

Você está em

Guia de CSS » Flexbox

Artigo

Invista em você! Saiba como a DevMedia pode ajudar sua carreira.



CSS3 Flexbox: Funcionamento e propriedades

Nesta documentação veremos o funcionamento do flexbox no CSS3, suas propriedades e exemplos de aplicações e como criar layouts flexíveis (responsivos) com CSS3.



Marcar como concluído



Anotar

Artigos



HTML e CSS



CSS3 Flexbox: Funcionamento e propriedades



Você está em

Guia de CSS » Flexbox

O **Flexbox** é um conceito do **CSS3** que visa organizar os elementos de uma página HTML dentro de seus containers de forma dinâmica. Ou seja, independente das suas dimensões eles sempre manterão um layout flexível dentro do seu elemento pai, reorganizando-se de acordo com a necessidade.

Neste documento conheceremos as **propriedades do CSS** que fazem parte desse recurso, analisando seu funcionamento, bem como exemplos práticos de uso.

Saiba mais: [CSS Tutorial Completo](#)

Tópicos

[Conceitos iniciais](#)

[display](#)

[flex-direction](#)

[flex-wrap](#)

Você está em

[Guia de CSS](#) » [Flexbox](#)[order](#)[flex-grow](#)[flex-shrink](#)[flex-basis](#)[flex](#)[align-self](#)[Exemplo prático](#)

Conceitos iniciais

O **Flexbox** é um conjunto de propriedades que tem por objetivo organizar itens dentro de um elemento pai, normalmente chamado de container, conforme ilustra a **Figura 1**.

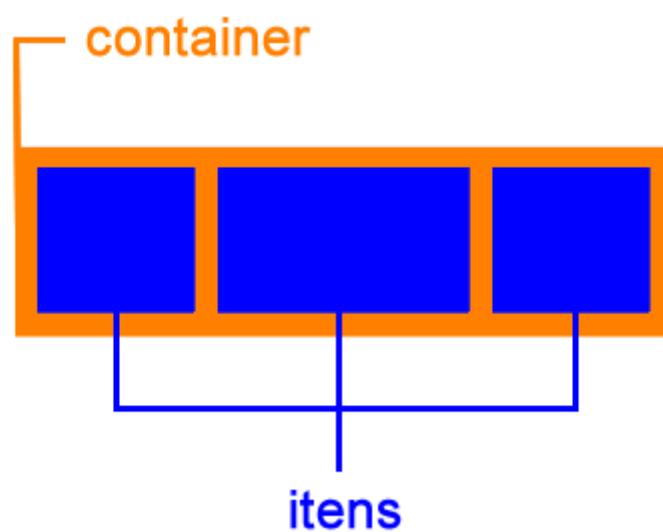


Figura 1 Estrutura dos elementos para aplicação do Flexbox

```
1 <div class="container">
2   <div class="item item1"></div>
3   <div class="item item2"></div>
4   <div class="item item3"></div>
5 </div>
```

Conforme veremos a seguir, algumas propriedades serão aplicadas ao container, enquanto outras serão aplicadas aos itens.

Alinhamento dos eixos

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.

Na **Figura 2** podemos ver esse e outros conceitos ilustrados para o caso de

Você está em

Guia de CSS » Flexbox

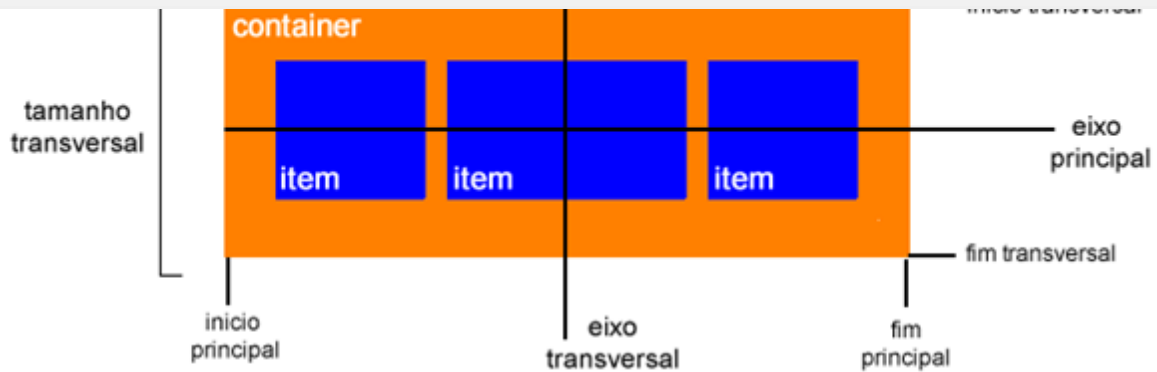


Figura 2. Alinhamento dos eixos do Flexbox

- **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** que veremos a seguir.

display

O primeiro passo para utilizar o Flexbox é definir a propriedade `display`

Você está em

Guia de CSS » Flexbox

```
1 | .container {  
2 |     display: flex;  
3 | }
```

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:

```
1 | .container {  
2 |     display: flex;  
3 |     flex-direction: [row / row-reverse / column / column-reverse]  
4 | }
```

- **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;

A **Figura 3** ilustra o funcionamento de cada valor.

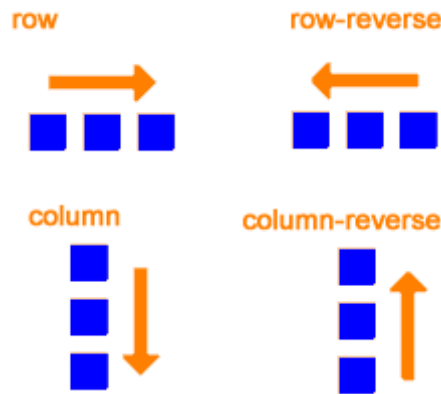


Figura 3. Funcionamento da propriedade flex-direction

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:

Você está em

Guia de CSS » Flexbox

- **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:

```
1  .container {  
2    display: flex;  
3    flex-direction: column;  
4    04 flex-wrap: wrap;  
5    05 }
```

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

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:

```
1 .container {  
2     display: flex;  
3     justify-content: [flex-start/flex-end/center/space-between/space-around/space-between];  
4 }
```

- **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).



Você está em

Guia de CSS » Flexbox

A **Figura 4** ilustra o funcionamento de cada valor:

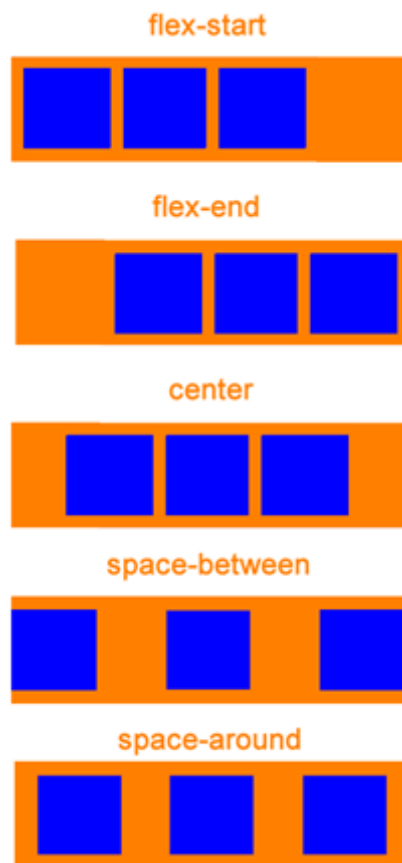


Figura 4 Representação da propriedade justify-content

align-content

Essa propriedade define como as linhas são distribuídas ao longo do eixo transversal do container. A sintaxe e os valores possíveis para essa

Você está em

Guia de CSS » Flexbox

```
3 align-content: [stretch/flex-start/flex-end/center/space-between/space-around];  
4 }
```

- **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.

A **Figura 5** ilustra o funcionamento de cada valor:

Você está em

Guia de CSS » Flexbox

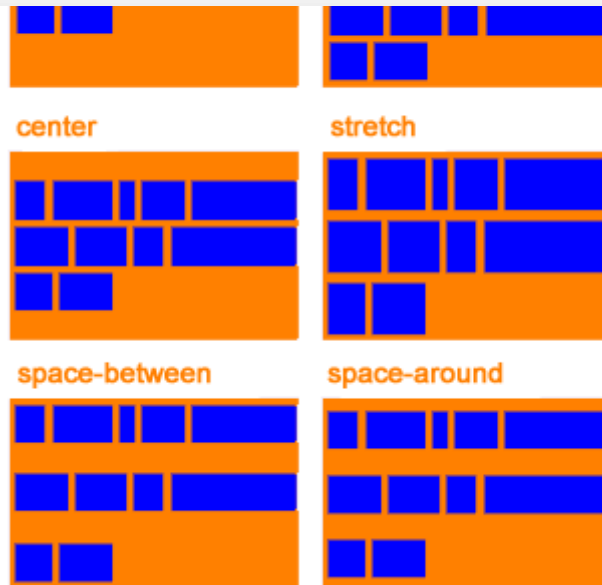


Figura 5. Funcionamento da propriedade align-content

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:

```
1  .container {  
2    display: flex;  
3    align-items: [stretch/flex-start/flex-end/center/baseline];  
4  }
```

Você está em

Guia de CSS » Flexbox

- **flex-start:** Os itens são deslocadas para o início do eixo transversal;
- **flex-end:** Os itens são deslocadas 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.

A **Figura 6** ilustra o funcionamento de cada valor:

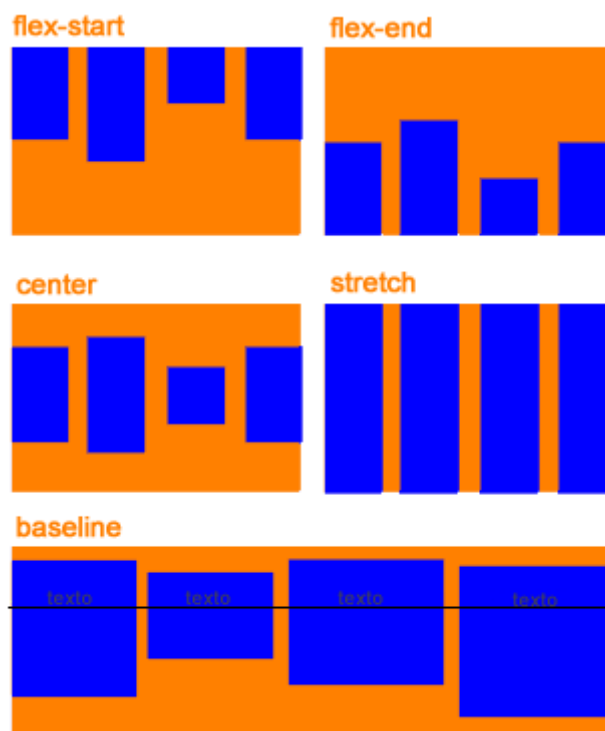


Figura 6. Representação da propriedade align-items

Order

Você está em

Guia de CSS » Flexbox

propriedade `order`, cuja sintaxe vemos abaixo:

```
1 | .item2 {  
2 |   order: [número]  
3 | }
```

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:

```
1 | .item2 {  
2 |   flex-grow: [número]  
3 | }
```

Você está em

Guia de CSS » Flexbox

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

A sintaxe dessa propriedade é a seguinte:

```
1 | .item2 {  
2 |     flex-shrink: [número]  
3 | }
```

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:

```
1 | .item2 {  
2 |     flex-basis: [número]  
3 | }
```

Você está em

Guia de CSS » Flexbox

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:

```
1 .item2 {  
2   flex: [flex-grow] [flex-shrink] [flex-basis]  
3 }
```

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:

Você está em

Guia de CSS » Flexbox

- **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.

Exemplo prático

A seguir podemos ver um exemplo prático de uso do Flexbox.

Primeiramente temos a estrutura do HTML na qual foram aplicadas as propriedades que vimos anteriormente:

```
1 <div class="container">
2   <div class="item item1">1</div>
3   <div class="item item2">2</div>
4   <div class="item item3">3</div>
5 </div>
```

Você está em

Guia de CSS » Flexbox

```
1  .container {
2      background-color: #FF5722;
3      height: 100vh;
4      width: 100%;
5      display: flex;
6      flex-direction: column;
7      justify-content: center;
8      align-items: center;
9  }
10
11  .item {
12      background-color: #0000ff;
13      color: #fff;
14      padding: 10px;
15      margin: 10px 0;
16  }
17
18  .item1{
19      height: 100px;
20      width: 100px;
21  }
22
23  .item2{
24      height: 50px;
25      width: 150px;
26  }
27
28  .item3{
29      height: 100px;
30      width: 200px;
31  }
```

Run

Você está em

Guia de CSS » Flexbox

Linhas 3 e 4: Definimos a altura e largura do container para ocupar 100% da página;

Linha 5: Definimos o display flex no container para que as demais propriedades surtam o efeito esperado;

Linha 6: Com a propriedade flex-direction definida como column definimos que os itens devem ser organizados em coluna (um abaixo do outro);

Linha 7: Nesta linha definimos que os itens serão centralizados no eixo principal (vertical);

Linha 8: A propriedade align-items define aqui que os itens serão centralizados também ao longo do eixo transversal (horizontal);

Linhas 11 a 30: Neste trecho estamos definindo algumas características dos itens. Eles possuem diferentes alturas e larguras, o que facilita a visualização do resultado final (mesmo com diferentes dimensões, o layout é mantido organizado).

Curso relacionado: **CSS3 Display Flex: Como organizar cards e boxes automaticamente**

Tecnologias:

CSS

HTML



Marcar como concluído



Anotar



Você está em

Guia de CSS » Flexbox

por apenas R\$49,90 por mês

[Assine agora](#)Por **Fernando**

Em 2013

Suporte ao aluno - Tire a sua dúvida.



Poste aqui a sua dúvida, nessa seção só você e o consultor podem ver os seus comentários.

[Enviar dúvida](#)[Planos de estudo](#)[Fale conosco](#)[Assinatura para empresas](#)[Assine agora](#)

Você está em

Guia de CSS » Flexbox



218

