

```
<!DOCTYPE html>
<html lang="pt-br">
<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>Position Detalhes</title>
  <link rel="stylesheet" href="004-style.css">
</head>
<body>
```

## `<!--` Position

Esta propriedade define como um elemento deve ser posicionado na página.

Por padrão todos os elementos são posicionados estaticamente (static) o que faz com que o espaço seja ocupado apenas por ele mesmo. Se colocarmos um segundo elemento junto a um elemento que está estático, ele vai ficar do lado ou embaixo dependendo o display do elemento estático, mas nunca vai poder ser aplicado em camada, ficando acima ou abaixo de um elemento.

Além de static, temos: fixed, absolute, relative e sticky como valores possíveis para posicionamento.

### Static

No position "static" o elemento ficará posicionado em relação ao lugar onde nós o declaramos no HTML, e ele irá ocupar todo o espaço dependendo se o seu display é "inline" ou "block".

Dos 5 valores do position, o Static é o único que NÃO SERÁ AFETADO pelas propriedades de deslocamento top, right, bottom e left.

### Fixed

Esta posição faz com o elemento fique posicionado fixamente em relação a janela de visualização, como se ele estivesse congelado na janela. O que fará com que não tenha sua posição afetada mesmo quando o scroll da página for rolado.

### Absolute

Esta posição faz com o elemento fique posicionado fixamente em relação ao ancestral posicionado mais próximo (Ou seja, ele vai ficar fixo ao elemento ancestral mais próximo que seja static, ou seja, que foi possível movê-lo na tela para diferentes direções). Caso não existe nenhum elemento ancestral posicionado ele usará o body como relação, ele NUNCA IRÁ USAR A JANELA COMO PONTO DE REFERÊNCIA. Isso fará com que ele acompanhe o scroll quando a página for rolada. Esse valor é excelente para posicionar um elemento dentro de outro elemento, como um botão dentro de uma div, ou uma imagem dentro de uma div e etc.

### Relative

Esta posição faz com o elemento fique posicionado em relação a sua própria posição normal. Pode ser usado para deslocar o elemento em relação a ele mesmo e também posicionar a fim de afetar elementos descendentes.

### Sticky

Esta posição faz com o elemento se comporte como relative e fixed em determinados momentos. Ele se comporta como relative em relação ao elemento pai onde ele está atribuído, mas quando a janela rola além da posição do elemento pai, o elemento com sticky fica fixo dentro do elemento pai. É como se ele fosse um ímã, ele fica colado numa posição, enquanto o resto da página rola.

```
-->
<div id="posicao">

    <button type="button" class="pulse">Hcode</button>

</div>

</body>
</html>
```

## CSS

```
/*Posicionamento de Botão*/
/*Aqui testamos o posicionamento do nosso botão dentro dos mais variados tipos de position*/

body {
    background-color: #17181c;
```

```

color: #fff;
font-family: sans-serif;
margin: 0; /*Tiramos as margens para que todos os elementos fiquem encostados*/
height: 200vh; /*Dobramos o tamanho do body para que ele ocupe o dobro do tamanho da janela do navegador*/
background: url(004-boku.png); /*Imagem para perceber que a página está andando*/
}

/* Só descomente aqui para ver os valores: absolute e sticky; */
/* #posicao {
    position: relative;
    width: 500px;
    height: 200px;
    background-color: #fff;
    left: 100px;
    top: 100px;
} */

.pulse {
    background: none;
    color: #ff760c;
    border: 2px solid;
    margin: 0; /*Tiramos as margens do nosso botão também para que ele fique completamente encostado para facilitar a visualização da
s posições*/
    padding: 15px 40px;
    transition: 0.25s;
    top: 100px;
    left: 300px; Ex:1 - veja que não teve efeito nenhum, por que, por padrão o elemento é static e não sofre alterações por top, bott
om, right ou left;
    /* position: relative; Ex:2 - Veja que quando colocamos ele retative, ele vai poder deslocar, mas como o próprio nome do atributo
diz, só "relativamente" ao elemento pai onde ele está, no caso do botão aqui, ele está deslocando relativamente em relação a body; */
    /* position: fixed; Ex:3 - Veja que o botão ficou fixo em relação a janela, assim como o relative, os elementos que tiverem posit
ion fixed são afetados pelo top, bottom, left e right; */
    /* position: absolute; Ex:4 - Descomente a formatação do id "posicao"! Percebemos agora que, o botão amarrado a uma div ancestral
, vai obedecer o marco 0 da parte superior do elemento ancestral, ficando fixo somente em relação ao ancestral e não em relação aos e
lementos anteriores ou a página. */

```

```
/* position: sticky; Ex:5 - Descomente a formatação do id "posicao"! Perceba que o botão fica fixo dentro da sua div, mas a medida que vamos rolando a página ele fica parado na div enquanto a página vai rolando até a div desaparecer. */
}

.pulse:hover, .pulse:focus {
  color: #fff;
  border-color: #ef8f6e;
  box-shadow: 0 0 0 15px rgba(255, 255, 255, 0);
  animation: pulsar 1s;
}

@keyframes pulsar {
  0% {
    box-shadow: 0 0 0 0 #ef8e6e;
  }
}
```