



CSS position: configurando a posição dos elementos na página!

# CSS position: configurando a posição dos elementos na página!

Última atualização 24 de junho de 2021

No [CSS](#), o **CSS position** é uma propriedade utilizada para determinar o posicionamento dos elementos na página. Ela é usada em conjunto com propriedades auxiliares, entre elas: **bottom**, **top**, **left** e **right**, que ajudam a definir de que forma o elemento será posicionado na tela.

É importante entender como o CSS position funciona, pois essa propriedade pode sobrepor elementos, além de ser utilizada para aplicar [efeitos interessantes](#), como a fixação do cabeçalho ao rolar a página. Para demonstrar como usar esse recurso, preparamos este post com os seguintes tópicos:

## Índice

**01** | O que é a propriedade position?

**02** | Qual é a sua sintaxe?



CSS position: configurando a posição dos elementos na página!

**05** | Exemplo de uso da propriedade position

**06** | Qual é a compatibilidade com os navegadores?

Continue com a gente e boa leitura!

## O que é a propriedade position?

Como mencionamos, a propriedade position define a posição de um elemento na [página HTML](#). Por padrão, seu valor é sempre definido como **static**, ou seja, acompanha o fluxo normal do posicionamento na página. Portanto, se nada for especificado, ele será posicionado conforme a ordem natural dos elementos usados na página.

## Qual é a sua sintaxe?

A sintaxe da propriedade position é:

```
position: static | relative | fixed | absolute | sticky
```

Exceto quando seu valor é definido como **static**, a propriedade **position** deve ser utilizada em conjunto com outras propriedades auxiliares. Como elas são necessárias para indicar o posicionamento do elemento, falaremos mais sobre elas nos próximos tópicos.



## CSS position: configurando a posição dos elementos na página!

A propriedade **position** pode ter diferentes valores, que afetam o comportamento do [elemento na página](#). Confira quais valores ela pode assumir.

### Mantendo a posição padrão do elemento: position: static

O valor da propriedade **CSS position static** determina que o elemento utilizará o posicionamento padrão, ou seja, atribuirá o comportamento natural na página. É importante dizer que, ao utilizar a condição position: static, as propriedades **top**, **bottom**, **left** e **right** atribuídas ao elemento são ignoradas pelo navegador.

```
position: static;
```

### Alterando a posição do elemento em relação à sua posição normal: position: relative

A propriedade **CSS position relative** determina que o elemento será posicionado de acordo com o fluxo natural da página. Entretanto, ao utilizá-lo em conjunto com as propriedades auxiliares, o posicionamento será ajustado de acordo com as indicações. Por exemplo, se definirmos **left: 20px**, o elemento será afastado 20 pixels à esquerda a partir de seu ponto de origem.

```
position: relative;  
left: 20px;
```

### Fixando a posição de elementos em uma janela de visualização: position:



## CSS position: configurando a posição dos elementos na página!

tela.

```
position: fixed;  
top:0;
```

## Posicionando um elemento em relação ao elemento pai: position: absolute

O valor da propriedade **CSS position absolute** indica que o elemento ficará em uma posição fixa em relação à página.

```
position: absolute;  
bottom:0;
```

## A posição do elemento é relativa até que ultrapasse os limites do bloco: sticky

A propriedade **position sticky CSS** tem o comportamento variado. No primeiro momento, ela funciona com a posição relativa até atingir determinado limite, que pode ser a altura da tela ou o tamanho de um container, caso ele esteja inserido em uma <div>, por exemplo.

Ao fazer a rolagem da página e atingir o limite do bloco, o elemento definido como **sticky** terá o comportamento como **fixed** e respeitará os valores definidos pelas propriedades auxiliares para indicar a posição. Essa propriedade é utilizada, por exemplo, para manter o header fixo no topo ao rolar a página.

```
position: sticky;
```



CSS position: configurando a posição dos elementos na página!

## padrao: initial

Definir o valor de position para **initial** é o mesmo que dizer ao navegador para utilizar o valor padrão definido para aquele elemento, que é o posicionamento **static**.

```
position: initial;
```

## Herdando a propriedade de seu elemento pai: inherit

Para herdar as características de posicionamento do elemento pai, basta definir o valor de position igual a **inherit**.

```
position: inherit;
```

### Newsletter da Trybe

Junte-se a mais de  
100.000 pessoas da nossa  
tribo que recebem  
conteúdos gratuitos e  
exclusivos em nossa  
newsletter semanal!

Quero me inscrever!



## CSS position: configurando a posição dos elementos na página!

Existem outras propriedades que ajudam a posicionar o elemento na posição e ordem desejada. Confira como elas funcionam.

### Bottom

Posiciona o elemento conforme a distância indicada e utiliza a área inferior da tela como ponto de início.

### Top

Posiciona o elemento a partir do topo da página.

### Left

Posiciona o elemento a partir do lado esquerdo da página.

### Right

Determina a posição do elemento a partir do lado direito da página.

### Clip

Define a área em que o conteúdo de um elemento será visualizado. Essa propriedade só pode ser usada em conjunto com o posicionamento **absolute** ou **fixed**.

```
.elemento{
  position: absolute;
  clip: rect(0px,100px,200px,0px);
}
```



## CSS position: configurando a posição dos elementos na página!

possíveis são:

- **scroll**: para adicionar barras de rolagens na vertical e horizontal, mesmo que uma delas fique desabilitada;
- **hidden**: para esconder o conteúdo em excesso;
- **auto**: adiciona barras de rolagem de acordo com a necessidade, ou seja, se o excesso de conteúdo for na direção horizontal, apenas essa barra será adicionada;
- **visible**: valor padrão para a propriedade e exibe o conteúdo fora dos limites do elemento.

## Z-index

A propriedade **z-index** é utilizada para ordenar os elementos na página, especialmente quando eles se sobrepõem. Na prática, funciona como um indicador de camadas, no qual o valor maior corresponde à posição superior.

É importante dizer que o z-index só funciona em conjunto com as propriedades **absolute**, **relative** ou **sticky**.

```
.elemento {  
    position: absolute;  
    top: 0px;  
    z-index: -1;  
}
```

## Exemplo de uso da propriedade position

No [código de exemplo](#) abaixo, definimos as duas primeiras divs verdes com a propriedade **position** igual **static** para demonstrar como os elementos se dispõem naturalmente na tela. Perceba que declaramos dois elementos com a cor lilás e com a position igual a



## CSS position: configurando a posição dos elementos na página!

tem o posicionamento como **sticky**. Perceba que, ao rolar a tela, ele ficará fixo na posição indicada. Observe, também, que o elemento de cor vermelha passa por trás do elemento **sticky**. Isso acontece porque definimos seu **z-index** como "-1". Caso contrário, ele passaria por cima da <div> com o **sticky**.

Por fim, definimos um elemento de cor azul e atribuímos a propriedade position igual a **absolute**, com distâncias a partir do topo e da parte esquerda da tela. Perceba que ele permanece nessa posição mesmo se rolarmos a página. Confira o [código](#) a seguir.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>CSS - Position</title>
  <style>
    .box{
      width: 150px;
      height: 150px;
      border: 2px solid black;
      background-color: rgb(195, 241, 195);
    }
    .positionRelative{
      position:relative;
      left:20px;
      background-color: rgb(245, 149, 149);
      z-index: -1;
    }
    .positionFixedTop{
      position:fixed;
      top:0;
      right:20px;
      background-color:rgb(208, 167, 246);
    }
    .positionFixedBottom{
      position:fixed;
      bottom:0;
```





## CSS position: configurando a posição dos elementos na página!

```
        top:250px;
        left:200px;
        background-color:rgb(167, 192, 246);
    }
    .positionSticky{
        position: sticky;
        top:5px;
        background-color:rgb(246, 221, 167);
    }
</style>
</head>
<body>
    <div class="box">Posicionamento padrão: static</div>
    <div class="box">Posicionamento padrão: static</div>
    <div class="box positionSticky">Posicionamento Sticky </div>
    <div class="box positionRelative">Posicionamento relative com
left = 20px</div>
    <div class="box positionFixedTop">Posicionamento fixed - top</
div>
    <div class="box positionFixedBottom">Posicionamento fixed - bo
ttom</div>
    <div class="box positionAbsolute">Posicionamento absolute</div
>
    <div class="box">Posicionamento padrão: static</div>
    <div class="box">Posicionamento padrão: static</div>
</body>
</html>
```

## Qual é a compatibilidade com os navegadores?

Antes de utilizar um recurso em uma [aplicação web](#), é importante observar qual a sua compatibilidade em relação aos principais navegadores disponíveis no mercado. Se uma funcionalidade não for compatível, poderá [ocasionar falhas](#) durante a [navegação pela pessoa usuária](#). Por isso, confira a partir de qual versão a propriedade **position** está disponível nos principais navegadores:

- Internet Explorer: 6;



## CSS position: configurando a posição dos elementos na página!

- iOS Safari: 3.2;
- Android Browser: 81;
- Opera Mobile: 59;
- Chrome for Android: 88;
- Firefox for Android: 85;
- Samsung Internet: 4.

A propriedade **CSS position** é usada para ajudar a distribuir os elementos pela página de forma organizada e a criar efeitos especiais, como fixar o cabeçalho na parte superior. Para isso, conta com o auxílio de propriedades adicionais que indicam com mais precisão em que ponto da página cada elemento deve ficar.

Gostou do nosso conteúdo sobre CSS position? Então, confira agora [nosso tutorial completo sobre HTML com tudo o que você precisa saber sobre o assunto!](#)



Michelle Horn

### DEIXE UM COMENTÁRIO

Você precisa fazer o [login](#) para publicar um comentário.

### **CSS: o que é, guia sobre como usar e vantagens!**

CSS Border

CSS Color

CSS Background-image

CSS Display

CSS Gradient Linear e Radial



## CSS position: configurando a posição dos elementos na página!



# Insights para te ajudar na carreira em tecnologia | Blog da Trybe

[Currículo](#)

[Dúvidas](#)

[Trabalhe Conosco](#)

[Gerador de CPF](#)

[Pague só quando trabalhar](#)

[Guia HTML](#)

[Guia Javascript](#)

[Guia Soft skills](#)

[Carreira](#)

[Tecnologia](#)

[Desenvolvimento Web](#)

[Linguagens de Programação](#)

[Framework de Programação](#)

[TXN](#)

[Ferramentas](#)

[Desabilitar cookies](#)

[Política de Privacidade](#)