



CSS color: como usar cores em CSS? Tutorial completo!

Última atualização 24 de junho de 2021

A propriedade **CSS color** é utilizada para definir a cor do texto em um determinado elemento ou para toda a página HTML. Entretanto, as cores também são aplicáveis em outras [propriedades CSS](#), como a **background-color**, **border-color**, entre outras.

Existem diferentes sistemas de cores que podem ser utilizados, como o RGB, RGBA, HSL e muito mais. Essa variedade é importante, pois permite a [criação de efeitos interessantes](#), como a aplicação de gradientes.

As pessoas que desenvolvem [aplicações front-end](#) precisam conhecer as características dessa propriedade para explorar ao máximo as possibilidades que ela oferece. Por isso, preparamos este conteúdo que contém os seguintes tópicos:

Índice

01 | O que é a propriedade color?

02 | Qual é a sua sintaxe?

03 | Valores: tipos e como usar!

04 | CSS Border Color: configurando a cor das bordas!

05 | CSS Text Color: configurando a cor dos textos!

06 | CSS Background Color: configurando a cor de fundo!

07 | CSS transparent: usando cores transparentes em CSS!

08 | CSS currentcolor: permitindo a herança de propriedades de cor!

09 | CSS Color names: como usar o nome das cores em CSS?

010 | CSS Color Schemes: conhecendo os esquemas de cores!

11 | Qual é a compatibilidade com os navegadores?

Continue com a gente e boa leitura!

O que é a propriedade color?

Basicamente, a propriedade **color** é usada para a definição da cor do texto em uma [página HTML](#). Entretanto, ao ser associada a outras propriedades, ela servirá para atribuir cores a elas. Por exemplo: **background-color**, **border-color**, **text-decoration-color**, **text-shadow** e muitas outras.

Qual é a sua sintaxe?

A sintaxe da propriedade **color** é bem simples, pois conta com apenas um atributo:

```
color: value | initial | inherit
```

Valores da propriedade

A propriedade **color** tem três possibilidades de valores. Confira cada um deles.

Value

Atribui um valor para a propriedade color e que pode ser definido por diferentes tipos, entre eles:

- valores em hexadecimal;
- hexadecimal com transparência;
- RGB;
- RGBA;
- HSL;
- HSLA;
- nomes das cores.

Falaremos sobre cada um desses modelos nos próximos tópicos.

Initial

Ao definirmos a propriedade color como **initial**, dizemos ao navegador que o elemento receberá a cor padrão para o elemento.

```
color: initial;
```

Inherit

O valor da propriedade color igual a **inherit** indica que ela herdará essa característica do elemento pai.

```
color: inherit;
```

Valores: tipos e como usar!

Como mencionamos, a propriedade **color** pode ser definida por meio de diferentes tipos de sistemas de cores. Confira como cada um deles funciona.

Como usar o valor HEX em CSS?

Os valores das cores em hexadecimal são definidos pela combinação de três cores e escrito em formato de **CSS color codes** com o símbolo # no primeiro caractere. Veja a estrutura:

```
#RRGGBB
```

No qual:

- **RR** corresponde à cor vermelha (red);
- **GG** é referente à cor verde (green);
- **BB** corresponde à cor azul (blue).

Vale ressaltar que a intensidade de cada cor pode variar entre 00 e FF. Por exemplo:

```
color: #00B718; /* corresponde a uma tonalidade verde */  
color: #B51800; /* corresponde a uma tonalidade vermelha */  
color: #1A00AF; /* equivale a um tom de azul */
```

Como usar o valor HTML em CSS?

Também podemos definir o valor da propriedade com a atribuição de CSS color names. Na prática, a [linguagem HTML](#) oferece [suporte a 140 cores](#) que são identificadas por todos os navegadores modernos. Veja alguns exemplos:

```
color: AntiqueWhite; /* corresponde à cor #FAEBD7 */  
color: DarkGreen; /* equivale à #006400 */
```

Como usar o valor RGB em CSS?

As cores **RGB** são definidas por meio de uma função CSS chamada **rgb()**, que utiliza valores correspondentes à intensidade das cores vermelha (red), verde (green) e azul (blue). Sua sintaxe é:

```
rgb(red, green, blue)
```

No qual:

- **red**: equivale à intensidade da cor vermelha e seu valor pode variar entre 0 e 255 ou em percentuais de 0% a 100%;
- **green**: corresponde à intensidade da cor verde e segue o mesmo critério para determinar o tom;
- **blue**: define a intensidade da cor azul.

Veja um exemplo:

```
color: rgb(15, 96, 22);
```

Como usar o valor HSL em CSS?

Outra forma de definir as cores é por meio da função CSS **hsl()**, que utiliza um sistema de coordenadas cilíndricas para determinar a cor. A função utiliza três parâmetros, são eles:

- **hue**: que significa matiz e identifica a cor conforme o grau da roda de cores que pode variar de 0° a 360°;
- **saturation**: equivale à saturação da cor e é representado por um valor percentual, no qual 0% corresponde à tonalidade cinza e 100% equivale à cor total;
- **lightness**: significa à luminosidade e também é representado por um valor percentual, em que 0% equivale à cor preta e 100% equivale à cor branca.

Veja um exemplo:

```
color: hsl(125, 84%, 38%);
```

Como usar o valor HWB em CSS?

O valor HWB — *Hue Whiteness Blackness* — corresponde a uma função CSS chamada **hwb()**. Na prática, ela recebe uma cor e adiciona quantidades determinadas de tonalidades branca e preta. Por isso, sua utilização é mais intuitiva, pois ela funciona como uma mistura de cores. Sua sintaxe é:

```
hwb(matiz, branco, preto)
```

Em que:

- **matiz**: corresponde à cor desejada e deve ser informado com base no círculo de cores;
- **branco**: percentual de cor branca adicionada à cor;
- **preto**: percentual de cor preta adicionada à cor.

Veja um exemplo:

```
color: hwb( 125, 0%, 25%);
```

Vale ressaltar que a função **hwb()** é uma sugestão para a nova versão da [linguagem de estilos](https://blog.betrybe.com/css/css-color/), a CSS4. Portanto, atualmente **esse**

recurso não está disponível nos principais navegadores.

CSS Border Color: configurando a cor das bordas!

A propriedade **color** pode ser associada à [propriedade CSS border](#) para definir qual a cor para a borda de um elemento. Podemos determinar essa característica de duas formas específicas. A primeira é por meio da propriedade **border**, em que informamos todas as características da borda, inclusive a cor.

```
border: 5px solid green;
```

Outra forma é com a propriedade **CSS border-color**, que utilizamos da mesma forma que a **color**. Veja um exemplo:

```
border-style: solid;  
border-color: #0F6016;
```

Vale ressaltar que precisamos definir a propriedade **border-style** quando vamos utilizar a **border-color** para que ela funcione da maneira desejada. Além disso, podemos definir as cores de acordo com os sistemas que demonstramos ao longo do texto.

CSS Text Color: configurando a cor dos textos!

As cores dos textos são definidas pela propriedade **color**. Ela pode ser declarada para atender todos os elementos da página [HTML](#) ou para ser aplicada em um elemento específico. Veja um [código de exemplo](#) sobre a definição de **CSS font color**:

```
<!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 Font Color</title>
<style>
  html{
    color: green;
  }
  p{
    font-size: large;
    color: #BC277E;
  }
  .color{
    color:rgb(196, 181, 43);
  }
</style>
</head>
<body>
  <h2>A cor verde foi definida para todos os elementos HTML.
</h2>
  <p>Exceto para os parágrafos.</p>
  <h2 class="color">Ou para elementos que utilizem a classe colo
r.</h2>
  <p>Outro parágrafo.</p>
  <h3>Um título h3 com a cor padrão para a página.</h3>
</body>
</html>
```

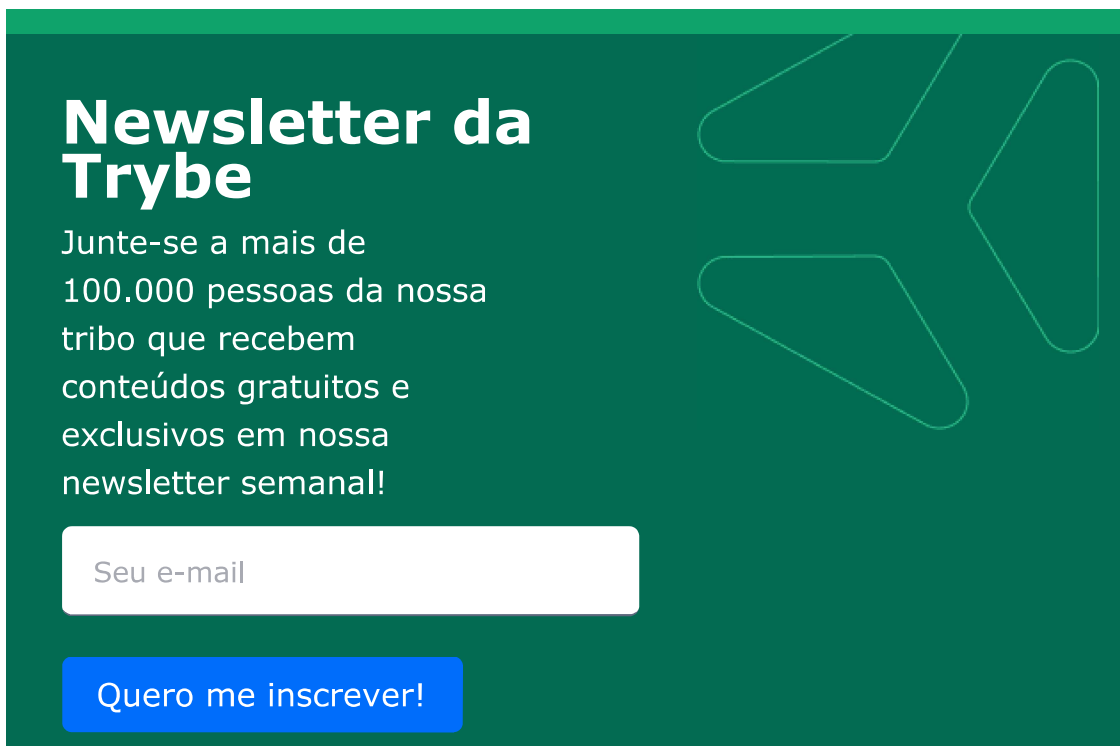
CSS Background Color: configurando a cor de fundo!

A propriedade **CSS background-color** também funciona de forma semelhante à **color**. Basta informarmos um dos sistemas de cores para definirmos a opção que será exibida como fundo em um elemento HTML. Veja em exemplo:

```
background-color: rgb(125, 20, 20);
background-color: green;
```

O sistema de cores também é utilizado para definir o CSS background color gradiente, que nada mais é que uma [imagem](#) de

fundo feita com a aplicação de duas ou mais cores com o efeito de transição entre elas. Falamos um pouco mais sobre a propriedade `background-image` [neste conteúdo](#).



CSS transparent: usando cores transparentes em CSS!

Também podemos definir níveis de transparência ao aplicar as cores em elementos HTML. Isso pode ser feito de diferentes formas, uma delas é com a utilização das funções CSS **`rgba()`** e **`hsla()`**. Na prática, a função **`rgba()`** funciona de forma semelhante à função **`rgb()`**, entretanto, contém um parâmetro adicional para determinar a intensidade da transparência.

A função **`hsla()`** segue o mesmo princípio e o seu último parâmetro também corresponde à opacidade, no qual o valor varia de 0 a 1, em que 0 corresponde à transparência máxima, enquanto 1 equivale a nenhuma transparência.

Também podemos aplicar a transparência em cores com o sistema hexadecimal. Para isso, adicionamos dois dígitos ao final da definição da cor para indicar o nível de transparência, que varia de 00 a FF. Veja um exemplo 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 Color</title>
  <style>
    .container{
      display:flex;
      flex-direction: row;
      flex-wrap: wrap;
    }
    .box{
      width:200px;
      height: 200px;
      margin:10px;
    }
    .background-rgb{
      background-color: rgb(86, 121, 80);
    }
    .background-rgba{
      background-color: rgba(86,121,80,0.74);
    }
    .background-hsl{
      background-color: hsl(111, 20%, 39%);
    }
    .background-hsla{
      background-color: hsla(111, 20%, 39%, 0.74);
    }
    .background-hexa{
      background-color: #567950;
    }
    .background-hexa-transparent{
      background-color: #567950AB;
    }
    .background-transparent{
      border:1px solid green;
      background-color: transparent;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="box background-
rgb">Cor de fundo com função rgb()</div>
    <div class="box background-
rgba">Cor de fundo com transparência e função rgba()</div>
```

```
<div class="box background-  
hsl">Cor de fundo com função hsl(</div>  
    <div class="box background-  
rgba">Cor de fundo com transparência e função hsla(</div>  
    <div class="box background-  
hexa">Cor de fundo hexadecimal</div>  
    <div class="box background-hexa-  
transparent">Cor de fundo hexadecimal com transparência</div>  
    <div class="box background-  
transparent">Background transparente</div>  
</div>  
</body>  
</html>
```

No [código fonte](#) acima utilizamos a mesma cor de fundo para todos os elementos e apenas modificamos o sistema de cor utilizado em cada um. Também aplicamos a transparência nas funções que oferecem esse recurso.

Perceba que, no último elemento, utilizamos a palavra-chave **transparent** para indicar o **CSS background-color transparent**. Esse recurso aplica a transparência máxima no elemento. Por isso, há a ausência de cor.

CSS currentcolor: permitindo a herança de propriedades de cor!

Ao definirmos o conjunto de estilos que serão aplicados em um elemento, é comum repetirmos a cor em determinados locais. Podemos, por exemplo, utilizar a mesma cor da fonte para a borda ao redor do elemento.

Entretanto, precisamos de propriedades diferentes para isso. Em vez de escrevermos a cor ou usarmos a função várias vezes, podemos simplesmente utilizar a palavra **currentcolor**. Na prática, ela diz ao navegador para utilizar a cor definida na propriedade **color**. Veja um exemplo:

```
.box{  
  color: green;  
  border: 5px solid currentColor;  
}
```

CSS Color names: como usar o nome das cores em CSS?

Conforme mencionamos, a linguagem de estilo conta com 140 nomes de cores conhecidas pelos navegadores. Portanto, esses nomes podem ser utilizados para indicar a cor desejada em um elemento HTML. Veja alguns exemplos:

```
<h1 style="color:darkcyan">Título com a cor DarkCyan</h1>  
<h2 style="color:cornflowerblue">Esse utiliza a cor CornFlowerBlue  
</h2>  
<h3 style="color:deeppink">Texto com a cor DeepPink</h3>
```

CSS Color Schemes: conhecendo os esquemas de cores!

A utilização de cores em uma página HTML deve ser pensada para que o visual seja harmônico e esteticamente agradável às pessoas usuárias. Uma forma de atingir esse objetivo é utilizar o esquema de cores. Dessa forma, as cores são selecionadas com base em combinações lógicas disponíveis no círculo cromático. Confira alguns desses esquemas:

- **monocromático:** em que é utilizado apenas uma cor, mas com tonalidades variadas;
- **cores análogas:** são cores vizinhas no círculo cromático;
- **cores complementares:** representam as cores que estão em lados opostos do círculo cromático.

Qual é a compatibilidade com os navegadores?

Quando criamos uma [aplicação web](#), é importante considerar se os recursos utilizados na página funcionam de maneira adequada, pois podem ocorrer [falhas](#) que comprometam a [usabilidade](#). Por isso, confira a partir de qual versão a propriedade **color** é suportada nos principais navegadores:

- Internet Explorer: 9;
- Edge: 12;
- Firefox: 3;
- Chrome: 4;
- Safari: 3.1;
- Opera: 10;
- iOS Safari: 3.2;
- Android Browser: 2.1;
- Opera Mobile: 12;
- Chrome for Android: 89;
- Firefox for Android: 86;
- Samsung Internet: 4.

Existem várias formas de utilizar a propriedade **CSS color** com a aplicação de diferentes sistemas de cores, como a definição hexadecimal ou o uso de funções CSS. Além disso, podemos inserir níveis de transparência e modificar a intensidade da cor. Por isso, é importante que as [pessoas que desenvolvem aplicações front-end](#) entendam como esse recurso funciona.

Gostou do nosso conteúdo sobre como usar a propriedade CSS color? Então confira este post sobre [o que faz uma pessoa programadora e como iniciar essa carreira!](#)

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

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