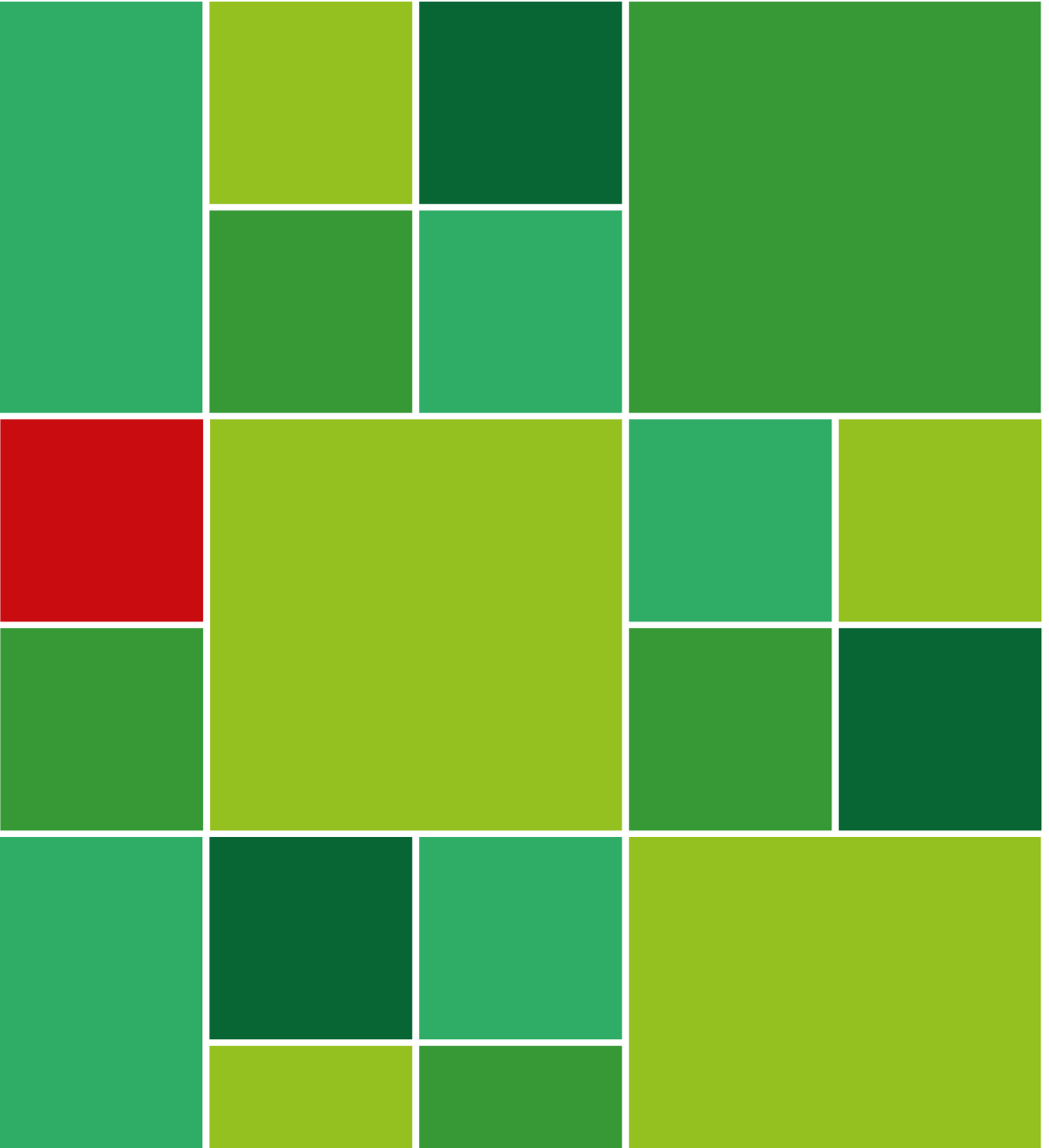


Flexbox

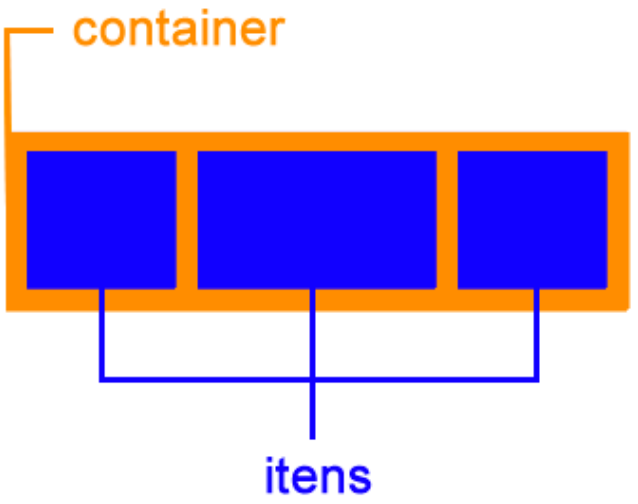
Desenvolvimento Web

David Paolini Develly

INSTITUTO FEDERAL
Espírito Santo

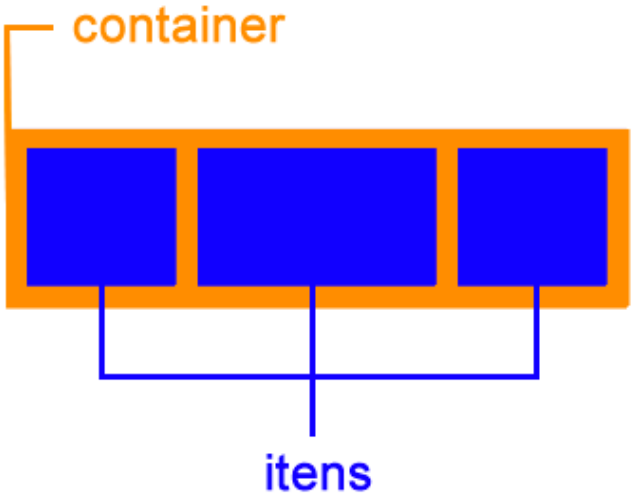
Definição

Conforme DevMedia, 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 e conforme a necessidade.



Definição

Na prática, há propriedades que se aplicam ao container e outras que se aplicam aos itens.



Definição

Para utilizar-lo é necessário ter no HTML ao menos um elemento (container) contendo outros (itens) :

```
<div class="container">
     <div class="item item1"></div>
     <div class="item item2"></div>
     <div class="item item3"></div>
</div>
```

Aplicação

Para utilizar-lo é necessário ter no HTML ao menos um elemento (container) contendo outros (itens) :

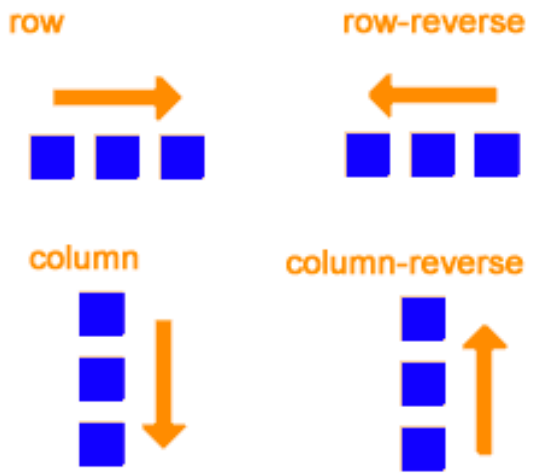
```
<div class="container">
    <div class="item item1"></div>
    <div class="item item2"></div>
    <div class="item item3"></div>
</div>
```

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

flex-direction

Define o eixo/fluxo de exibição em que os elementos serão organizados.

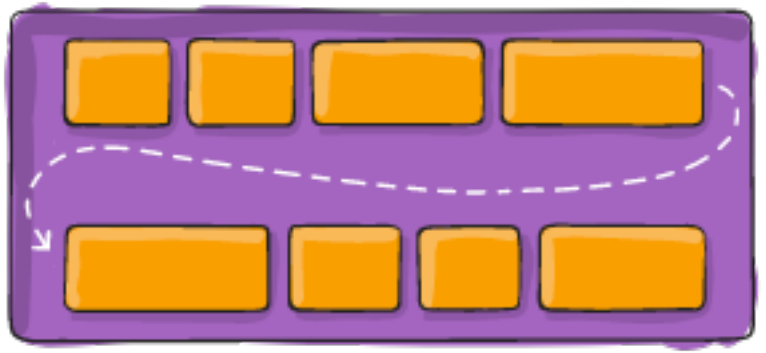
```
.container {
    display: flex;
    flex-direction: [row / row-reverse / column / column-reverse];
 }
```



flex-rap

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.

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



flex-flow

Forma abreviada para a escrita das propriedades flex-direction e flex-wrap, nesta ordem.

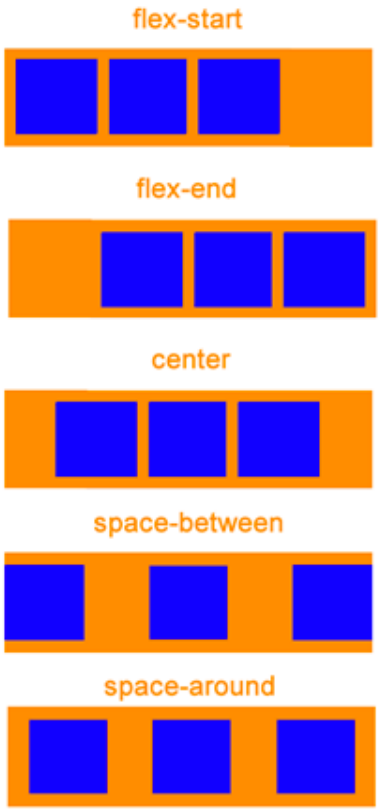
```
.container {
  display: flex;
 flex-direction: column;
04  flex-wrap: wrap;
05 }
```

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

justify-content

Define o alinhamento dos itens ao longo do eixo principal do container.

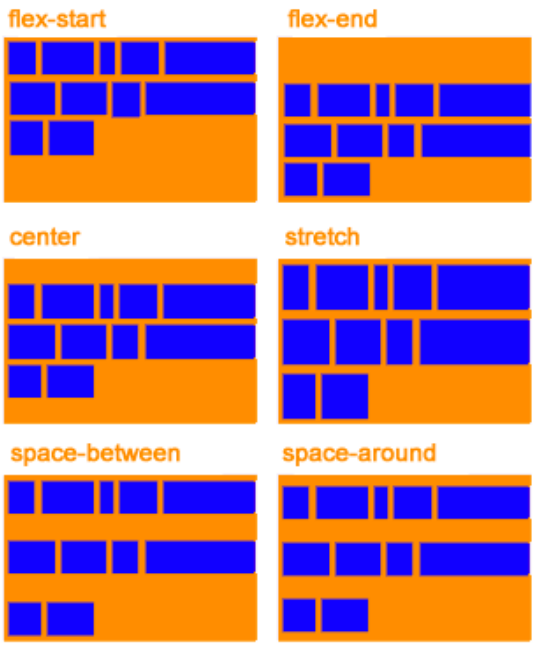
```
.container {
    display: flex;
    justify-content: [flex-start/flex-end/center/space-between/space-around];
}
```



align-content

Define como as linhas são distribuídas ao longo do eixo secundário do container.

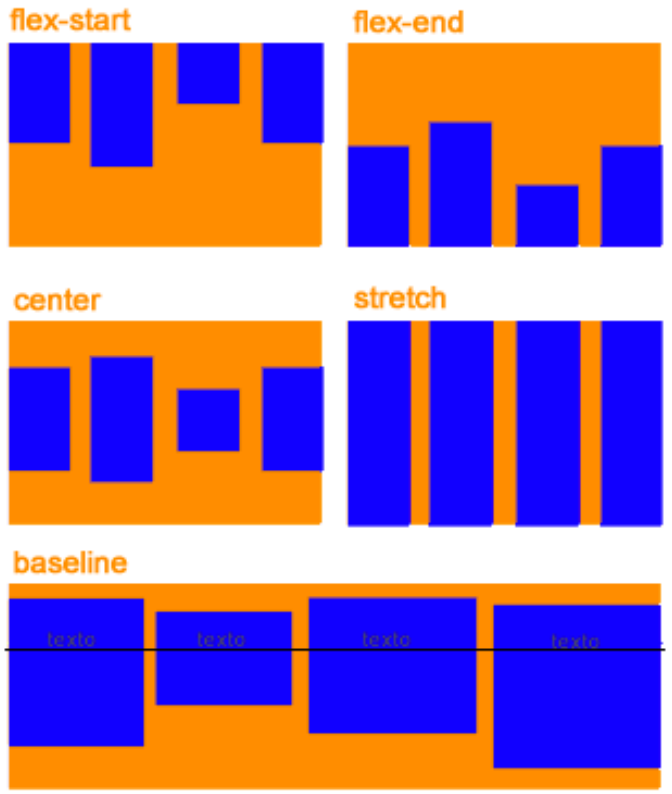
```
.container {
    display: flex;
    align-content: [stretch/flex-start/flex-end/center/space-between/space-around];
}
```



align-items

Define como os itens são distribuídos ao longo do eixo secundário do container.

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



Aplicação nos items

Até agora, todas as propriedades que vimos foram aplicadas no container. Apresentaremos agora as propriedades dos items.

```
<div class="container">
    <div class="item item1"></div>
    <div class="item item2"></div>
    <div class="item item3"></div>
</div>
```

order

Por padrão, os itens são distribuídos no container na ordem em que são inseridos no HTML, tendo o valor de order 0 por padrão.

```
.item2 {
  order: [número];
}
```

O valor numérico atribuído a essa propriedade define a ordem do item. Quanto mais alta, mais distante do início o item ficará. Também aceita números negativos.

flex-grow

Define a habilidade de um flex item de 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.

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

flex-shrink

Define a capacidade de um item flexível diminuir, se necessário. Normalmente aplicado em items com muito conteúdo.

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

flex-basis

Define o tamanho inicial que um item deve ter antes que o espaço ao seu redor seja distribuído.

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).

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

flex

Forma abreviada para a escrita das propriedades flex-grow, flex-shrink e flex-basis, nesta ordem.

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

align-self

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

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