CSS padding: a distância entre o elemento e as bordas da página!

Última atualização 24 de junho de 2021

O **CSS padding faz parte do conjunto de propriedades da linguagem de estilo CSS utilizada para definir o espaçamento ao redor de um elemento**. É importante entender como essa propriedade funciona e quando utilizá-la, pois ela é capaz de desposicionar elementos e fazer com que a pessoa desenvolvedora de aplicações front-end passe uma boa parte do tempo em busca de soluções.

Além disso, existe uma certa confusão entre a funcionalidade do padding em comparação à propriedade margin. Para demonstrar como esse recurso funciona e o que é possível fazer com ele, preparamos esse conteúdo que contém os seguintes tópicos:

Índice

Continue com a gente e boa leitura!

Saiba o que é a propriedade CSS padding

Basicamente, a propriedade **padding** corresponde à distância interna entre um elemento e a sua borda. Na prática, cada elemento HTML contém uma largura e altura padrão, que pode ou não ocupar toda a largura da página independente se o conteúdo também ocupa esse mesmo espaço.

Os elementos que ocupam toda a linha são chamados elementos de bloco ou ***block***, entre eles: <h1>, , <p>, entre outros. Já os elementos em linha ou ***inline***, são aqueles que ocupam apenas o espaço correspondente ao seu conteúdo, como: , <a>, <img> etc. Veja como a propriedade **padding** funciona quando é aplicada nesses dois tipos de elementos:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>CSS - Padding</title>
```

```
adding</h1>
    <h1 class="bordaVerde">Elemento block sem padding</h1>
    <a href="#" class="bordaVermelha paddingElemento">Elemento inl
ine com padding</a>
    <a href="#" class="bordaVerde">Elemento inline sem padding</a>

    <style>
        .bordaVermelha{
            border:5px solid red;
        }
        .bordaVerde{
            border:5px solid green;
        }
        .paddingElemento{
            padding: 20px;
        }
    </style>
</body>
</html>
```

Perceba que o elemento <h1> ocupa toda a largura da página, mesmo se escrevermos apenas poucas palavras entre as tags de abertura e fechamento. Nesse caso, o **padding** é aplicado em toda a linha. Já no elemento <a>, que não ocupa a linha inteira, o distanciamento é aplicado apenas ao redor do espaço ocupado pelo conteúdo.

Confira a sintaxe da propriedade CSS padding

A propriedade **padding** pode conter diferentes valores. Veja qual é a sua sintaxe:

padding: height / inherit / initial.

No qual:

- **height**: corresponde a um tamanho predefinido, que pode ser especificado em unidades de medidas absolutas ou relativas.

- **Initial**: corresponde ao valor padrão do elemento.

Entenda a diferença entre CSS padding e CSS margin

É comum haver uma certa confusão entre as propriedades CSS **padding** e **margin**, pois as duas adicionam distâncias ao elemento. Entretanto, a **margin** corresponde ao espaçamento externo, pois ela é aplicada ao redor da borda. Já na propriedade **padding**, como dissemos, a distância é aplicada no interior do elemento, antes da borda. Veja o mesmo exemplo anterior se aplicarmos uma margin de 20px:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>CSS - Padding</title>
</head>
<body>
    <h1 class="bordaVermelha paddingElemento">Elemento block com p
adding e sem margin</h1>
    <h1 class="bordaVerde">Elemento block sem padding e sem margin
</h1>
    <h1 class="bordaAzul paddingElemento marginElemento">Elemento
block com padding e margin</h1>
    <h1> Elemento block sem nenhum estilo</h1>
    <a href="#" class="bordaVermelha paddingElemento">Elemento inl
ine com padding e sem margin</a>
    <a href="#" class="bordaVerde">Elemento inline sem padding e s
em margin</a>
    <a href="#" class="bordaAzul paddingElemento marginElemento">E
lemento inline com padding e margin</a>
    <a href="#">Elemento inline sem estilo</a>
    <style>
        .bordaVermelha{
            border:5px solid red;
        }
        .bordaAzul{
            border:5px solid blue;
```

```
        }
        .paddingElemento{
            padding: 20px;
        }
        .marginElemento{
            margin: 20px;
        }
    </style>
</body>
</html>
```

Perceba que nos elementos com a borda em azul existe uma distância adicional de 20 px antes do início e após o final da borda. Além disso, colocamos as formatações do exemplo anterior para demonstrar a diferença entre elas.

É importante dizer que, **nos elementos *inline,* as propriedades margin e padding são ignoradas para as posições top e bottom**. Para que essa condição se modifique e elas tenham o comportamento de um elemento *block*, é preciso utilizar a propriedade **display** e atribuir o seu valor igual a **block**.

Outra observação importante nesse exemplo é em relação à largura do elemento quando utilizamos a propriedade **padding**. Compare o elemento *inline* com a borda vermelha ao mesmo elemento com a borda verde e veja como houve um acréscimo na largura no primeiro caso.

Quando utilizamos o padding, esse acréscimo pode causar o desposicionamento dos elementos HTML em uma página. Uma forma de evitar esse problema é descontando o espaço atribuído ao **padding** na propriedade **width** do elemento. Dessa forma, podemos evitar a quebra do layout.

Veja quais são os valores da propriedade CSS padding!

Como mencionamos, os valores de **padding** podem ser diferenciados. Veja abaixo como ele funciona para cada possibilidade.

Definindo o tamanho do preenchimento com um valor fixo: length

A primeira possibilidade é a definição de um valor fixo para o **padding**. Utilizamos essa condição nos exemplos anteriores ao definirmos 20 pixels na propriedade. Vale ressaltar que quando usamos um valor fixo, essa condição será respeitada seja qual for o tamanho da tela do dispositivo. Dessa forma, o resultado poderá nem sempre é harmônico em todos os cenários.

É importante dizer que os valores fixos são considerados absolutos por não sofrerem variações. Eles podem ser representados por diferentes unidades de medida como: pixels (px), points (pt), centímetros (cm), in (polegadas) etc.

Definindo o tamanho do preenchimento relativo ao tamanho do bloco: percentage(%)

navegador fará o cálculo de acordo com o espaço disponível e o percentual indicado para a propriedade. Essa alternativa faz com que **o espaçamento seja proporcionalmente o mesmo em cada tamanho de tela**. Veja o exemplo no código fonte abaixo:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>CSS - Padding</title>
</head>
<body>
    <h1 style="padding:10%; border:5px solid red">Elemento com pad
ding em percentual - redimensione a tela para perceber o funcionam
ento!</h1>
</body>
</html>
```

Além do percentual (%), existem outras unidades de medidas que também geram valores relativos para o padding, entre elas:

- **em**: equivale a duas vezes o tamanho da fonte utilizada no elemento;
- **rem**: o valor é calculado com base no tamanho da fonte do elemento root, que corresponde ao elemento raiz da página HTML;
- **vw** e **vh**: relativos à largura (width) e altura (height) da área visível da tela (viewport).

Redefinindo a propriedade para o valor padrão: initial

A utilização do valor **initial** ao **padding** significa que ele terá a condição padrão definida pelo navegador. Mostraremos um exemplo no próximo tópico ao demonstrar a diferença entre **initial** e o valor igual a **inherit**.

elemento pai: inherit

O valor **inherit** para o **padding** indica que o atual elemento herdará a característica atribuída ao elemento pai. Veja um código de exemplo com as duas possibilidades para o **padding: initial e inherit**.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>CSS - Padding</title>
</head>
<body>
    <div class="bordaVermelha paddingValor">
        <p class="bordaVerde paddingInherit">
            Parágrafo com o padding inherit!
        </p>
        <p class="bordaAzul paddingInitial">
            Parágrafo com o padding initial!
        </p>
    </div>
    <style>
        .bordaVermelha{
            border:5px solid red;
        }
        .bordaAzul{
            border:5px solid blue;
        }
        .bordaVerde{
            border:5px solid green;
        }
        .paddingValor{
            padding: 20px;
        }
        .paddingInherit{
            padding: inherit;
        }
        .paddingInitial{
            padding: initial;
        }
    </style>
```

Perceba que definimos o valor de **padding** igual a 20px no elemento . Ele contém dois parágrafos, que são considerados elementos filhos. No primeiro parágrafo definimos a propriedade **padding** como **inherit**, o que significa que ele herda o valor igual a 20px. Portanto, o conteúdo está com essa distância da borda verde.

Já no segundo elemento, definimos o **padding** como **initial**. Portanto, o valor atribuído ao espaçamento é igual a zero. Desse modo, o início do conteúdo está posicionado imediatamente após a borda azul, sem nenhum distanciamento.

Conheça todas as propriedades padding CSS

Quando dizemos que a propriedade **padding** é igual a 20px, por exemplo, significa que os quatro lados ao redor do elemento terão o mesmo valor. Entretanto, existe a possibilidade de atribuirmos valores diferentes a cada uma dessas posições. Para isso, utilizamos a especificação de valor para cada lado do box representado pelo elemento. Veja a seguir:

- **padding-bottom**: indica a distância na parte inferior do conteúdo;
- **padding-left**: representa a distância no lado esquerdo do box;
- **padding-right**: indica a distância aplicada no lado direito do elemento;
- **padding-top**: indica qual o espaçamento superior no elemento.

É importante dizer que, ao escrever o código CSS, podemos utilizar as especificações do posicionamento conforme a nomenclatura acima. Ou seja, definir os valores de forma individual. Veja abaixo:

```
padding-bottom: 20px;
padding-left: 10px;
```

Outra opção é escrevermos o valor individual para cada posição conforme a sintaxe abaixo:

```
pading: padding-top, padding-right, padding-bottom, padding-left.
```

Exemplos:

```
padding: 0 20px 0 20px; /* apenas os lados direito e esquerdo
terão valores iguais a 20px. */
padding: 15px 10px;  /* valores definidos para o padding-top e
padding-right. */
```

Confira qual a compatibilidade com os navegadores

Quando utilizamos um recurso em uma aplicação web, é importante verificarmos o seu comportamento nos principais navegadores. Isso porque se não houver compatibilidade, a pessoa usuária da página poderá sofrer falhas durante a utilização. Confira a partir de qual versão os principais navegadores suportam a propriedade **padding**:

- Internet Explorer: 6;
- Edge: 12;
- Firefox: 2;
- Chrome: 4;
- Safari: 3.1;
- Opera: 10;
- iOS Safari: 3.2;
- Android Browser: 81;
- Opera Mobile: 59;
- Chrome for Android: 88;
- Firefox for Android: 83;
- Samsung Internet: 4.

importante entender como ela funciona em uma página HTML para que sua aplicação não prejudique o posicionamento dos elementos no layout da página.

Gostou do nosso conteúdo sobre como funciona a propriedade CSS padding? Então, confira este post sobre usabilidade e confira o que é e sua aplicação em interfaces!

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 Hover
CSS Margin
CSS Padding
CSS Position
CSS Transition

Trabalhe Conosco
Gerador de CPF
Pague só quando trabalhar
Guia HTML
Guia Javascript
Guia Soft skills
Carreira
Linguagens de Programação
Framework de Programação
TXN
Ferramentas
Desabilitar cookies
Política de Privacidade