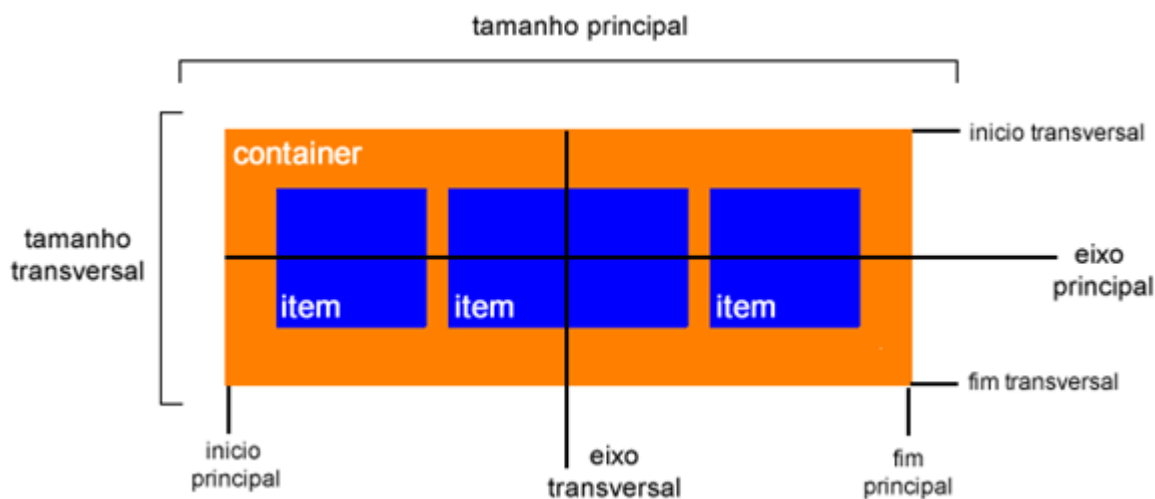
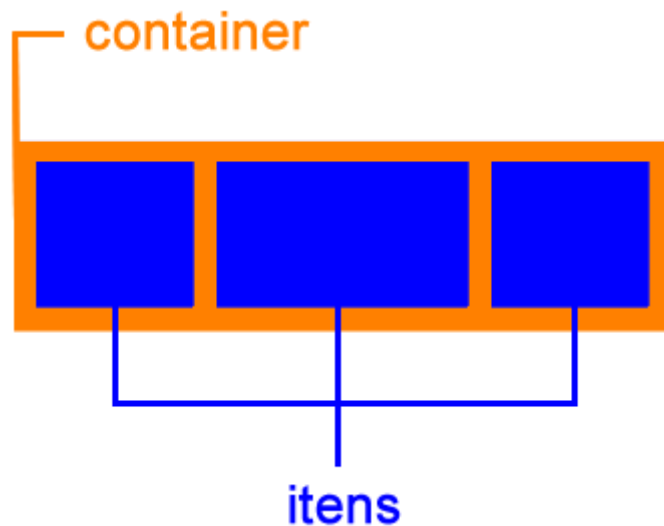


O **Flexbox** é um conjunto de propriedades que tem por objetivo organizar itens dentro de um elemento pai, normalmente chamado de container:

Portanto, para utilizar esse recurso é necessário ter no HTML ao menos um elemento (container) contendo outros (itens), como no código abaixo:

Para entender o **funcionamento do Flexbox** é necessário ter em mente alguns conceitos fundamentais que dizem respeito a como os itens são distribuídos no container. O principal deles é o conceito de eixo principal e eixo transversal, que depende do valor atribuído à propriedade flex-direction. Se essa propriedade receber o valor row ou row-reverse (organização em linhas), o eixo principal do container será o horizontal. Já se essa propriedade receber o valor column ou column-reverse (organização em coluna), o eixo principal será o vertical. Consequentemente isso definirá qual é o eixo transversal. Se o principal for o vertical, o transversal será o horizontal e vice-versa.



- **Tamanho principal:** É a dimensão do elemento na direção do eixo principal (largura, caso horizontal e altura caso vertical);
- **Tamanho transversal:** É a dimensão do elemento na direção do eixo transversal (largura, caso horizontal e altura caso vertical);
- **Início principal e final principal:** Representam o início e o fim do eixo principal;
- **Início transversal e final transversal:** Representam o início e o fim do eixo transversal.

Essas informações serão importantes para compreendermos o funcionamento das diversas **propriedades do Flexbox**

## display

O primeiro passo para **utilizar o Flexbox** é definir a propriedade `display` do container com o valor `flex`. Isso é necessário para que as demais propriedades apresentem o resultado esperado.

A sintaxe de uso dessa propriedade é a seguinte:

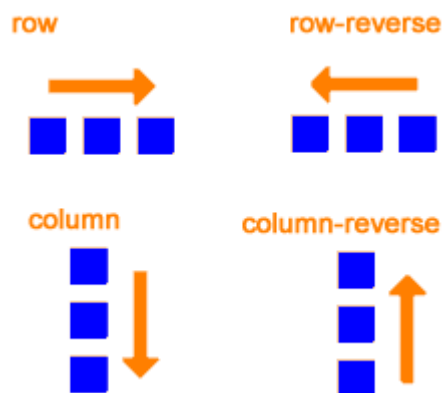
```
.container {  
  display: flex;  
}
```

## flex-direction

A propriedade `flex-direction` deve ser aplicada ao container e define o eixo/fluxo de exibição em que os elementos serão organizados. A sintaxe e os valores possíveis para essa propriedade são apresentados a seguir:

```
.container {  
  display: flex;  
  flex-direction: [row / row-reverse / column / column-reverse];  
}
```

- **row** (padrão): Os itens são organizados em forma de linha da esquerda para a direita;
- **row-reverse**: Os itens são organizados em forma de exibição em linha da direita para a esquerda;
- **column**: Os itens são organizados em forma de colunas iniciando de cima para baixo;
- **column-reverse**: Os itens são organizados em forma de colunas iniciando de baixo para cima.



## flex-wrap

Por padrão os itens do container tentarão se ajustar em uma única linha dentro do container, mas para isso a sua largura original pode ser ajustada para que todos caibam na largura do elemento pai. Com a propriedade `flex-wrap` aplicada ao container podemos alterar esse comportamento, fazendo com que ocorra a “quebra de linha” nos itens.

A sintaxe e os valores possíveis para essa propriedade são apresentados a seguir:

```
.container {  
  display: flex;  
  flex-wrap: [nowrap / wrap / wrap-reverse];  
}
```

- **nowrap** (padrão): Todos os itens serão dispostos em uma linha;
- **wrap**: Ocorrerá a quebra de linha e os itens mais à direita serão deslocados para a linha de baixo;
- **wrap-reverse**: Ocorrerá a quebra de linha e os itens mais à direita serão deslocados para a linha de cima;

## flex-flow

Esta propriedade é uma forma abreviada para a escrita das propriedades flex-direction e flex-wrap, nesta ordem. Portanto, ela se aplica ao container.

Normalmente essas propriedades são definidas uma a uma, da seguinte forma:

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

Já com o flex-flow podemos escrever as duas de forma simplificada:

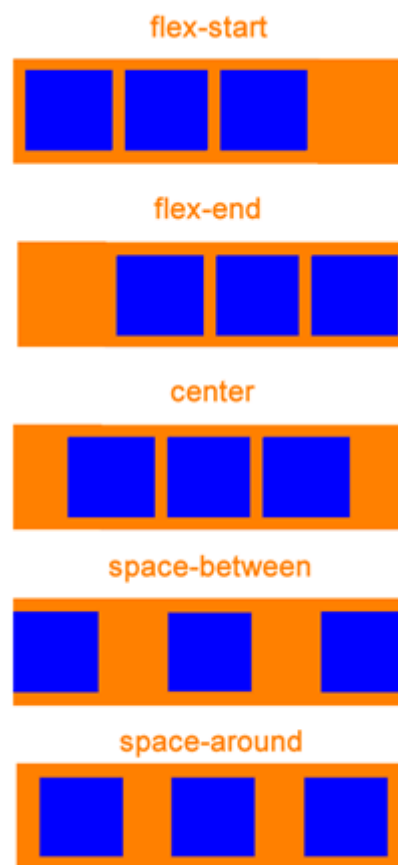
```
flex-flow: column wrap;
```

## justify-content

A propriedade justify-content define o alinhamento dos itens ao longo do eixo principal do container. A sintaxe e os valores possíveis para essa propriedade são apresentados a seguir:

```
.container {  
  display: flex;  
  justify-content: [flex-start/flex-end/center/space-between/space-around];  
}
```

- **flex-start (padrão):** Os itens são alinhados a partir do início do eixo principal;
- **flex-end:** Os itens são alinhados a partir do fim do eixo principal;
- **center:** Os itens são alinhados ao centro do eixo principal;
- **space-between:** O primeiro item é deslocado para o início do eixo principal, o último é deslocado para o final do eixo principal e os demais são distribuídos uniformemente entre eles;
- **space-around:** Os itens são uniformemente distribuídos ao longo do eixo principal. Aqui, porém, são atribuídas margens iguais à esquerda e à direita (ou acima e abaixo, dependendo da direção do eixo principal). Por isso o primeiro e o último item não ficam “colados” nas bordas do container.

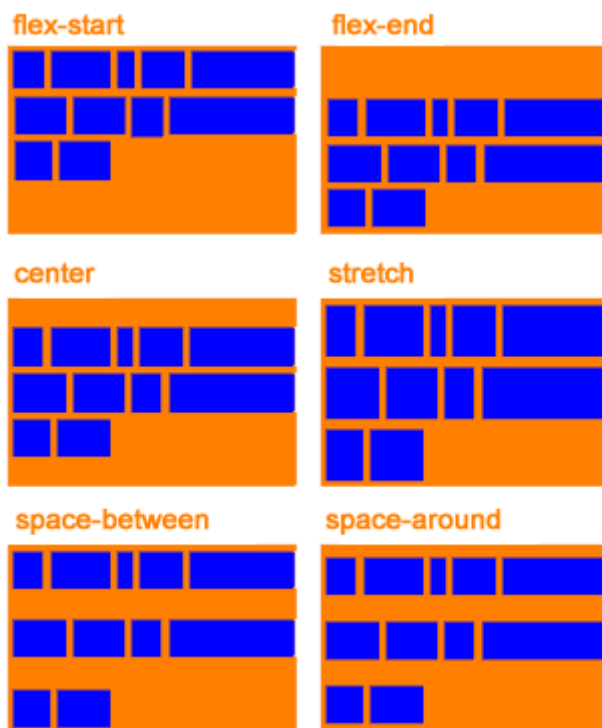


## align-content

Essa propriedade define como as linhas são distribuídas ao longo do eixo transversal do container. Ela só terá efeito se o elemento tiver mais de uma linha, ou seja, se ele tiver elementos suficientes para quebrar a linha e a propriedade `flex-wrap: wrap` tiver sido definida. A sintaxe e os valores possíveis para essa propriedade são apresentados a seguir:

```
.container {  
  display: flex;  
  align-content: [stretch/flex-start/flex-end/center/space-between/space-around];  
}
```

- **stretch** (padrão): As linhas são distribuídas uniformemente ao longo do eixo transversal, ocupando todo o espaço disponível;
- **flex-start**: As linhas são distribuídas a partir do início do eixo transversal;
- **flex-end**: As linhas são distribuídas a partir do fim do eixo transversal;
- **center**: As linhas são mantidas no centro do eixo transversal;
- **space-between**: A primeira linha é deslocada para o início do eixo transversal, a última é deslocada para o final do eixo transversal e as demais são distribuídas uniformemente entre eles;
- **space-around**: As linhas são uniformemente distribuídas ao longo do eixo transversal. Aqui, porém, são atribuídas margens iguais à esquerda e à direita (ou acima e abaixo, dependendo da direção do eixo transversal). Por isso a primeira e a última linha não ficam “coladas” nas bordas do container.



## align-items

Essa propriedade define como os itens são distribuídos ao longo do eixo transversal do container. A sintaxe e os valores possíveis para essa propriedade são apresentados a seguir:

```
.container {  
  display: flex;  
  align-items: [stretch/flex-start/flex-end/center/baseline];  
}
```

- **stretch** (padrão): Os itens serão esticados para preencher toda a dimensão do eixo transversal (altura ou largura);
- **flex-start**: Os itens são deslocados para o início do eixo transversal;
- **flex-end**: Os itens são deslocados para o final do eixo transversal;
- **center**: Os itens são centralizados no eixo transversal;
- **baseline**: Os itens são alinhados a partir da base da primeira linha de texto de cada um.

## Order

Por padrão, os itens são distribuídos no container na ordem em que são inseridos no HTML. No entanto, essa ordem pode ser alterada por meio da propriedade `order`, cuja sintaxe vemos abaixo:

```
.item2 {
  order: [número];
}
```

O valor numérico atribuído a essa propriedade define a ordem do item. Por exemplo, o valor 2 faz com que o item seja o segundo item ao longo do eixo principal, enquanto o valor -1 faz com que ele apareça antes do primeiro.

## flex-grow

Esta propriedade define a proporção com que um item deve crescer caso seja necessário. Por padrão seu valor é 0, o que indica que o item não deve crescer, e são aceitos apenas valores numéricos positivos.

A sintaxe dessa propriedade é a seguinte:

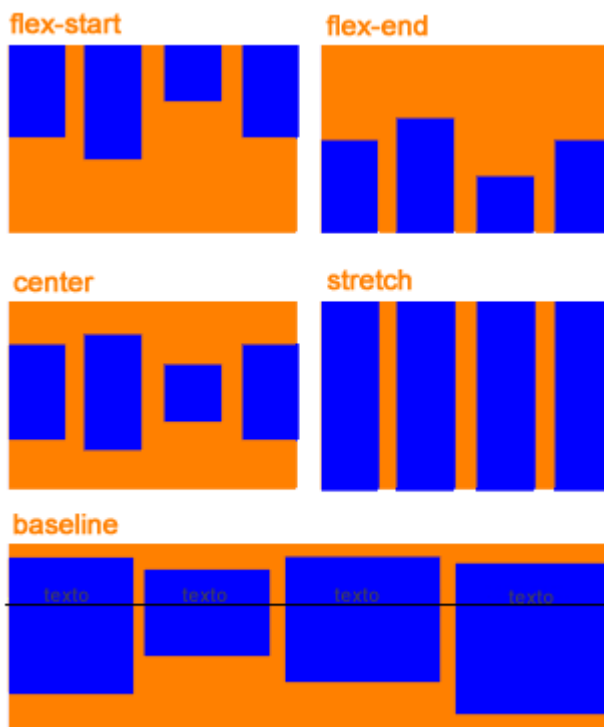
```
.item2 {
  flex-grow: [número];
}
```

## flex-shrink

Esta propriedade define a proporção com que um item deve encolher caso seja necessário. Essa propriedade aceita apenas valores positivos, e seu valor padrão é 1.

A sintaxe dessa propriedade é a seguinte:

```
.item2 {
  flex-shrink: [número];
}
```



## flex-basis

O flex-basis define o tamanho inicial que um item deve ter antes que o espaço ao seu redor seja distribuído. Ou seja, dependendo da direção do eixo principal (horizontal ou vertical), essa propriedade define a largura ou altura mínima do elemento antes que ele seja redimensionado por outras propriedades.

A sintaxe dessa propriedade é a seguinte:

```
.item2 {  
    flex-basis: [número];  
}
```

O valor atribuído a essa propriedade pode ser em percentual, em pixels, ou a palavra auto, que é o valor padrão (considera as dimensões do item - width e height).

## flex

Esta propriedade é uma forma abreviada para a escrita das propriedades flex-grow, flex-shrink e flex-basis, nesta ordem.

A sintaxe dessa propriedade é a seguinte:

```
.item2 {  
    flex: [flex-grow] [flex-shrink] [flex-basis];  
}
```

## align-self

Esta propriedade permite sobrescrever no item o comportamento que foi definido pela propriedade align-items.

A sintaxe e os valores possíveis para essa propriedade são apresentados a seguir:

```
.item2 {  
    align-self: [auto/stretch/flex-start/flex-end/center/baseline];  
}
```

- **auto** (padrão): Respeita o comportamento definido no container por meio do align-items;
- **stretch**: O item será esticado para preencher toda a dimensão do eixo transversal (altura ou largura);
- **flex-start**: O item é deslocado para o início do eixo transversal;
- **flex-end**: O item é deslocado para o final do eixo transversal;
- **center**: O item é centralizado no eixo transversal;
- **baseline**: O item é alinhado a partir da base da primeira linha de texto dos demais.

## Propriedades do Container Flex (aplicadas no pai)

### display

- flex → ativa o flexbox.
  - inline-flex → ativa, mas o container se comporta como elemento inline.
- 

### flex-direction

Define a direção do eixo principal:

- row (padrão, itens em linha, da esquerda → direita)
  - row-reverse (linha invertida)
  - column (coluna, de cima → baixo)
  - column-reverse (coluna invertida)
- 

### flex-wrap

Define se os itens quebram de linha:

- nowrap (padrão, não quebra)
  - wrap (quebra quando falta espaço)
  - wrap-reverse (quebra invertendo a ordem vertical)
- 

### flex-flow

Atalho para flex-direction + flex-wrap.

Exemplo:

```
flex-flow: row wrap;
```

---

### justify-content (alinhamento no eixo principal → linha ou coluna, depende do flex-direction)

- flex-start (padrão, tudo grudado no início)
- flex-end (no fim)
- center (centralizado)
- space-between (espaço apenas entre os itens)
- space-around (espaço em volta de cada item)
- space-evenly (espaço igual em tudo)

---

### **align-items (alinhamento no eixo cruzado → perpendicular ao principal)**

- stretch (padrão, itens esticam para ocupar altura/largura)
- flex-start (alinhados ao topo/esquerda)
- flex-end (alinhados ao fundo/direita)
- center (centraliza no eixo cruzado)
- baseline (alinha pela linha de base do texto)

---

### **align-content (só funciona quando há múltiplas linhas → com flex-wrap)**

- stretch (padrão, ocupa todo espaço disponível)
- flex-start (linhas no topo/esquerda)
- flex-end (linhas no fundo/direita)
- center (linhas no meio)
- space-between (espaço entre linhas)
- space-around (espaço em volta das linhas)

---

## **Propriedades dos Itens Flex (aplicadas nos filhos)**

### **order**

Muda a ordem de exibição (padrão = 0).

---

### **flex-grow**

Define se o item **cresce** para ocupar espaço sobrando.

- 0 (padrão, não cresce)
- 1 (cresce proporcionalmente)

---

### **flex-shrink**

Define se o item **encolhe** quando falta espaço.

- 1 (padrão, encolhe)
- 0 (não encolhe nunca)

---

### **flex-basis**

Define o **tamanho inicial** do item antes do ajuste.

- pode ser auto, px, %, etc.
-



## flex

Atalho para: flex-grow flex-shrink flex-basis.

Exemplo:

```
flex: 1 0 200px; /* cresce 1x, não encolhe, tamanho base 200px */
```

---

## align-self

Permite sobrescrever o align-items só para um item.

- auto (usa o valor do pai)
  - flex-start, flex-end, center, baseline, stretch
- 

## Resumo

### Container (pai):

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

### Itens (filhos):

- order
- flex-grow
- flex-shrink
- flex-basis
- flex (atalho)
- align-self

## <!--exemplo flex.html-->

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <title>Guia Flexbox Visual</title>
  <link rel="stylesheet" href="../css/exemplo_flex.css">
</head>

<body>

  <h1 style="text-align:center;">Guia Prático de Flexbox</h1>
  <ul>
    <li><strong>Flex-direction:</strong> itens lado a lado (row) e empilhados (column)</li>
    <li><strong>Justify-content:</strong> mostra diferentes alinhamentos no eixo principal</li>
    <li><strong>Align-items:</strong> mostra alinhamento vertical (topo, centro, esticados)</li>
    <li><strong>Flex-grow:</strong> itens crescem proporcionalmente ao espaço disponível</li>
  </ul>

  <!-- EXEMPLO 1: flex-direction -->
  <section class="exemplo">
    <h2>flex-direction</h2>
    <div class="container flex-direction row">
      <div class="item">1</div>
      <div class="item">2</div>
      <div class="item">3</div>
      <div class="item">4</div>
      <div class="item">5</div>
      <div class="item">6</div>
    </div>
    <div class="container flex-direction column">
      <div class="item">1</div>
      <div class="item">2</div>
      <div class="item">3</div>
      <div class="item">4</div>
      <div class="item">5</div>
      <div class="item">6</div>
    </div>
  </section>

  <!-- EXEMPLO 2: justify-content -->
  <section class="exemplo">
    <h2>justify-content</h2>
    <div class="container justify-content flex-start">
      <div class="item">A</div>
      <div class="item">B</div>
      <div class="item">C</div>
      <div class="item">D</div>
      <div class="item">E</div>
      <div class="item">F</div>
    </div>
```

```
<div class="container justify-content space-between">
<div class="item">A</div>
<div class="item">B</div>
<div class="item">C</div>
<div class="item">D</div>
<div class="item">E</div>
<div class="item">F</div>
</div>
<div class="container justify-content center">
<div class="item">A</div>
<div class="item">B</div>
<div class="item">C</div>
<div class="item">D</div>
<div class="item">E</div>
<div class="item">F</div>
</div>
</section>
```

```
<!-- EXEMPLO 3: align-items -->
<section class="exemplo">
<h2>align-items</h2>
<div class="container align-items flex-start" style="height:100px;">
<div class="item tall">1</div>
<div class="item">2</div>
<div class="item">3</div>
<div class="item">4</div>
<div class="item">5</div>
<div class="item">6</div>
</div>
<div class="container align-items center" style="height:100px;">
<div class="item tall">1</div>
<div class="item">2</div>
<div class="item">3</div>
<div class="item">4</div>
<div class="item">5</div>
<div class="item">6</div>
</div>
<div class="container align-items stretch" style="height:100px;">
<div class="item tall">1</div>
<div class="item">2</div>
<div class="item">3</div>
<div class="item">4</div>
<div class="item">5</div>
<div class="item">6</div>
</div>
</section>
```

```
<!-- EXEMPLO 4: flex-grow / flex-shrink / flex-basis -->
<section class="exemplo">
<h2>flex-grow, flex-shrink e flex-basis</h2>
<div class="container flex-grow-example">
<div class="item" style="flex:1;">1</div>
<div class="item" style="flex:1;">1</div>
```

```
<div class="item" style="flex:2;">2</div>
<div class="item" style="flex:2;">2</div>
<div class="item" style="flex:3;">3</div>
<div class="item" style="flex:6;">6</div>
</div>
</section>
```

```
</body>
```

```
</html>
```

```
/* ===== exemplo_flex.css ===== */
```

```
body {  
  margin: 0;  
  font-family: Arial, sans-serif;  
  padding: 20px;  
  background: #f5f5f5;  
}
```

```
h1 {  
  margin-bottom: 40px;  
}
```

```
.exemplo {  
  margin-bottom: 50px;  
}
```

```
.container {  
  display: flex;  
  background: #ddd;  
  padding: 10px;  
  margin: 10px 0;  
  min-height: 50px;  
  border-radius: 8px;  
}
```

```
/* ===== FLEX-DIRECTION ===== */
```

```
.flex-direction.row {  
  flex-direction: row;  
}  
.flex-direction.column {  
  flex-direction: column;  
}
```

```
/* ===== JUSTIFY-CONTENT ===== */
```

```
.justify-content.flex-start {  
  justify-content: flex-start;  
}  
.justify-content.center {  
  justify-content: center;  
}  
.justify-content.space-between {  
  justify-content: space-between;  
}  
.justify-content.space-around {  
  justify-content: space-around;  
}  
.justify-content.space-evenly {  
  justify-content: space-evenly;  
}
```

```
/* ===== ALIGN-ITEMS ===== */
```

```
.align-items.flex-start {
```

```
    align-items: flex-start;
}
.align-items.center {
    align-items: center;
}
.align-items.flex-end {
    align-items: flex-end;
}
.align-items.stretch {
    align-items: stretch;
}
.align-items.baseline {
    align-items: baseline;
}

/* ===== ITENS ===== */
.item {
    background: #4caf50;
    color: white;
    font-weight: bold;
    padding: 20px;
    text-align: center;
    border-radius: 6px;
}

/* Exemplo com altura diferente */
.item.tall {
    height: 70px;
}

/* ===== FLEX-GROW ===== */
.flex-grow-example .item {
    background: #2196f3;
}
```

## <!--visual completo Flexbox.html-->

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <title>Guia Completo de Flexbox</title>
  <link rel="stylesheet" href="../css/visual completo Flexbox.css">
</head>

<body>

  <h1 style="text-align:center;">Guia Completo de Flexbox</h1>

  <!-- ===== FLEX-DIRECTION ===== -->
  <section class="exemplo">
    <h2>flex-direction</h2>
    <div class="container flex-direction row">
      <div class="item">1</div>
      <div class="item">2</div>
      <div class="item">3</div>
      <div class="item">4</div>
      <div class="item">5</div>
      <div class="item">6</div>
    </div>
    <div class="container flex-direction column">
      <div class="item">1</div>
      <div class="item">2</div>
      <div class="item">3</div>
      <div class="item">4</div>
      <div class="item">5</div>
      <div class="item">6</div>
    </div>
  </section>

  <!-- ===== JUSTIFY-CONTENT ===== -->
  <section class="exemplo">
    <h2>justify-content</h2>
    <div class="container justify-content flex-start">
      <div class="item">A</div>
      <div class="item">B</div>
      <div class="item">C</div>
      <div class="item">D</div>
      <div class="item">E</div>
      <div class="item">F</div>
    </div>
    <div class="container justify-content center">
      <div class="item">A</div>
      <div class="item">B</div>
      <div class="item">C</div>
      <div class="item">D</div>
    </div>
  </section>
</body>
</html>
```

```
<div class="item">E</div>
<div class="item">F</div>
</div>
<div class="container justify-content space-between">
  <div class="item">A</div>
  <div class="item">B</div>
  <div class="item">C</div>
  <div class="item">D</div>
  <div class="item">E</div>
  <div class="item">F</div>
</div>
</section>
```

```
<!-- ===== ALIGN-ITEMS ===== -->
<section class="exemplo">
<h2>align-items</h2>
<div class="container align-items flex-start" style="height:100px;">
  <div class="item tall">1</div>
  <div class="item">2</div>
  <div class="item">3</div>
  <div class="item">4</div>
  <div class="item">5</div>
  <div class="item">6</div>
</div>
<div class="container align-items center" style="height:100px;">
  <div class="item tall">1</div>
  <div class="item">2</div>
  <div class="item">3</div>
  <div class="item">4</div>
  <div class="item">5</div>
  <div class="item">6</div>
</div>
<div class="container align-items stretch" style="height:100px;">
  <div class="item tall">1</div>
  <div class="item">2</div>
  <div class="item">3</div>
  <div class="item">4</div>
  <div class="item">5</div>
  <div class="item">6</div>
</div>
</section>
```

```
<!-- ===== ORDER ===== -->
<section class="exemplo">
<h2>order</h2>
<div class="container order-example">
  <div class="item" style="order:3">1</div>
  <div class="item" style="order:1">2</div>
  <div class="item" style="order:2">3</div>
</div>
</section>
```

```
<!-- ===== FLEX-GROW / FLEX-SHRINK / FLEX-BASIS ===== -->
```



```
<section class="exemplo">
<h2>flex-grow / flex-shrink / flex-basis</h2>
<div class="container flex-example">
  <div class="item" style="flex:1 1 100px;">1</div>
  <div class="item" style="flex:2 1 100px;">2</div>
  <div class="item" style="flex:1 1 100px;">3</div>
</div>
</section>

<!-- ===== ALIGN-SELF ===== -->
<section class="exemplo">
<h2>align-self</h2>
<div class="container align-items flex-start" style="height:100px;">
  <div class="item" style="align-self:flex-start;">A</div>
  <div class="item" style="align-self:center;">B</div>
  <div class="item" style="align-self:flex-end;">C</div>
  <div class="item">D</div>
</div>
</section>

</body>

</html>
```

```
/* ===== visual completo Flexbox.css ===== */
```

```
body {  
    margin: 0;  
    font-family: Arial, sans-serif;  
    padding: 20px;  
    background: #f5f5f5;  
}
```

```
h1 {  
    margin-bottom: 40px;  
}
```

```
.exemplo {  
    margin-bottom: 50px;  
}
```

```
/* ===== CONTAINERS FLEX ===== */
```

```
.container {  
    display: flex;  
    background: #ddd;  
    padding: 10px;  
    margin: 10px 0;  
    border-radius: 8px;  
}
```

```
/* FLEX-DIRECTION */  
.flex-direction.row {  
    flex-direction: row;  
}
```

```
.flex-direction.column {  
    flex-direction: column;  
}
```

```
/* JUSTIFY-CONTENT */  
.justify-content.flex-start {  
    justify-content: flex-start;  
}
```

```
.justify-content.center {  
    justify-content: center;  
}
```

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

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

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

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

/* ALIGN-ITEMS */
.align-items.flex-start {
  align-items: flex-start;
}

.align-items.center {
  align-items: center;
}

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

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

.align-items.baseline {
  align-items: baseline;
}

/* ITEMS FLEX */
.item {
  background: #4caf50;
  color: white;
  font-weight: bold;
  padding: 20px;
  text-align: center;
  border-radius: 6px;
  margin: 5px;
}

.item.tall {
  height: 70px;
}

/* ORDER */
.order-example .item {
  background: #2196f3;
}

/* FLEX-GROW / FLEX-SHRINK / FLEX-BASIS */
.flex-example .item {
  background: #ff9800;
}
```

```
/* ALIGN-SELF */  
.align-items .item {  
  background: #9c27b0;  
}
```