

05

CSS FlexBox

Por muito tempo, as únicas ferramentas disponíveis para a criação de leiautes em CSS e posicionar elementos com boa compatibilidade entre browsers eram `float` e `position`. Porém, essas ferramentas possuem algumas limitações um pouco frustrantes.

Algumas tarefas que consideramos básicas em um leiaute, como centralização vertical de uma div-filha com relação a uma div-mãe, fazer com que divs-filhas ocupem a mesma quantidade de espaço e colunas terem o mesmo tamanho independente da quantidade de conteúdo interno, podem ser um pouco mais 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: as filhas de um elemento com Flexbox podem se posicionar em qualquer direção e pode ter dimensões flexíveis para se adaptar.

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`.

```
<style>
.flex-container {
  display: flex;
}

</style>

<div class="flex-container">
  <div class="flex-item">1</div>
  <div class="flex-item">2</div>
  <div class="flex-item">3</div>
</div>
```

Flex Item são elementos-filhas do flex container. Quando utilizamos o Flexbox, é muito importante saber quais propriedades são declaradas no elemento-mãe (por exemplo, uma div que irá conter os elementos a serem alinhados) e quais serão declaradas nos elementos-filhas. Normalmente declaramos no elemento-mãe (o `flex-container`) as propriedades que cuidarão do alinhamento dos elementos-filhas (cada `flex-item`).

Lembrando que o FlexBox não enxerga "parentesco" entre elementos além da relação mãe-filha, ou seja, entre elementos diretamente relacionados e aninhados um diretamente dentro do outro.

Para praticar mais, o [Guia do CSS Tricks para FlexBox \(https://css-tricks.com/snippets/css/a-guide-to-flexbox/\)](https://css-tricks.com/snippets/css/a-guide-to-flexbox/). (em inglês) tem as propriedades de Flex bem explicadas com imagens.