



# Aprenda o layout de CSS

[Índice](#)[Tabela de Conteúdos](#)

## posição

De forma a criar layouts mais complexos, precisamos discutir a propriedade `position`. Têm um grupo de valores possíveis, e os seus nomes não fazem sentido e são impossíveis de recordar. Vamos lá passar por eles um a um, talvez devas guardar esta página também.

### static

```
.static {  
  position: static;  
}
```

**CSS**

```
<div class="static">
```

`static` é o valor por defeito. Um elemento com `position: static`; não é posicionado de uma forma qualquer. Diz-se que um elemento estático está *não posicionado* e um elemento com a sua posição com outro valor qualquer diz-se *posicionado*.

```
</div>
```

### relative

```
.relative1 {  
  position: relative;  
}  
.relative2 {  
  position: relative;  
  top: -20px;  
  left: 20px;  
  background-color: white;  
  width: 500px;  
}
```

**CSS**

```
<div class="fixed">  
  Hello! Don't pay  
  attention to me yet.
```

```
</div>
```

```
<div class="relative1">
```

relative comporta-se igual a static a não ser que adicione algumas propriedades extra.

```
<div class="relative2">
```

```
</div>
```

Atribuindo as propriedades top, right, bottom, e left de um elemento relativamente posicionado irá fazer com que seja ajustado para longe da sua posição normal. Outro conteúdo não será ajustado para caber dentro de algum espaço que sobre do elemento.

```
</div>
```

## fixed

Um elemento fixo é posicionado relativamente ao viewport, significa que fica sempre no mesmo sitio, mesmo que faça scroll da página. Com relative, são utilizadas as propriedades top, right, bottom, e left.

Tenho a certeza que notaste o elemento fixo no canto inferior direito da página. Estou a dar-te permissão para lhe dares atenção agora. Aqui está o CSS que o coloca lá:

```
.fixed {  
  position: fixed;  
  bottom: 0;  
  right: 0;  
  width: 200px;  
  background-color: white;  
}
```

CSS

Um elemento fixo não deixa espaço na página onde estaria normalmente localizado.

Browsers de dispositivos móveis têm um apoio instável para fixar. [Aprender aqui mais sobre a situação.](#)

## absolute

absolute é o valor de posicionamento mais complicado. absolute comporta-se como fixed só que relativo *ao posicionamento do antecessor mais próximo* em vez de relativo ao viewport. Se um elemento posicionado absolutamente não tem um antecessor posicionado, utiliza o corpo do documento, e continua a mover-se igualmente com o scroll da página. Lembra-te, um elemento "posicionado" é aquele em que a posição é tudo excepto static.

Aqui está um exemplo simples:

```
<div class="fixed">
```

Hello! Don't pay  
attention to me yet.

```
</div>
```

```
.relative {  
  position: relative;  
  width: 600px;  
  height: 400px;  
}  
.absolute {  
  position: absolute;  
  top: 120px;  
  right: 0;  
  width: 300px;  
  height: 200px;  
}
```

```
<div class="relative">
```

Este elemento está posicionado relativamente. Se este elemento tivesse `position: static;` o seu filho posicionado absolutamente iria escapar e ficaria posicionado relativo ao corpo do documento.

```
<div class="absolute">
```

Este elemento está posicionado absolutamente. Está posicionado relativamente ao seu pai.

```
</div>
```

```
</div>
```

Estas coisas são complicadas, mas é essencial para criar grandes layouts com CSS. Na próxima página vamos utilizar `position` num exemplo mais prático.

[Anterior](#)[Próximo](#)

7 / 19

[Tweet](#)[Like 8.1K](#)[G+](#)[CC BY](#)Ads by *Bocoup*

```
<div class="fixed">  
  Hello! Don't pay  
  attention to me yet.  
</div>
```