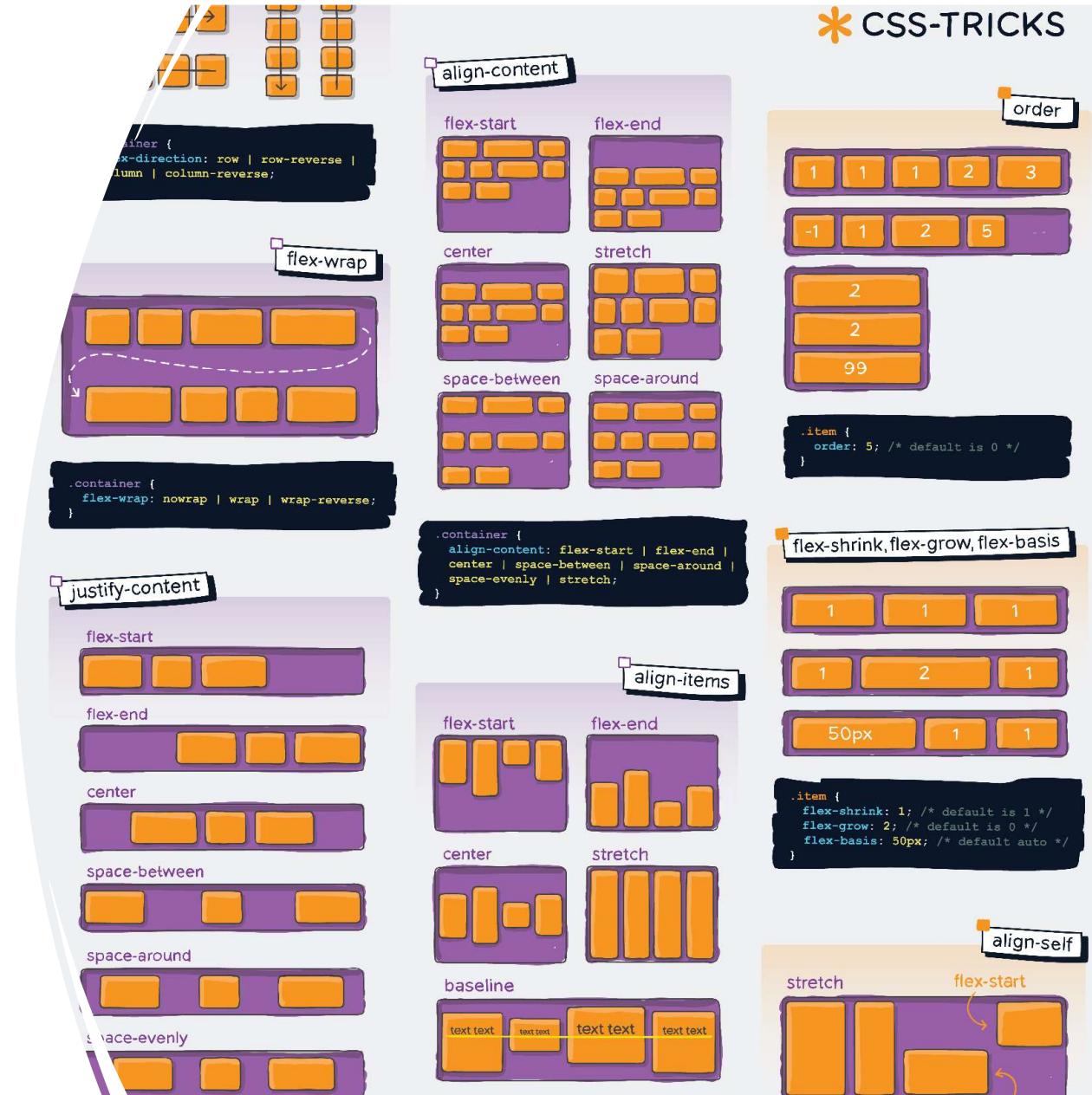
```
.container {
  background-color:#427090;
  display:flex;
  padding: 10px;
  justify-content: space-around;
  flex-direction: row;
  align-items: center;
  flex-wrap: wrap;
  height:500px;
  align-content: flex-end;
}

.A {
  padding: 5px;
  margin: 5px;
  background-color: #f14380;
  color: #FFF;
  font-size: 20px;
}
```

The Result tab shows a blue background with 12 red rectangular boxes arranged in two rows. The first row contains items 1 through 10. The second row contains items 11 and 12. Each box contains a white number corresponding to its index.

# CSS-Tricks

- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>



# FLEXBOX FROGGY

Level 1 of 24

Welcome to Flexbox Froggy, a game where you help Froggy and friends by writing CSS code! Guide this frog to the lily pad on the right by using the **justify-content** property, which aligns items horizontally and accepts the following values:

- **flex-start**: Items align to the left side of the container.
- **flex-end**: Items align to the right side of the container.
- **center**: Items align at the center of the container.
- **space-between**: Items display with equal spacing between them.
- **space-around**: Items display with equal spacing around them.

For example, **justify-content: flex-end;** will move the frog to the right.

```
1 #pond {  
2   display: flex;  
3   |  
4 }  
5
```



## Flexbox Froggy

Next

# Exercicio1



**Rio de Janeiro**

- ✓ Aéreo ida e volta
- ✓ 03 Diárias
- ✓ Café da manhã

A partir de **R\$ 368**  
TAXAS INCLUSAS  
[EM ATÉ 10X SEM JUROS](#)

[Confira](#)



**Porto Seguro**

- ✓ Aéreo ida e volta
- ✓ 04 diárias
- ✓ Café da manhã

A partir de **R\$ 637**  
TAXAS INCLUSAS  
[EM ATÉ 10X SEM JUROS](#)

[Confira](#)



**Fortaleza**

- ✓ Aéreo ida e volta
- ✓ 04 diárias
- ✓ Café da manhã

A partir de **R\$ 674**  
TAXAS INCLUSAS  
[EM ATÉ 10X SEM JUROS](#)

[Confira](#)



**Foz do Iguaçu**

- ✓ Aéreo ida e volta
- ✓ 05 Diárias
- ✓ Café da manhã

A partir de **R\$ 379**  
TAXAS INCLUSAS  
[EM ATÉ 10X SEM JUROS](#)

[Confira](#)



**Natal**

- ✓ Aéreo ida e volta
- ✓ 04 Diárias
- ✓ Café da manhã

A partir de **R\$ 774**  
TAXAS INCLUSAS  
[EM ATÉ 10X SEM JUROS](#)

[Confira](#)



**Gramado**

- ✓ Aéreo ida e volta
- ✓ 03 diárias
- ✓ Café da manhã

A partir de **R\$ 440**  
TAXAS INCLUSAS  
[EM ATÉ 10X SEM JUROS](#)

[Confira](#)



**Porto de Galinhas**

- ✓ Aéreo ida e volta
- ✓ 04 Diárias
- ✓ Café da manhã

A partir de **R\$ 775**  
TAXAS INCLUSAS  
[EM ATÉ 10X SEM JUROS](#)

[Confira](#)



**Maceió**

- ✓ Aéreo ida e volta
- ✓ 05 diárias
- ✓ Café da manhã

A partir de **R\$ 817**  
TAXAS INCLUSAS  
[EM ATÉ 10X SEM JUROS](#)

[Confira](#)

# Exercicio1



Salvador

- ✓ Áereo ida e volta
- ✓ 03 diárias
- ✓ Café da Manhã

**R\$670,00**

Taxas Inclusas  
Em até 10x sem Juros

[Comprar](#)



Fortaleza

- ✓ Áereo ida e volta
- ✓ 06 diárias
- ✓ Café da Manhã

**R\$1513,00**

Taxas Inclusas  
Em até 10x sem Juros

[Comprar](#)



Campinas

- ✓ Áereo ida e volta
- ✓ 04 diárias
- ✓ Café da Manhã

**R\$900,00**

Taxas Inclusas  
Em até 10x sem Juros

[Comprar](#)

# Exercicio1



Ajustar automaticamente para telas menores que 1200px



Salvador

- ✓ Áereo ida e volta
- ✓ 03 diárias
- ✓ Café da Manhã

R\$670,00

Taxas Inclusas  
Em até 10x sem Juros

Comprar



Fortaleza

- ✓ Áereo ida e volta
- ✓ 06 diárias
- ✓ Café da Manhã

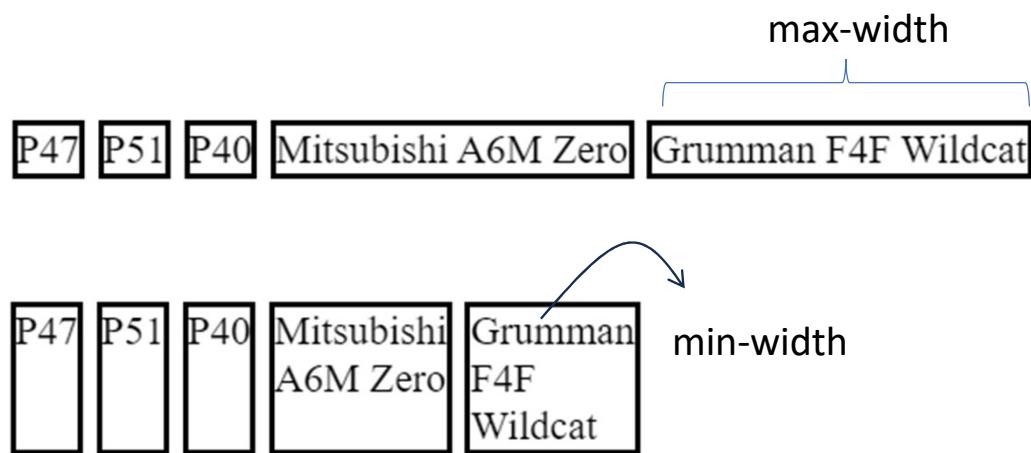
R\$1513,00

Taxas Inclusas  
Em até 10x sem Juros

Comprar



# Max-width x min-width



# Referências

- CSS Flexbox - [https://www.w3schools.com/css/css3\\_flexbox.asp](https://www.w3schools.com/css/css3_flexbox.asp)
- Flexbox - [https://developer.mozilla.org/pt-BR/docs/Learn/CSS/CSS\\_layout/Flexbox](https://developer.mozilla.org/pt-BR/docs/Learn/CSS/CSS_layout/Flexbox)
- Flexbox30 - <https://www.samanthaming.com/flexbox30/>
- Flex-Direction CSS property - <https://www.scaler.com/topics/flex-direction-css/>