



O que é CSS?

CSS é chamado de linguagem *Cascading Style Sheet* e é usado para estilizar elementos escritos em uma linguagem de marcação como HTML.

O CSS separa o conteúdo da representação visual do site. Pense na decoração da sua página. Utilizando o CSS é possível alterar a cor do texto e do fundo, fonte e espaçamento entre parágrafos. Também pode usar variações de layouts, ajustar imagens para suas respectivas telas e assim por diante.

CSS foi desenvolvido pelo W3C ([World Wide Web Consortium](https://www.w3.org/)) em 1996, por uma razão bem simples. O HTML não foi projetado para ter tags que ajudariam a formatar a página. Você deveria apenas escrever a marcação para o site.

Tags como foram introduzidas na versão 3.2 do HTML e causaram muitos problemas para os desenvolvedores. Como os sites tinham diferentes fontes, cores e estilos, era um processo longo, doloroso e caro para reescrever o código.

Assim, o CSS foi criado pelo W3C para resolver este problema.

A relação entre HTML e CSS é bem forte. Como o HTML é uma linguagem de marcação (o alicerce de um site) e o CSS é focado no estilo (toda a estética de um site), eles andam juntos.

CSS não é tecnicamente uma necessidade, mas provavelmente você não gostaria de olhar para um site que usa apenas HTML, pois isso pareceria completamente abandonado e sem atrativos.

Vantagens do CSS

A diferença entre um site que implementa CSS e outro que não o usa é gigantesca e notável.

Você já deve ter visto um site que não carrega completamente ou tem um plano de fundo branco com texto azul e preto. Isso significa que a parte CSS do site não foi carregada corretamente ou não existe.

Antes de usar CSS, toda a estilização tinha que ser incluída na marcação HTML.

O CSS permite que você estilize tudo em um arquivo diferente, criando assim o estilo separadamente, e, mais tarde, faça integração do arquivo CSS na parte superior da marcação HTML. Isso mantém a marcação HTML limpa e fácil de manter.

Com o CSS você não precisa mais escrever repetidamente como os elementos individuais se parecem. Isso economiza tempo, encurta o código e diminui a chance de erros.

O CSS permite que você tenha vários estilos em uma página HTML, tornando as possibilidades de personalização quase infinitas.

Hoje em dia, isso está se tornando mais uma necessidade do que um simples recurso.

Como CSS Funciona

O CSS é uma ferramenta muito potente que possibilita criar diversas funcionalidades ao invés de usar JavaScript ou outra linguagem mais pesada. Se usado com moderação, CSS pode viabilizar uma ótima experiência ao desenvolvedor e usuários das páginas web.

Com o *Cascading Style Sheets* é possível criar animações complexas, criar efeitos com uso de *parallax*, que faz parecer que a imagem de fundo tem uma profundidade diferente uma das outras, criar sites interativos e também jogos com HTML5 e CSS3.

O CSS usa uma sintaxe simples baseada em inglês com um conjunto de regras que o governam.

O HTML nunca teve a intenção de usar elementos de estilo, apenas a marcação da página. Foi criado para descrever apenas o conteúdo.

Por exemplo:

```
<p>Este é um parágrafo.</p>.
```

Cada declaração inclui um nome de propriedade CSS e um valor, separados por dois pontos.

Uma declaração CSS sempre termina com um ponto-e-vírgula e os blocos de declaração são cercados por chaves.

Exemplo:

Todos os elementos `<p>` serão estilizados e serão coloridos de azul:

```
<style>
p {
color: blue;
}
</style>
```

Em outro exemplo, todos os elementos `<p>` serão centralizados, com tamanho 16px e de cor pink.

```
<style>
p {
text-align: center;
font-size: 16px;
color: pink;
}
</style>
```

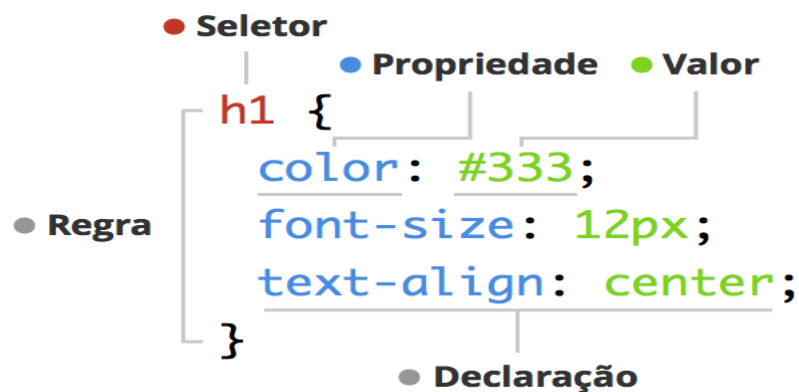
Anatomia de um comando CSS

O CSS estipula regras para o arquivo em html. Com cada regra é possível estilizar o conteúdo todo ou somente determinados elementos.

Por isso entenda, um comando básico é composto por seletor e declarações, que contém propriedade e valor.

SELETOR {PROPRIEDADE: VALOR;}

Onde:



A sintaxe do CSS é muito simples de aprender. O seletor seleciona quais elementos em html receberão a propriedade. Pode ser `p` (parágrafo) ou `body` (corpo da sua página). Já a propriedade pode ser a cor ou algo mais específico como cor do fundo (`background`). E por último o valor, que determina o valor da propriedade.

Comandos básicos CSS

São tantas opções entre seletores, propriedades e valores que pode ser muito difícil para um desenvolvedor lembrar de todos eles apenas confiando na memória, precisa consultar a documentação.

Estilos CSS Interno, Externo e Inline

Diferença entre Estilos de CSS Inline, Interno e Externo

Existem três maneiras que podemos usar para implementar o CSS: interna, externa e inline.

CSS Interno

O CSS interno ou incorporado requer que você adicione a tags `<style>` na seção `<head>` do seu documento HTML.

Este estilo de CSS é um método efetivo de estilizar uma única página. Contudo, usar esse estilo em múltiplas páginas pode consumir muito tempo, já que você precisa definir as regras CSS para cada página do seu site.

Formas de como podemos usar o CSS interno:

1. Abra a sua página HTML e localize a tag de abertura `<head>`.
2. Coloque o seguinte código logo depois da tag `<head>`.

`<style >`

3. Adicione as regras CSS numa nova linha.

Aqui está um exemplo:

```
body {  
  background-color: blue;  
}
```

```
h1 {  
  color: red;  
  padding: 60px;  
}
```

4. Digite a seguinte tag de fechamento:

```
</style>
```

O seu arquivo HTML ficará parecido com isso:

```
<!DOCTYPE html>  
<html>  
  <head>  
    <style>  
      body {  
        background-color: blue;  
      }  
      h1 {  
        color: red;  
        padding: 60px;  
      }  
    </style>  
  </head>  
  <body>  
  
    <h1>Descodificadas</h1>  
    <p>Este é o nosso parágrafo.</p>  
  
  </body>  
</html>
```

Desvantagens de CSS Interno:

- Adicionar o código para o documento HTML pode aumentar o tamanho da página e o tempo de carregamento.

CSS externo

Com o CSS externo, você vai linkar as páginas da internet com um arquivo .css externo, que você pode criar usando qualquer editor de texto no seu dispositivo

Este tipo de CSS é um método mais eficiente, especialmente se você está estilizando um site grande. Ao editar um arquivo .css, você pode modificar um site inteiro de uma só vez.

Na seção <head> da sua planilha HTML, adicione uma referência para o seu arquivo .css logo depois da tag <title>:

```
<link rel="stylesheet" type="text/css" href="style.css" />
```

Não se esqueça de modificar o style.css com o nome do seu arquivo .css caso seja diferente.

Vantagens de CSS Externo

- Como o código CSS está num documento separado, os seus arquivos HTML terão uma estrutura mais limpa e serão menores
- O mesmo arquivo .css pode ser usado em várias páginas.

Desvantagens de CSS Externo

- Até que o CSS externo seja carregado, a página pode não ser processada corretamente.
- Fazer links para múltiplos arquivos CSS pode aumentar o tempo de download do seu site.

CSS Inline

O CSS inline é usado para dar estilo a um elemento HTML específico. Para este estilo de CSS somente é preciso adicionar o atributo `style` para cada tag HTML, sem usar os seletores.

Este tipo de CSS não é realmente recomendado, já que cada tag HTML precisa ser estilizada de maneira individual. Gerenciar o seu site pode se tornar uma tarefa bem difícil de você só usar o CSS inline.

Contudo, o CSS inline no HTML pode ser útil para algumas situações.

Por exemplo, em casos onde você não tem acesso aos arquivos CSS ou precisa aplicar estilos para um único elemento.

Exemplo.

Aqui, foi adicionado um CSS inline para as tags `<p>` e `<h1>`:

```
<!DOCTYPE html>
<html>
<body style="background-color:black;">

<h1 style="color:white; padding:30px;">Descodificadas</h1>
<p style="color:white;">Usado somente aqui.</p>

</body>
</html>
```

Vantagens do CSS Inline

- Podemos inserir elementos CSS de maneira fácil e rápida numa página HTML.
- É por isso que esse método é útil para testar e pré-visualizar mudanças, assim como executar correções rápidas no seu site.

- Não precisamos criar e fazer upload de um documento separado como no estilo externo.

Desvantagens do Inline CSS

- Adicionar regras CSS para cada elemento HTML consome muito tempo e faz a estrutura HTML ficar bagunçada.
- Estilizar múltiplos elementos podem afetar o tamanho da página e o tempo para download.

Tipografia

A tipografia consiste nas fontes de letras que são utilizadas por designers e profissionais de criação para realizar inúmeros tipos de produção de textos e projetos gráficos – para web e impressos.

Famílias de fontes

Se você já usou um software de processamento de texto, como o Microsoft Word ou o Google Docs, provavelmente também usou um recurso que permitia alterar a fonte em que estava digitando. Fonte refere-se ao termo técnico tipo de letra , ou *família* de fontes .

Para alterar o tipo de letra do texto em sua página da Web, você pode usar a font-family.

Exemplo:

```
h1 {  
  font-family: Garamond;  
}
```

Tamanho da fonte

Alterar o tipo de letra não é a única maneira de personalizar o texto. Muitas vezes, diferentes seções de uma página da Web são destacadas ao modificar o tamanho da fonte.

Para alterar o tamanho do texto em sua página da web, você pode usar a propriedade `font-size`.

Exemplo:

```
p {  
  font-size: 18px;  
}
```

Espessura da fonte

Em CSS, a propriedade `font-weight` controla como o texto em negrito ou fino aparecerá.

Exemplo:

```
p {  
  font-weight: bold;  
}
```

Alinhamento de texto

Não importa quanto estilo é aplicado ao texto (tipo de letra, tamanho, peso etc.), o texto sempre aparece no lado esquerdo do contêiner em que reside.

Para alinhar o texto, podemos usar a propriedade `text-align`. A propriedade `text-align` alinhará o texto ao elemento que o contém, também conhecido como `pai`.

Exemplo:

```
h1 {  
  text-align: right;  
}
```

A propriedade `text-align` pode ser definida para um dos seguintes valores comumente usados:

- left— alinha o texto ao lado esquerdo de seu elemento pai, que neste caso é o navegador.
- center— centraliza o texto dentro de seu elemento pai.
- right— alinha o texto ao lado direito de seu elemento pai.
- justify— espaça o texto para alinhar com o lado direito e esquerdo do elemento pai.

Cor e cor de fundo

Antes de discutir as especificidades da cor, é importante fazer duas distinções sobre a cor. A cor pode afetar os seguintes aspectos de design:

- Cor do primeiro plano
- Cor de fundo

A cor de primeiro plano é a cor na qual um elemento aparece. Por exemplo, quando um título é estilizado para parecer verde, a cor de primeiro plano do título foi estilizada. Por outro lado, quando um título é estilizado de modo que seu plano de fundo pareça amarelo, a cor de plano de fundo do título foi estilizada.

Em CSS, esses dois aspectos de design podem ser estilizados com as duas propriedades a seguir:

- color: esta propriedade estiliza a cor de primeiro plano de um elemento
- background-color: esta propriedade estiliza a cor de fundo de um elemento

Exemplo:

```
h1 {  
  color: red;  
  background-color: blue;  
}
```

Opacidade

A opacidade é a medida de quão transparente é um elemento. É medido de 0 a 1, com 1 representando 100%, ou totalmente visível e opaco, e 0 representando 0%, ou totalmente invisível.

A opacidade pode ser usada para fazer os elementos desaparecerem em outros para um bom efeito de sobreposição. Para ajustar a opacidade de um elemento, a sintaxe se parece com isso:

Exemplo:

```
.p {  
  opacity: 0.5;  
}
```

Imagem de fundo

CSS tem a capacidade de alterar o plano de fundo de um elemento. Uma opção é transformar o plano de fundo de um elemento em uma imagem. Isso é feito através da propriedade CSS `background-image`.

Sua sintaxe fica assim:

```
.fundo {  
  background-image: url('https://www.example.com/image.jpg');  
}
```

A propriedade `background-image` definirá o plano de fundo do elemento para exibir uma imagem.

O valor fornecido `background-image` é um url. O url deve ser um URL para uma imagem. O url pode ser um arquivo dentro do seu projeto ou pode ser um link para um site externo. Para vincular a uma imagem dentro de um projeto existente, você deve fornecer um caminho de arquivo relativo . Se houvesse uma pasta de imagens no projeto, com uma imagem chamada `mountains.jpg`, o caminho do arquivo relativo ficaria assim:

Exemplo:

```
.banner {  
  background-image: url('images/mountains.jpg');  
}
```

Introdução as cores

CSS suporta uma ampla variedade de cores. Isso inclui cores nomeadas, como blue, Black e limegreen, juntamente com cores descritas por um valor numérico. O uso de um sistema numérico nos permite aproveitar todo o espectro de cores que os navegadores suportam. Nesta lição, vamos explorar todas as opções de cores que o CSS oferece.

As cores em CSS podem ser descritas de três maneiras diferentes:

- Cores nomeadas — palavras em inglês que descrevem cores, também chamadas de cores de palavras-chave;
- RGB — valores numéricos que descrevem uma mistura de vermelho, verde e azul;
- HSL — valores numéricos que descrevem uma mistura de matiz, saturação e luminosidade.

Hexadecimal

Uma sintaxe que podemos usar para especificar uma cor é chamada hexadecimal. As cores especificadas usando este sistema são chamadas de cores hexadecimais. Uma cor hexadecimal começa com um caractere hash (#) seguido por três ou seis caracteres. Os caracteres representam valores para vermelho, azul e verde.

Exemplos:

```
darkseagreen: #8FBC8F  
sienna: #A0522D  
saddlebrown: #8B4513
```

brown: #A52A2A
black: #000000 or #000
white: #FFFFFF or #FFF
aqua: #00FFFF or #0FF

Cores RGB

Existe outra sintaxe para representar as cores que são valores RGB, comumente chamada de “valor RGB” ou apenas “RGB”, que usa números decimais em vez de números hexadecimais, e se parece com isso:

```
h1 {  
  color: rgb(23, 45, 23);  
}
```

Cada um dos três valores representa um componente de cor e cada um pode ter um valor numérico decimal de 0 a 255. O primeiro número representa a quantidade de vermelho, o segundo é verde e o terceiro é azul. Essas cores são exatamente iguais às hexadecimais, mas com uma sintaxe diferente e um sistema numérico diferente.

Em geral, as representações de cores hexadecimais e RGB são equivalentes. Qual você escolhe é uma questão de gosto pessoal. Dito isso, é bom escolher um e ser consistente em todo o seu CSS, porque é mais fácil comparar hexadecimal com hexadecimal e RGB com RGB.

Matiz, saturação e luminosidade

O esquema de cores RGB é conveniente porque é muito próximo de como os computadores representam as cores internamente. Há outro sistema igualmente poderoso em CSS chamado esquema de cores hue-saturation-lightness, abreviado como HSL .

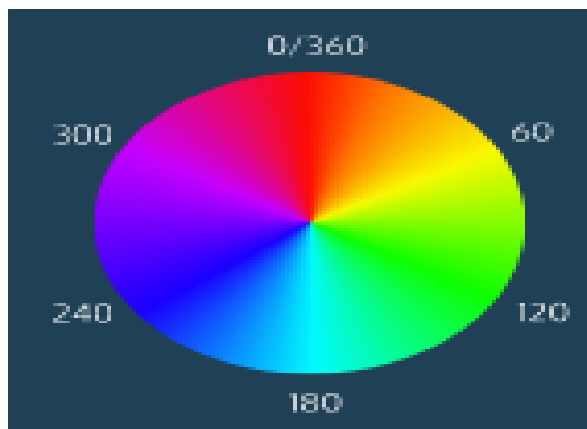
A sintaxe para HSL é semelhante à forma decimal de RGB, embora seja diferente de maneiras importantes. O primeiro número representa o grau de matiz e pode estar entre 0 e 360. O segundo e terceiro números são porcentagens que representam saturação e luminosidade, respectivamente.

Aqui está um exemplo:

color: hsl(120, 60%, 70%);

Matiz é o primeiro número. Refere-se a um ângulo em uma roda de cores. Vermelho é 0 graus, Verde é 120 graus, Azul é 240 graus, e então volta para Vermelho em 360.

Você pode ver um exemplo de uma roda de cores abaixo.



Saturação refere-se à intensidade ou pureza da cor. A saturação aumenta para 100% à medida que a cor se torna mais rica. A saturação diminui para 0% à medida que a cor se torna mais cinzenta.

A luminosidade refere-se a quão clara ou escura é a cor. A meio caminho, ou 50%, é a leveza normal. Imagine um dimmer deslizante em um interruptor de luz que começa no meio do caminho. Deslizar o dimmer até 100% torna a cor mais clara, mais próxima do branco. Deslizar o dimmer para 0% torna a cor mais escura, mais próxima do preto.

HSL é conveniente para ajustar cores. Em RGB, tornar a cor um pouco mais escura pode afetar todos os três componentes de cor. No HSL, isso é tão fácil quanto alterar o valor da luminosidade. O HSL também é útil para criar um conjunto de cores que funcionem bem juntas, selecionando várias cores que têm a mesma luminosidade e saturação, mas tons diferentes.

Unidades absolutas e relativas

Unidades relativas são unidades cujo valor depende de um fator externo, como por exemplo a %(porcentagem) que depende do valor do elemento pai, do vh que varia de

acordo com a altura do viewport, e assim por diante. Unidades absolutas são unidades ancoradas em medidas físicas reais, é o caso do `px` (pixels) e do `cm` (centímetros).

`px`

Um pixel é o conjunto de três pontos minúsculos - um vermelho, um verde e um azul (RGB) - que juntos compõem tudo aquilo que você enxerga na tela. Cada cor desse pixel exibe de 0 a 255 gamas de cores, somando um pouco mais que 16 milhões de possibilidades de cor.

Com o surgimento de diversos tamanhos de dispositivos, cada um com diferentes formatos e taxas de densidade de pixel por polegada, ficou muito difícil manter esse valor absoluto para elementos que precisam se adaptar de acordo com a tela.

Quando usar?

Generalizando, use para elementos e medidas que não irão mudar, independente do contexto ou tamanho de viewport (tela do dispositivo). Um bom exemplo de uso são bordas e alguns box-shadows e para definir valores mínimos e máximos de elementos.

`%`

Porcentagens são quase sempre relativas às medidas do elemento pai delas. Se você declarar uma `width` de 20% em um elemento, a largura dele será 20% da largura do pai, mesma coisa com outras propriedades como `height` e `margin / padding`.

`em` e `rem`

Tem o mesmo conceito de `%` mas ao invés de serem baseadas no tamanho de um elemento, essas medidas são baseadas em tamanho de fonte. **`em`** usa o tamanho da fonte do elemento pai e **`rem`** usa o tamanho da fonte do `<html>`.

Saiba mais

<https://www.w3schools.com/css/>

<https://www.w3.org/Style/Examples/007/units.en.html>

<https://www.devmedia.com.br/css-unidades/41239>

<https://youtu.be/LWU2OR19ZG4>

<https://youtu.be/NGfPXjGiNH8>

<https://youtu.be/229xfk3EEM8>