



O que é o Flexbox

Por muito tempo, as únicas ferramentas disponíveis para criar layouts em CSS e posicionar elementos com boa compatibilidade entre browsers eram float e position. Porém, essas ferramentas possuem algumas limitações, especialmente no que diz respeito à responsividade.

Algumas tarefas que consideramos básicas em um layout, como centralização vertical de um elemento-filho com relação a um elemento-pai ou fazer com que elementos-filhos ocupem a mesma quantidade de espaço, ou colunas terem o mesmo tamanho independente da quantidade de conteúdo interno, eram impossíveis ou muito difíceis de serem manejadas com floats ou position, ao menos de forma prática e flexível.

A ferramenta Flexbox (de Flexible Box) foi criada para tornar essas tarefas mais simples e funcionais: os filhos de um elemento com Flexbox podem se posicionar em qualquer direção e podem ter dimensões flexíveis para se adaptar.

Elementos

O Flexbox é um módulo completo e não uma única propriedade; algumas delas devem ser declaradas no container (o elemento-pai, que chamamos de flex container), enquanto outras devem ser declaradas nos elementos-filhos (os flex itens).

Flex container é o elemento que envolve sua estrutura. Você define que um elemento é um Flex Container com a propriedade display e valores flex ou inline-flex.

O Flex Container é a tag que envolve os itens flex, ao indicar display: flex, essa tag passa a ser um Flex Container.

1 • display

Define o elemento como um flex container, tornando os seus filhos flex-itens.

- `display: flex;` // Torna o elemento um flex container automaticamente transformando todos os seus filhos diretos em flex itens.

HTML

CSS

```
<h1>display: block;</h1>
<section class="container">
  <div class="item">1</div>
  <div class="item">2</div>
  <div class="item">3</div>
  <div class="item">4</div>
</section>
```

```
<h1>display: flex;</h1>
<section class="container flex">
  <div class="item">1</div>
  <div class="item">2</div>
  <div class="item">3</div>
  <div class="item">4</div>
</section>
```

```
<h1>display: flex;</h1>
<section class="container flex">
  <div class="item">Teste 1</div>
  <div class="item">Teste 2</div>
  <div class="item">Teste 3</div>
  <div class="item">Teste 4</div>
  <div class="item">Teste 5</div>
  <div class="item">Teste 6</div>
  <div class="item">Teste 7</div>
</section>
```

```
<h1>display: flex;</h1>
<section class="container flex flex-wrap">
  <div class="item">Teste 1</div>
  <div class="item">Teste 2</div>
  <div class="item">Teste 3</div>
  <div class="item">Teste 4</div>
  <div class="item">Teste 5</div>
  <div class="item">Teste 6</div>
  <div class="item">Teste 7</div>
</section>
```

```
<h1>display: flex; // flex-item-1</h1>
```

```
/* Flex */
.flex {
  display: flex;
}

.flex-wrap {
  flex-wrap: wrap;
}

.flex-item-1 {
  flex: 1;
}

/* Flex Item */
.item {
  margin: 5px;
  background: tomato;
  text-align: center;
  font-size: 1.5em;
}

.container {
  max-width: 400px;
  margin: 0 auto;
  border: 1px solid #ccc;
}

h1 {
  text-align: center;
  margin: 20px 0 0 0;
  font-size: 1.25em;
  font-weight: normal;
}

body {
  font-family: monospace;
  color: #333;
}
```

<pre><section class="container flex"> <div class="item flex-item-1">Teste 1</div> <div class="item flex-item-1">Teste 2</div> <div class="item flex-item-1">Teste 3</div> </section></pre>	
--	--

2 • flex-direction

Define a direção dos flex itens. Por padrão ele é row (linha), por isso quando o display: flex; é adicionado, os elementos ficam em linha, um do lado do outro.

- `flex-direction: row;` // Os itens ficam em linha
- `flex-direction: row-reverse;` // Os itens ficam em linha reversa, ou seja 3, 2, 1.
- `flex-direction: column;` // Os itens ficam em uma única coluna, um embaixo do outro.
- `flex-direction: column-reverse;` // Os itens ficam em uma única coluna, um embaixo do outro, porém em ordem reversa: 3, 2 e 1.

HTML

CSS

<pre><h1>flex-direction: row;</h1> <section class="container row"> <div class="item">1</div> <div class="item">2</div> <div class="item">3</div> </section> <h1>flex-direction: row-reverse;</h1> <section class="container row-reverse"> <div class="item">1</div> <div class="item">2</div> <div class="item">3</div> </section> <h1>flex-direction: column;</h1> <section class="container column"> <div class="item">1</div> <div class="item">2</div> <div class="item">3</div> </section> <h1>flex-direction: column-reverse;</h1> <section class="container column-reverse"> <div class="item">1</div> <div class="item">2</div></pre>	<pre>.row { flex-direction: row; } .row-reverse { flex-direction: row-reverse; } .column { flex-direction: column; } .column-reverse { flex-direction: column-reverse; } /* Flex Container */ .container { max-width: 400px; margin: 0 auto; display: flex; border: 1px solid #ccc; } /* Flex Item */ .item { /* O flex: 1; é necessário para que cada item se expanda ocupando o tamanho máximo do container. */</pre>
--	---

<pre><div class="item">3</div> </section></pre>	<pre>flex: 1; margin: 5px; background: tomato; text-align: center; font-size: 1.5em; } h1 { text-align: center; margin: 20px 0 0 0; font-size: 1.25em; font-weight: normal; } body { font-family: monospace; color: #333; }</pre>
---	---

3 • flex-wrap

Define se os itens devem quebrar ou não a linha. Por padrão eles não quebram linha, isso faz com que os flex itens sejam compactados além do limite do conteúdo.

Essa é geralmente uma propriedade que é quase sempre definida como flex-wrap: wrap; Pois assim quando um dos flex itens atinge o limite do conteúdo, o último item passa para a coluna debaixo e assim por diante.

- `flex-wrap: nowrap;` // Valor padrão, não permite a quebra de linha.
- `flex-wrap: wrap;` // Quebra a linha assim que um dos flex itens não puder mais ser compactado.
- `flex-wrap: wrap-reverse;` // Quebra a linha assim que um dos flex itens não puder mais ser compactado. A quebra é na direção contrária, ou seja para a linha acima.

HTML

CSS

<pre><h1>flex-wrap: nowrap;</h1> <section class="container nowrap"> <!-- O nome grudado foi necessário pois eu preciso definir um conteúdo que não possa ser quebrado. Como uma palavra ou uma imagem. Se fosse uma frase, ele quebraria as linhas da frase antes de ultrapassar o container. --> <div class="item">TesteDoItem1</div></pre>	<pre>.nowrap { flex-wrap: nowrap; } .wrap { flex-wrap: wrap; } .wrap-reverse { flex-wrap: wrap-reverse;</pre>
--	---

<pre> <div class="item">TesteDoItem2</div> <div class="item">TesteDoItem3</div> </section> <h1>flex-wrap: wrap;</h1> <section class="container wrap"> <div class="item">TesteDoItem1</div> <div class="item">TesteDoItem2</div> <div class="item">TesteDoItem3</div> </section> <h1>flex-wrap: wrap-reverse;</h1> <section class="container wrap-reverse"> <div class="item">TesteDoItem1</div> <div class="item">TesteDoItem2</div> <div class="item">TesteDoItem3</div> </section> </pre>	<pre> } /* Flex Container */ .container { max-width: 360px; margin: 0 auto; display: flex; border: 1px solid #ccc; } /* Flex Item */ .item { /* O flex: 1; é necessário para que cada item se expanda ocupando o tamanho máximo do container. */ flex: 1; margin: 5px; background: tomato; text-align: center; font-size: 1.5em; } h1 { text-align: center; margin: 20px 0 0 0; font-size: 1.25em; font-weight: normal; } body { font-family: monospace; color: #333; } </pre>
---	--

4 • flex-flow

O flex-flow é um atalho para as propriedades flex-direction e flex-wrap. Você não verá muito o seu uso, pois geralmente quando mudamos o flex-direction para column, mantemos o padrão do flex-wrap que é nowrap.

E quando mudamos o flex-wrap para wrap, mantemos o padrão do flex-direction que é row.

- `flex-flow: row nowrap;` // Coloca o conteúdo em linha e não permite a quebra de linha.
- `flex-flow: row wrap;` // Coloca o conteúdo em linha e permite a quebra de linha.
- `flex-flow: column nowrap;` // Coloca o conteúdo em coluna e não permite a quebra de linha.

<pre> <h1>flex-flow: row nowrap;</h1> <section class="container row-nowrap"> <div class="item">TesteDoItem1</div> <div class="item">TesteDoItem2</div> <div class="item">TesteDoItem3</div> </section> <h1>flex-flow: row wrap;</h1> <section class="container row-wrap"> <div class="item">TesteDoItem1</div> <div class="item">TesteDoItem2</div> <div class="item">TesteDoItem3</div> </section> <h1>flex-flow: column-nowrap;</h1> <section class="container column-nowrap"> <div class="item">TesteDoItem1</div> <div class="item">TesteDoItem2</div> <div class="item">TesteDoItem3</div> </section> </pre>	<pre> .row-nowrap { flex-flow: row nowrap; } .column-nowrap { flex-flow: column nowrap; } .row-wrap { flex-flow: row wrap; } /* Flex Container */ .container { max-width: 360px; margin: 0 auto; display: flex; border: 1px solid #ccc; } /* Flex Item */ .item { /* O flex: 1; é necessário para que cada item se expanda ocupando o tamanho máximo do container. */ flex: 1; margin: 5px; background: tomato; text-align: center; font-size: 1.5em; } h1 { text-align: center; margin: 20px 0 0 0; font-size: 1.25em; font-weight: normal; } body { font-family: monospace; color: #333; } </pre>
---	---

5 • justify-content

Alinha os itens flex no container de acordo com a direção. A propriedade só funciona se os itens atuais não ocuparem todo o container. Isso significa que ao definir flex: 1; ou algo similar nos itens, a propriedade não terá mais função

Excelente propriedade para ser usada em casos que você deseja alinhar um item na ponta esquerda e outro na direita, como em um simples header com marca e navegação.

- `justify-content: flex-start;` // Alinha os itens ao início do container.

- `justify-content: flex-end;` // Alinha os itens ao final do container.
- `justify-content: center;` // Alinha os itens ao centro do container.
- `justify-content: space-between;` // Cria um espaçamento igual entre os elementos. Mantendo o primeiro grudado no início e o último no final.
- `justify-content: space-around;` // Cria um espaçamento entre os elementos. Os espaçamentos do meio são duas vezes maiores que o inicial e final.

HTML

CSS

```
<h1>justify-content: flex-start;</h1>
<section class="container flex-start">
  <div class="item">1</div>
  <div class="item">Teste 2</div>
  <div class="item">3</div>
  <div class="item">4</div>
</section>
```

```
<h1>justify-content: flex-end;</h1>
<section class="container flex-end">
  <div class="item">1</div>
  <div class="item">Teste 2</div>
  <div class="item">3</div>
  <div class="item">4</div>
</section>
```

```
<h1>justify-content: center;</h1>
<section class="container center">
  <div class="item">1</div>
  <div class="item">Teste 2</div>
  <div class="item">3</div>
  <div class="item">4</div>
</section>
```

```
<h1>justify-content: space-between;</h1>
<section class="container space-between">
  <div class="item">1</div>
  <div class="item">Teste 2</div>
  <div class="item">3</div>
  <div class="item">4</div>
</section>
```

```
<h1>justify-content: space-around;</h1>
<section class="container space-around">
  <div class="item">1</div>
  <div class="item">Teste 2</div>
  <div class="item">3</div>
  <div class="item">4</div>
</section>
```

```
.flex-start {
  justify-content: flex-start;
}

.flex-end {
  justify-content: flex-end;
}

.center {
  justify-content: center;
}

.space-between {
  justify-content: space-between;
}

.space-around {
  justify-content: space-around;
}

/* Flex Container */
.container {
  max-width: 420px;
  margin: 0 auto;
  display: flex;
  border: 1px solid #ccc;
}

.column {
  min-height: 200px;
  flex-direction: column;
}

/* Flex Item */
.item {
  margin: 5px;
  padding: 0 10px;
  background: tomato;
  text-align: center;
}
```

<pre> <h1>justify-content: space-around; // itens com flex: 1;</h1> <section class="container space-around"> <div class="item flex">1</div> <div class="item flex">Teste 2</div> <div class="item flex">3</div> <div class="item flex">4</div> </section> <h1>justify-content: flex-start; // column</h1> <section class="container flex-start column"> <div class="item">1</div> <div class="item">Teste 2</div> <div class="item">3</div> <div class="item">4</div> </section> <h1>justify-content: flex-end; // column</h1> <section class="container flex-end column"> <div class="item">1</div> <div class="item">Teste 2</div> <div class="item">3</div> <div class="item">4</div> </section> <h1>justify-content: center; // column</h1> <section class="container center column"> <div class="item">1</div> <div class="item">Teste 2</div> <div class="item">3</div> <div class="item">4</div> </section> <h1>justify-content: space-between; // column</h1> <section class="container space-between column"> <div class="item">1</div> <div class="item">Teste 2</div> <div class="item">3</div> <div class="item">4</div> </section> <h1>justify-content: space-around; // column</h1> <section class="container space-around column"> <div class="item">1</div> <div class="item">Teste 2</div> <div class="item">3</div> <div class="item">4</div> </section> <h1>justify-content: space-around; // column // itens com flex: 1;</h1> <section class="container space-around column"> <div class="item flex">1</div> <div class="item flex">Teste 2</div> <div class="item flex">3</div> <div class="item flex">4</div> </pre>	<pre> font-size: 1.5em; } /* Indica que o conteúdo deve se expandir */ .flex { flex: 1; } h1 { text-align: center; margin: 20px 0 0 0; font-size: 1.25em; font-weight: normal; } body { font-family: monospace; color: #333; } </pre>
---	--

6 • align-items

O align-items alinha os flex itens de acordo com o eixo do container. O alinhamento é diferente para quando os itens estão em colunas ou linhas.

Essa propriedade permite o tão sonhado alinhamento central no eixo vertical, algo que antes só era possível com diferentes hacks.

- `align-items: stretch;`// Valor padrão, ele que faz com que os flex itens cresçam igualmente.
- `align-items: flex-start;`// Alinha os itens ao início.
- `align-items: flex-end;`// Alinha os itens ao final.
- `align-items: center;`// Alinha os itens ao centro.
- `align-items: baseline;`// Alinha os itens de acordo com a linha base da tipografia.

HTML

CSS

```
<h1>align-items: stretch;</h1>
<section class="container stretch">
  <div class="item">1</div>
  <div class="item">Teste 2 do Item</div>
  <div class="item">Teste 3 do Item Agora
Mesmo</div>
  <div class="item">Teste 4</div>
</section>
```

```
<h1>align-items: flex-start;</h1>
<section class="container flex-start">
  <div class="item">1</div>
  <div class="item">Teste 2 do Item</div>
  <div class="item">Teste 3 do Item Agora
Mesmo</div>
  <div class="item">Teste 4</div>
</section>
```

```
<h1>align-items: flex-end;</h1>
<section class="container flex-end">
  <div class="item">1</div>
  <div class="item">Teste 2 do Item</div>
  <div class="item">Teste 3 do Item Agora
Mesmo</div>
  <div class="item">Teste 4</div>
</section>
```

```
<h1>align-items: center;</h1>
```

```
.stretch {
  align-items: stretch;
}

.flex-start {
  align-items: flex-start;
}

.flex-end {
  align-items: flex-end;
}

.center {
  align-items: center;
}

.baseline {
  align-items: baseline;
}

/* Flex Container */
.container {
  max-width: 420px;
  margin: 0 auto;
  display: flex;
  border: 1px solid #ccc;
}
```

```

<section class="container center">
  <div class="item">1</div>
  <div class="item">Teste 2 do Item</div>
  <div class="item">Teste 3 do Item Agora
Mesmo</div>
  <div class="item">Teste 4</div>
</section>

```

```

<h1>align-items: baseline;</h1>
<section class="container baseline">
  <div class="item teste">1</div>
  <div class="item">Teste 2 do Item</div>
  <div class="item">Teste 3 do Item Agora
Mesmo</div>
  <div class="item">Teste 4</div>
</section>

```

```

<h1>align-items: flex-stretch; // column</h1>
<section class="container stretch column">
  <div class="item">1</div>
  <div class="item">Teste 2 do Item</div>
  <div class="item">Teste 3 do Item Agora
Mesmo</div>
  <div class="item">Teste 4</div>
</section>

```

```

<h1>align-items: flex-start; // column</h1>
<section class="container flex-start column">
  <div class="item">1</div>
  <div class="item">Teste 2 do Item</div>
  <div class="item">Teste 3 do Item Agora
Mesmo</div>
  <div class="item">Teste 4</div>
</section>

```

```

<h1>align-items: flex-end; // column</h1>
<section class="container flex-end column">
  <div class="item">1</div>
  <div class="item">Teste 2 do Item</div>
  <div class="item">Teste 3 do Item Agora
Mesmo</div>
  <div class="item">Teste 4</div>
</section>

```

```

<h1>align-items: center; // column</h1>
<section class="container center column">
  <div class="item">1</div>
  <div class="item">Teste 2 do Item</div>
  <div class="item">Teste 3 do Item Agora
Mesmo</div>
  <div class="item">Teste 4</div>
</section>

```

```

<h1>align-items: baseline; // column</h1>
<section class="container baseline column">
  <div class="item teste">1</div>
  <div class="item">Teste 2 do Item</div>

```

```

.column {
  flex-direction: column;
}

```

```

/* Flex Item */
.item {
  flex: 1;
  margin: 5px;
  padding: 0 10px;
  background: tomato;
  text-align: center;
  font-size: 1.5em;
}

```

```

.item:first-child {
  padding-top: 20px;
}

```

```

.alinhamento-central {
  height: 400px;
  justify-content: center;
  align-items: center;
}

```

```

.alinhamento-central .item {
  flex: 0;
  padding: 10px;
}

```

```

h1 {
  text-align: center;
  margin: 20px 0 0 0;
  font-size: 1.25em;
  font-weight: normal;
}

```

```

body {
  font-family: monospace;
  color: #333;
}

```

<pre> <div class="item">Teste 3 do Item Agora Mesmo</div> <div class="item">Teste 4</div> </section> <h1>align-items: center; e justify-content: center;</h1> <section class="container alinhamento-central"> <div class="item">Esse item está verticalmente alinhado ao centro.</div> </section> </pre>	
---	--

7 • align-content

Alinha as linhas do container em relação ao eixo vertical. A propriedade só funciona se existir mais de uma linha de flex-itens. Para isso o flex-wrap precisa ser wrap.

Além disso o efeito dela apenas será visualizado caso o container seja maior que a soma das linhas dos itens. Isso significa que se você não definir height para o container, a propriedade não influencia no layout.

- `align-content: stretch;` // Valor padrão, ele que faz com que os flex itens cresçam igualmente na vertical.
- `align-content: flex-start;` // Alinha todas as linhas de itens ao início.
- `align-content: flex-end;` // Alinha todas as linhas de itens ao final.
- `align-content: center;` // Alinha todas as linhas de itens ao centro.
- `align-content: space-between;` // Cria um espaçamento igual entre as linhas. Mantendo a primeira grudada no topo e a última no bottom.
- `align-content: space-around;` // Cria um espaçamento entre as linhas. Os espaçamentos do meio são duas vezes maiores que o top e bottom.

HTML

CSS

<pre> <h1>align-content: stretch;</h1> <section class="container stretch"> <div class="item">TestandoPalavra1</div> <div class="item">Teste 2 do Item</div> <div class="item">Teste 3 do Item Agora Mesmo</div> <div class="item">Teste 4</div> <div class="item">Teste 5</div> <div class="item">Teste 6</div> <div class="item">Teste 7</div> </pre>	<pre> .stretch { align-content: stretch; } .flex-start { align-content: flex-start; } .flex-end { align-content: flex-end; } </pre>
--	---

```

    <div class="item">Teste 8</div>
</section>

<h1>align-content: flex-start;</h1>
<section class="container flex-start">
    <div class="item">TestandoPalavra1</div>
    <div class="item">Teste 2 do Item</div>
    <div class="item">Teste 3 do Item Agora
Mesmo</div>
    <div class="item">Teste 4</div>
    <div class="item">Teste 5</div>
    <div class="item">Teste 6</div>
    <div class="item">Teste 7</div>
    <div class="item">Teste 8</div>
</section>

<h1>align-content: flex-end;</h1>
<section class="container flex-end">
    <div class="item">TestandoPalavra1</div>
    <div class="item">Teste 2 do Item</div>
    <div class="item">Teste 3 do Item Agora
Mesmo</div>
    <div class="item">Teste 4</div>
    <div class="item">Teste 5</div>
    <div class="item">Teste 6</div>
    <div class="item">Teste 7</div>
    <div class="item">Teste 8</div>
</section>

<h1>align-content: center;</h1>
<section class="container center">
    <div class="item">TestandoPalavra1</div>
    <div class="item">Teste 2 do Item</div>
    <div class="item">Teste 3 do Item Agora
Mesmo</div>
    <div class="item">Teste 4</div>
    <div class="item">Teste 5</div>
    <div class="item">Teste 6</div>
    <div class="item">Teste 7</div>
    <div class="item">Teste 8</div>
</section>

<h1>align-content: space-between;</h1>
<section class="container space-between">
    <div class="item">TestandoPalavra1</div>
    <div class="item">Teste 2 do Item</div>
    <div class="item">Teste 3 do Item Agora
Mesmo</div>
    <div class="item">Teste 4</div>
    <div class="item">Teste 5</div>
    <div class="item">Teste 6</div>
    <div class="item">Teste 7</div>
    <div class="item">Teste 8</div>
</section>

<h1>align-content: space-around;</h1>
<section class="container space-around">

```

```

}

.center {
    align-content: center;
}

.space-between {
    align-content: space-between;
}

.space-around {
    align-content: space-around;
}

/* Flex Container */
.container {
    max-width: 420px;
    height: 300px;
    margin: 0 auto;
    display: flex;
    border: 1px solid #ccc;
    flex-wrap: wrap;
}

/* Flex Item */
.item {
    flex: 1;
    margin: 5px;
    padding: 0 10px;
    background: tomato;
    text-align: center;
    font-size: 1.5em;
}

h1 {
    text-align: center;
    margin: 20px 0 0 0;
    font-size: 1.25em;
    font-weight: normal;
}

body {
    font-family: monospace;
    color: #333;
}

```

<pre> <div class="item">TestandoPalavra1</div> <div class="item">Teste 2 do Item</div> <div class="item">Teste 3 do Item Agora Mesmo</div> <div class="item">Teste 4</div> <div class="item">Teste 5</div> <div class="item">Teste 6</div> <div class="item">Teste 7</div> <div class="item">Teste 8</div> </section> </pre>	
--	--

Flex Item

Os Flex Itens são os filhos diretos do Flex Container, lembrado que uma tag se torna um flex container a partir do momento que você definir display: flex.

É possível que um Flex Item seja também um Flex Container, basta definir display: flex nele. Assim os filhos desse item também serão flex itens.

1 • flex-grow

Define a habilidade de um flex item crescer. Por padrão o valor é zero, assim os flex itens ocupam um tamanho máximo relacionado o conteúdo interno deles ou ao width definido.

Ao definir 1 para todos os Flex Itens, eles tentarão ter a mesma largura e vão ocupar 100% do container. Digo tentarão pois caso um elemento possua um conteúdo muito largo, ele irá respeitar o mesmo.

Se você tiver uma linha com quatro itens, onde três são flex-grow: 1 e um flex-grow: 2, o flex-grow: 2 tentará ocupar 2 vezes mais espaço extra do que os outros elementos.

OBS: justify-content não funciona em itens com flex-grow definido.

- **flex-grow: número;** // Basta definir um número
- **flex-grow: 0;** // Obedece o width do elemento ou o flex-basis.

HTML

CSS

<pre> <h1>flex-grow: 0;</h1> <section class="container"> <div class="item grow0">1</div> <div class="item grow0">2</div> <div class="item grow0">3</div> </section> <h1>flex-grow: 1;</h1> <section class="container"> <div class="item grow1">1</div> <div class="item grow1">2</div> </pre>	<pre> /* Os itens não se expandem para ocupar o container. */ .grow0 { flex-grow: 0; } /* Os itens se expandem igualmente para ocupar o container. Utilizando apenas o flex-grow, o crescimento é distribuído igualmente antes e depois do conteúdo de cada elemento. (pois por padrão o flex-basis é auto) */ </pre>
--	--

<pre> <div class="item grow1">3</div> </section> <h1>flex-grow: 2;</h1> <section class="container column"> <div class="item grow2">1</div> <div class="item grow2">2</div> <div class="item grow2">3</div> </section> <h1>flex-grow: 1; // conteúdo largo</h1> <section class="container column-reverse"> <div class="item grow1">1</div> <div class="item grow1">TestandoTestando2</div> <div class="item grow1">3</div> </section> <h1>flex-grow: (misturado);</h1> <section class="container column-reverse"> <div class="item grow0">1</div> <div class="item grow1">2</div> <div class="item grow2">3</div> </section> </pre>	<pre> .grow1 { flex-grow: 1; } /* Os itens se expandem duas vezes mais que aqueles com flex-grow: 1; */ .grow2 { flex-grow: 2; } .item { margin: 5px; background: tomato; text-align: center; font-size: 1.5em; } /* Flex Container */ .container { max-width: 400px; margin: 0 auto; display: flex; border: 1px solid #ccc; } /* Flex Item */ h1 { text-align: center; margin: 20px 0 0 0; font-size: 1.25em; font-weight: normal; } body { font-family: monospace; color: #333; } </pre>
---	--

2 • flex-basis

Indica o tamanho inicial do flex item antes da distribuição do espaço restante.

Quando definimos o flex-grow: 1; e possuímos auto no basis, o valor restante para ocupar o container é distribuído ao redor do conteúdo do flex-item.

- **flex-basis: auto;** // Esse é o padrão, ele faz com que a largura da base seja igual a do item. Se o item não tiver tamanho especificado, o tamanho será de acordo com o conteúdo.
- **flex-basis: unidade;** // Pode ser em %, em, px e etc.
- **flex-basis: 0;** // Se o grow for igual ou maior que 1, ele irá tentar manter todos os elementos com a mesma

largura, independente do conteúdo (por isso 0 é o valor mais comum do flex-basis). Caso contrário o item terá a largura do seu conteúdo.

HTML

CSS

```
<h1>flex-basis: auto;</h1>
<section class="container">
  <div class="item basis-auto">Palavra 1</div>
  <div class="item basis-auto">2</div>
  <div class="item basis-auto-grow-2">3</div>
  <div class="item basis-auto-width">4</div>
  <div class="item basis-auto-grow-0">5</div>
</section>

<section class="container overlay">
  <div class="item basis-auto">Palavra 1</div>
  <div class="item basis-auto">2</div>
  <div class="item basis-auto-grow-2">3</div>
  <div class="item basis-auto-width">4</div>
  <div class="item basis-auto-grow-0">5</div>
</section>

<h1>flex-basis: 0;</h1>
<section class="container">
  <div class="item basis-0">Test de Cont</div>
  <div class="item basis-0">2</div>
  <div class="item basis-0-grow-2">Palavra</div>
  <div class="item basis-0-width">4</div>
  <div class="item basis-0-grow-0">5</div>
</section>

<section class="container overlay">
  <div class="item basis-0">Test de Cont</div>
  <div class="item basis-0">2</div>
  <div class="item basis-0-grow-2">Palavra</div>
  <div class="item basis-0-width">4</div>
  <div class="item basis-0-grow-0">5</div>
</section>

<h1>flex-basis: número;</h1>
<section class="container basis-numero-container">
  <div class="item basis-110">1</div>
  <div class="item basis-110">12</div>
  <div class="item
basis-110">12345678910</div>
  <div class="item basis-300">1234</div>
  <div class="item basis-110-porcento">1</div>
  <div class="item basis-110-porcento">12</div>
  <div class="item
basis-110-porcento">123</div>
  <div class="item
basis-110-porcento">1234</div>
</section>
```

```
/* FLEX BASIS AUTO */

.basis-auto {
  flex-grow: 1;
  flex-basis: auto;
}

/* O espaço distribuído é 2 vezes maior que o
flex-grow: 1; */
.basis-auto-grow-2 {
  flex-grow: 2;
  flex-basis: auto;
}

/* É o mesmo espaço distribuído que os demais
itens com flex-grow: 1; Porém esse espaço inicia
após os 60px iniciais */
.basis-auto-width {
  flex-grow: 1;
  flex-basis: auto;
  width: 60px;
}

/* Com o flex-grow: 0; nada é distribuído */
.basis-auto-grow-0 {
  flex-basis: auto;
}

.container.overlay {
  margin-top: 10px;
}

.overlay .basis-auto::before, .overlay
.basis-auto-grow-2::before, .overlay
.basis-auto-width::before {
  content: "";
  display: block;
  width: 16px;
  height: 36px;
  background: blue;
  position: absolute;
  opacity: .5;
}

.overlay .basis-auto-grow-2::before {
  border-right: 16px solid blueviolet;
}

.overlay .basis-auto-width::before {
```

```

<h1>flex-basis: número; // Diferente Conteúdo</h1>
<section class="container basis-numero-container">
  <div class="item basis-110">Conteúdo
Grande</div>
  <div class="item
basis-110">PalavraMaior</div>
  <div class="item basis-110">Oi</div>
</section>

```

```

<h1>flex-basis: número; // grow 1</h1>
<section class="container basis-numero-container">
  <div class="item basis-110-grow-1">1</div>
  <div class="item basis-110-grow-1">12</div>
  <div class="item basis-110-grow-1">1234</div>
  <div class="item
basis-110-grow-1">12345</div>
  <div class="item basis-600-grow-1">5</div>
</section>

```

```

border-right: 30px solid;
}

/* FLEX BASIS AUTO FIM */

/* FLEX BASIS 0 */

/* Com o flex-basis 0, itens com flex-grow 1
tentam ficar com o mesmo tamanho total de
width. É por isso que o mais comum é utilizar
flex-basis 0 */
.basis-0 {
  flex-grow: 1;
  flex-basis: 0;
}

/* O tamanho total é de duas vezes dos itens com
flex: 1; */
.basis-0-grow-2 {
  flex-grow: 2;
  flex-basis: 0;
}

/* A definição do width é ignorada. Porém ele
ainda respeita a definição de min-width. */
.basis-0-width {
  flex-grow: 1;
  flex-basis: 0;
  width: 120px;
  min-width: 80px;
}

/* Flex basis 0 com flex-grow 0, não influenciar
em nada no item. Assim o mesmo fica do
tamanho do seu conteúdo. */
.basis-0-grow-0 {
  flex-basis: 0;
}

.overlay .basis-0::before, .overlay
.basis-0-grow-2::before, .overlay
.basis-0-width::before {
  content: "";
  display: block;
  width: 65px;
  height: 36px;
  background: blue;
  position: absolute;
  opacity: .5;
}

.overlay .basis-0-grow-2::before {
  border-right: 65px solid blueviolet;
}

/* FLEX BASIS 0 FIM */

```



```
.basis-numero-container {
    flex-wrap: wrap;
}

.basis-110 {
    flex-basis: 110px;
}

.basis-300 {
    flex-basis: 300px;
}

.basis-110-porcento {
    flex-basis: calc(25% - 10px);
}

.basis-110-grow-1 {
    flex-grow: 1;
    flex-basis: 110px;
}

/* O Flex-basis não ultrapassa o tamanho do
container. Pois por padrão o flex-shrink é 1, ou
seja, ele permite que o item se ajuste para caber
no container. */
.basis-600-grow-1 {
    flex-grow: 1;
    flex-basis: 600px;
}

/* ESTILOS GERAIS */

.item {
    margin: 5px;
    background: tomato;
    text-align: center;
    font-size: 1.5em;
    position: relative;
}

/* Flex Container */
.container {
    max-width: 400px;
    margin: 0 auto;
    display: flex;
    border: 1px solid #ccc;
}

/* Flex Item */
img {
    display: block;
    max-width: 100%;
    margin: 10px auto;
}

h1 {
    text-align: center;
```

	<pre> margin: 20px 0 0 0; font-size: 1.25em; font-weight: normal; } body { font-family: monospace; color: #333; } </pre>
--	---

3 • flex-shrink

Define a capacidade de redução de tamanho do item.

- `flex-shrink: 1;` // Valor padrão, permite que os itens tenham os seus tamanhos (seja esse tamanho definido a partir de width ou flex-basis) reduzidos para caber no container.
- `flex-shrink: 0;` // Não permite a diminuição dos itens, assim um item com `flex-basis: 300px;` nunca diminuirá menos do que 300px, mesmo que o conteúdo não ocupe todo esse espaço.
- `flex-shrink: número;` // Um item com `shrink: 3` diminuirá 3 vezes mais que um item com 1.

HTML

CSS

<pre> <h1>flex-shrink: 1;</h1> <section class="container"> <div class="item shrink-1">1</div> <div class="item shrink-1">2</div> <div class="item shrink-1">3</div> <div class="item shrink-1">4</div> </section> <h1>flex-shrink: 1; // Conteúdo diferente</h1> <section class="container"> <div class="item shrink-1">1</div> <div class="item shrink-1">Teste 2</div> <div class="item shrink-1">Oi 3</div> <div class="item shrink-1">4</div> </section> <h1>flex-shrink: misturado; // Conteúdo diferente</h1> <section class="container"> <div class="item shrink-1">1</div> <div class="item shrink-1">Teste 2</div> <div class="item shrink-3">Oi 3</div> <div class="item shrink-3">4</div> </section> <h1>flex-shrink: 1; // Conteúdo maior que o container</h1> </pre>	<pre> .shrink-1 { flex-grow: 1; flex-shrink: 1; } .shrink-3 { flex-grow: 1; flex-shrink: 3; } .shrink-1-basis { flex-grow: 1; flex-shrink: 1; flex-basis: 150px; } .shrink-0-basis { flex-grow: 1; flex-shrink: 0; flex-basis: 150px; } .shrink-3-basis { flex-grow: 1; flex-shrink: 3; } </pre>
--	--

```

<section class="container">
  <div class="item shrink-1">TesteTesteTesteTeste</div>
  <div class="item shrink-1">TesteTesteTeste 2</div>
  <div class="item shrink-1">Oi 3</div>
  <div class="item shrink-1">4</div>
</section>

```

<h1>flex-shrink: misturado; // Conteúdo maior que o container</h1>

```

<section class="container">
  <div class="item shrink-1">TesteTesteTesteTeste</div>
  <div class="item shrink-1">TesteTesteTeste 2</div>
  <div class="item shrink-3">Oi 3</div>
  <div class="item shrink-3">4</div>
</section>

```

<h1>flex-shrink: 1; / flex-basis: valor;</h1>

```

<section class="container">
  <div class="item shrink-1-basis">TesteTesteTesteTeste</div>
  <div class="item shrink-1-basis">TesteTesteTeste 2</div>
  <div class="item shrink-1-basis">Oi 3</div>
  <div class="item shrink-1-basis">4</div>
</section>

```

<h1>flex-shrink: 0; / flex-basis: valor;</h1>

```

<section class="container">
  <div class="item shrink-0-basis">TesteTesteTesteTeste</div>
  <div class="item shrink-0-basis">TesteTesteTeste 2</div>
  <div class="item shrink-0-basis">Oi 3</div>
  <div class="item shrink-0-basis">4</div>
</section>

```

<h1>flex-shrink: 0; / flex-basis: valor; / wrap</h1>

```

<section class="container flex-wrap">
  <div class="item shrink-0-basis">TesteTesteTesteTeste</div>
  <div class="item shrink-0-basis">TesteTesteTeste 2</div>
  <div class="item shrink-0-basis">Oi 3</div>
  <div class="item shrink-0-basis">4</div>
</section>

```

<h1>flex-shrink: misturado; / flex-basis: valor;</h1>

```

<section class="container">
  <div class="item shrink-1-basis">1</div>
  <div class="item shrink-1-basis">Teste 2</div>
  <div class="item shrink-3-basis">3</div>
  <div class="item shrink-3-basis">4</div>
</section>

```

<h1>flex-shrink: misturado; / flex-basis: valor; / wrap</h1>

```

<section class="container flex-wrap">
  <div class="item shrink-1-basis">1</div>

```

```

    flex-basis: 150px;
  }

.flex-wrap {
  flex-wrap: wrap;
}

.item {
  margin: 5px;
  background: tomato;
  text-align: center;
  font-size: 1.5em;
}

/* Flex Container */
.container {
  max-width: 400px;
  margin: 0 auto;
  display: flex;
  border: 1px solid #ccc;
}

/* Flex Item */

h1 {
  text-align: center;
  margin: 20px 0 0 0;
  font-size: 1.25em;
  font-weight: normal;
}

body {
  font-family: monospace;
  color: #333;
}

```

<pre> <div class="item shrink-1-basis">Teste 2</div> <div class="item shrink-3-basis">3</div> <div class="item shrink-3-basis">4</div> </section> </pre>	
--	--

4 • flex

Atalho para as propriedades flex-grow, flex-shrink e flex-basis. Geralmente você verá a propriedade flex nos flex itens ao invés de cada um dos valores separados.

Para melhor consistência entre os browsers, é recomendado utilizar a propriedade flex ao invés de cada propriedade separada.

No exemplo é possível ver as mesmas configurações do exemplo do flex-basis porém agora utilizando apenas a propriedade flex.

- `flex: 1;` // Define flex-grow: 1; flex-shrink: 1; e flex-basis: 0; (em alguns browsers define como 0%, pois estes ignoram valores sem unidades, porém a função de 0 e 0% é a mesma.)
- `flex: 0 1 auto;` // Esse é o padrão, se você não definir nenhum valor de flex ou para as outras propriedades separadas, o normal será flex-grow: 0, flex-shrink: 1 e flex-basis: auto.
- `flex: 2;` // Define exatamente da mesma forma que o flex: 1; porém neste caso o flex-grow será de 2, o flex-shrink continuará 1 e o flex-basis 0.
- `flex: 3 2 300px;` // flex-grow: 3, flex-shrink: 2 e flex-basis: 300px;

HTML

CSS

```

<h1>flex: n 1 auto;</h1>
<section class="container">
  <div class="item basis-auto">Palavra 1</div>
  <div class="item basis-auto">2</div>
  <div class="item basis-auto-grow-2">3</div>
  <div class="item basis-auto-width">4</div>
  <div class="item basis-auto-grow-0">5</div>
</section>

<h1>flex: 1;</h1>
<section class="container">
  <div class="item basis-0">Test de Cont</div>
  <div class="item basis-0">2</div>
  <div class="item basis-0-grow-2">Palavra</div>
  <div class="item basis-0-width">4</div>

```

```

/* FLEX BASIS AUTO */

.basis-auto {
  flex: 1 1 auto;
}

.basis-auto-grow-2 {
  flex: 2 1 auto;
}

.basis-auto-width {
  flex: 1 1 auto;
  width: 60px;
}

```

```

    <div class="item basis-0-grow-0">5</div>
</section>

<h1>flex: flex: 0 1 n;</h1>
<section class="container basis-numero-container">
    <div class="item basis-110">1</div>
    <div class="item basis-110">12</div>
    <div class="item basis-110">12345678910</div>
    <div class="item basis-300">1234</div>
    <div class="item basis-110-porcento">1</div>
    <div class="item basis-110-porcento">12</div>
    <div class="item basis-110-porcento">123</div>
    <div class="item basis-110-porcento">1234</div>
</section>

<h1>flex: 0 1 n; // Diferente Conteúdo</h1>
<section class="container basis-numero-container">
    <div class="item basis-110">Conteúdo Grande</div>
    <div class="item basis-110">PalavraMaior</div>
    <div class="item basis-110">Oi</div>
</section>

<h1>flex: 1 1 n; //</h1>
<section class="container basis-numero-container">
    <div class="item basis-110-grow-1">1</div>
    <div class="item basis-110-grow-1">12</div>
    <div class="item basis-110-grow-1">1234</div>
    <div class="item basis-110-grow-1">12345</div>
    <div class="item basis-600-grow-1">5</div>
</section>

```

```

.basis-auto-grow-0 {
    flex: 0 1 auto;
}

/* FLEX BASIS AUTO FIM */

/* FLEX BASIS 0 */

.basis-0 {
    flex: 1;
}

/* O tamanho total é de duas vezes dos
itens com flex: 1; */
.basis-0-grow-2 {
    flex: 2;
}

.basis-0-width {
    flex: 1;
    width: 120px;
    min-width: 80px;
}

.basis-0-grow-0 {
    flex: 0 1 0;
}

/* FLEX BASIS 0 FIM */

.basis-numero-container {
    flex-wrap: wrap;
}

.basis-110 {
    flex: 0 1 110px;
}

.basis-300 {
    flex: 0 1 300px;
}

.basis-110-porcento {
    flex: 0 1 calc(25% - 10px);
}

.basis-110-grow-1 {
    flex: 1 1 110px;
}

.basis-600-grow-1 {
    flex: 1 1 600px;
}

/* ESTILOS GERAIS */

.item {

```

	<pre> margin: 5px; background: tomato; text-align: center; font-size: 1.5em; position: relative; } /* Flex Container */ .container { max-width: 400px; margin: 0 auto; display: flex; border: 1px solid #ccc; } /* Flex Item */ img { display: block; max-width: 100%; margin: 10px auto; } h1 { text-align: center; margin: 20px 0 0 0; font-size: 1.25em; font-weight: normal; } body { font-family: monospace; color: #333; } </pre>
--	---

5 • order

Modifica a ordem dos flex itens. Sempre do menor para o maior, assim order: 1, aparece na frente de order: 5.

- `order: número;` // Número para modificar a ordem padrão. Pode ser negativo.
- `order: 0;` // 0 é o valor padrão e isso significa que a ordem dos itens será a ordem apresentada no HTML. Se você quiser colocar um item do meio da lista no início da mesma, sem modificar os demais, o ideal é utilizar um valor negativo para este item, já que todos os outros são 0.

HTML

CSS

<pre> <h1>order: misturados;</h1> <section class="container"> </pre>	<pre> .order1 { order: 1; } </pre>
--	--

```

<div class="item order2">1</div>
<div class="item order4">2</div>
<div class="item order1">3</div>
<div class="item order3">4</div>
</section>

<h1>order: negativo; / flex-grow</h1>
<section class="container">
  <div class="item order2 flex-grow">1</div>
  <div class="item order-1 flex-grow">2</div>
  <div class="item order1 flex-grow">3</div>
  <div class="item order3 flex-grow">4</div>
</section>

<h1>order: valores iguais; / grow / column</h1>
<section class="container column">
  <div class="item order2 flex-grow">1</div>
  <div class="item order2 flex-grow">2</div>
  <div class="item order1 flex-grow">3</div>
  <div class="item order1 flex-grow">4</div>
</section>

<h1>order: valores iguais; / grow / column</h1>
<section class="container column">
  <div class="item flex-grow">1</div>
  <div class="item order2 flex-grow">2</div>
  <div class="item flex-grow">3</div>
  <div class="item order-1 flex-grow">4</div>
</section>

```

```

}

.order2 {
  order: 2;
}

.order3 {
  order: 3;
}

.order4 {
  order: 4;
}

.order-1 {
  order: -1;
}

.flex-grow {
  flex-grow: 1;
}

.column {
  flex-direction: column;
}

.item {
  margin: 5px;
  padding: 0 5px;
  background: tomato;
  text-align: center;
  font-size: 1.5em;
}

/* Flex Container */
.container {
  max-width: 400px;
  margin: 0 auto;
  display: flex;
  border: 1px solid #ccc;
}

/* Flex Item */

h1 {
  text-align: center;
  margin: 20px 0 0 0;
  font-size: 1.25em;
  font-weight: normal;
}

body {
  font-family: monospace;
  color: #333;
}

```

6 • align-self

O align-self serve para definirmos o alinhamento específico de um único flex item dentro do nosso container. Caso um valor seja atribuído, ele passara por cima do que for atribuído no align-items do container.

Vale lembrar que o alinhamento acontece tanto em linha quanto em colunas. Por exemplo o flex-start quando os itens estão em linhas, alinha o item ao topo da sua linha. Quando em colunas, alinha o item ao início (esquerda) da coluna.

- `align-self: auto;` // Valor inicial padrão. Vai respeitar o que for definido pelo align-items no flex-container.
- `align-self: flex-start;` // Alinha o item ao início.
- `align-self: flex-end;` // Alinha o item ao final.
- `align-self: center;` // Alinha o item ao centro.
- `align-self: baseline;` // Alinha o item a linha de base.
- `align-self: stretch;` // Estica o item.

HTML

CSS

```
<h1>align-self: multiplos;</h1>
<section class="container">
  <div class="item flex-end">1</div>
  <div class="item flex-start">Teste 2 do Item</div>
  <div class="item">Teste 3 do Item Agora Mesmo Esse
Mesmo.</div>
  <div class="item center">Teste 4</div>
</section>
```

```
<h1>align-self: multiplos;</h1>
<section class="container">
  <div class="item baseline">1</div>
  <div class="item baseline">Teste 2 do Item</div>
  <div class="item">Teste 3 do Item Agora Mesmo Esse
Mesmo.</div>
  <div class="item center">Teste 4</div>
</section>
```

```
<h1>align-self: stretch; // align-items: flex-end;</h1>
<section class="container align-items">
  <div class="item">1</div>
  <div class="item stretch">Teste 2 do Item</div>
  <div class="item">Teste 3 do Item Agora Mesmo Esse
Mesmo.</div>
  <div class="item">Teste 4</div>
</section>
```

```
<h1>align-self: multiplos;</h1>
```

```
.flex-start {
  align-self: flex-start;
}

.flex-end {
  align-self: flex-end;
}

.center {
  align-self: center;
}

.baseline {
  align-self: baseline;
}

.stretch {
  align-self: stretch;
}

.align-items {
  align-items: flex-end;
}

.item {
  flex: 1;
  margin: 5px;
  padding: 0 5px;
```


<pre> <section class="container column"> <div class="item flex-end">1</div> <div class="item flex-start">Teste 2 do Item</div> <div class="item">Teste 3 do Item Agora Mesmo Esse Mesmo.</div> <div class="item center">Teste 4</div> </section> <h1>align-self: multiplos;</h1> <section class="container column"> <div class="item baseline">1</div> <div class="item baseline">Teste 2 do Item</div> <div class="item">Teste 3 do Item Agora Mesmo Esse Mesmo.</div> <div class="item center">Teste 4</div> </section> <h1>align-self: stretch; // align-items: flex-end;</h1> <section class="container column align-items"> <div class="item">1</div> <div class="item stretch">Teste 2 do Item</div> <div class="item">Teste 3 do Item Agora Mesmo Esse Mesmo.</div> <div class="item">Teste 4</div> </section> </pre>	<pre> background: tomato; text-align: center; font-size: 1.5em; } .item:first-child { padding-top: 20px; } /* Flex Container */ .container { max-width: 400px; margin: 0 auto; display: flex; border: 1px solid #ccc; } .column { flex-direction: column; } h1 { text-align: center; margin: 20px 0 0 0; font-size: 1.25em; font-weight: normal; } body { font-family: monospace; color: #333; } </pre>
--	--

Saiba mais:

https://youtu.be/KbjLtEgmZ_E

https://youtu.be/hjz6ezV9_uc

https://www.w3schools.com/css/css3_flexbox.asp

<https://origamid.com/projetos/flexbox-guia-completo/>

Divirta-se e aprenda com os jogos:

<https://flexboxfroggy.com/>

<https://dev.to/saragomesdev/5-jogos-gratuitos-para-praticar-css-4pa7>