

O que é CSS?

[CSS](#) (Folhas de Estilo em Cascata) permite a você criar páginas web agradáveis, mas como isso funciona por baixo dos panos? Este artigo explica o que é CSS, com um exemplo de sintaxe simples, e, também, trata alguns conceitos-chaves sobre a linguagem.

A web seria um lugar chato se todos os web sites tivessem estilos iguais ao mostrado na imagem acima. Usando CSS você pode controlar exatamente a aparência dos elementos HTML no navegador, apresentando a sua marcação com o design que desejar.

Para que serve o CSS?

Como falamos antes, CSS é uma linguagem para especificar como documentos são apresentados aos usuários — como eles são estilizados, dispostos etc.

Um **documento** é normalmente um arquivo texto estruturado usando uma linguagem de marcação — [HTML](#) é a linguagem de marcação mais comum, mas você também pode encontrar outras, como [SVG](#) ou [XML](#).

Apresentar um documento para um usuário significa convertê-lo para um formato utilizável pelo seu público. [Browsers](#), como [Firefox](#), [Chrome \(en-US\)](#), ou [Edge \(en-US\)](#), são projetados para apresentar documentos visualmente, por exemplo, em uma tela de computador, projetor ou impressora.

Nota: Observação: Um navegador web é às vezes chamado de [user agent \(en-US\)](#), o que, basicamente, significa um programa de computador que representa uma pessoa por trás do sistema. Navegadores web são o principal tipo de agente do usuário que nos referimos quando falamos sobre CSS, contudo, ele não é o único. Há outros agentes de usuário disponíveis — tais como aqueles que convertem documentos HTML e CSS para PDF a serem impressos.

O CSS pode ser usado para estilizar um documento muito básico de texto — por exemplo, alterando a [cor](#) e [tamanho](#) dos títulos e links. Pode ser usado para criar layout — por exemplo, [transformando uma simples coluna de texto em um layout \(en-US\)](#) com uma área de conteúdo principal e um sidebar (barra lateral) para as informações relacionadas. Pode até ser usado para efeitos tais como [animação](#). Dê uma olhada nos links deste parágrafo, para ver exemplos específicos.

Sintaxe CSS

CSS é uma linguagem baseada em regras. — Você define regras especificando grupos de estilo que devem ser aplicados para elementos particulares ou grupos de elementos na sua página web. Por exemplo, "Quero que o título principal, na minha página, seja mostrado como um texto grande e de cor vermelha."

O código seguinte mostra uma regra CSS muito simples, que chegaria perto do estilo descrito acima:

```
h1 {  
  color: red;  
  font-size: 12px;  
}
```

A regra é aberta com um [selector](#). Isso *seleciona* o elemento HTML que vamos estilizar. Neste caso, estamos estilizando títulos de nível um ([<h1> \(en-US\)](#)).

Temos, então, um conjunto de chaves {}. Dentro deles, haverá uma ou mais **declarações**, que tomam a forma de pares **propriedade e valor**. Cada par especifica uma propriedade do(s) elemento(s) que estamos selecionando e, em seguida, então um valor que gostaríamos de atribuir à propriedade

Antes dos dois pontos, temos a propriedade, e, depois, o valor. CSS [properties](#) possui diferentes valores permitidos, dependendo de qual propriedade está sendo especificado. Em nosso exemplo, temos a propriedade color, que pode tomar vários [valores para cor \(en-US\)](#). Também temos a propriedade font-size. Essa propriedade pode ter vários [unidades de tamanho \(en-US\)](#) como um valor.

Uma folha de estilo CSS conterá muitas regras tais como essa, escrita uma após a outra.

```
h1 {  
  color: red;  
  font-size: 12px;  
}  
  
p {  
  color: black;  
}
```

Você constatará que rapidamente aprende alguns valores, enquanto outros precisará pesquisar. As páginas de propriedades individuais no MDN dá a você uma maneira rápida de procurar propriedades e seus respectivos valores, quando você esquecer, ou desejar saber quais valores pode usar.

Nota: Observação: Você pode achar links para todas as páginas de propriedades CSS (junto com outros recursos CSS) listados no MDN [referência CSS](#). Alternativamente, você deve se acostumar a pesquisar por "mdn css-feature-name" no seu motor de busca favorito sempre que precisar procurar mais informações sobre uma característica CSS. Por exemplo, experimente pesquisar por "mdn color" e "mdn font-size"!

Módulos CSS

Como existem tantas coisas que você pode estilizar com CSS, a linguagem é dividida em módulos.

Especificações CSS

Todas as tecnologias de padrões web (HTML, CSS, JavaScript, etc.) são definidos em documentos gigantes chamados especificações (ou simplesmente "specs"), que são publicados por organizações de padrões (tais como [W3C](#), [WHATWG](#), [ECMA](#), ou [Khronos \(en-US\)](#)) e definem precisamente como essas tecnologias devem se comportar.

Com CSS não é diferente — ele é desenvolvido por um grupo dentro do W3C chamado [CSS Working Group](#). Esse grupo é formado por representantes de fornecedores de navegadores web e outras companhias que tem interesse em CSS. Também existe outras pessoas, conhecidas como peritos convidados (*invited experts*), que agem como vozes independentes; eles não são associados como um membro de alguma organização.

Novas características CSS são desenvolvidas, ou especificadas, pelo CSS Working Group. As vezes, porque um navegador em particular está interessado em alguma capacidade, outras vezes, porque designers web e desenvolvedores estão perguntando por uma característica, e, algumas vezes, porque o Working Group em si tem identificado uma necessidade. O CSS está em constante desenvolvimento, com novas peculiaridades ficando disponíveis. Contudo, uma ideia chave sobre CSS é que todos trabalham pesado para nunca alterar as coisas de uma maneira que não quebrem os sites antigos. Um site construído no ano 2000, usando um CSS limitado da época, deverá ainda ser utilizável em um navegador moderno!

Como iniciante no CSS, é provável que você ache as especificações CSS impressionantes — eles são direcionados a engenheiros para implementar suporte aos recursos nos agentes de usuário (navegadores),

não para desenvolvedores lerem com o intuito de entender CSS. Muitos desenvolvedores experientes preferem consultar a documentação do MDN ou outros tutoriais. No entanto, vale a pena saber que eles existem, entender a relação entre o CSS que você usa, suporte ao navegador (veja abaixo), e os specs (especificações).

Uma vez que o CSS tenha sido especificado, então se torna útil para nós, em termos de desenvolvimento de páginas web, apenas se um ou mais navegadores implementá-los. Isso significa que o código foi escrito para transformar as instruções do nosso arquivo CSS em algo que possa ser mostrado na tela.

CSS básico

CSS (Folha de Estilo em Cascata) é o código que você usa para dar estilo à sua página Web. CSS *básico* apresenta tudo que você precisa para começar. Responderemos a perguntas do tipo: Como mudo meu texto para preto ou vermelho? Como faço para que meu conteúdo apareça em determinados lugares na tela? Como decoro minha página com imagens e cores de fundo?

Então, o que realmente é CSS?

Assim como o HTML, o CSS não é realmente uma linguagem de programação. Também não é uma linguagem de marcação — é uma *linguagem de folhas de estilos*. Isso significa que o CSS permite aplicar estilos seletivamente a elementos em documentos HTML. Por exemplo, para selecionar **todos** os elementos parágrafo de uma página HTML e tornar o texto dentro deles vermelho, você escreveria este CSS:

```
p {  
  color: red;  
}
```

Vamos tentar: cole as três linhas de CSS acima em um novo arquivo, no seu editor de texto, e salve o arquivo como `estilo.css` na sua pasta `estilos`.

Ainda assim, precisamos aplicar o CSS ao seu documento HTML. Do contrário, o estilo CSS não irá afetar a maneira como o seu navegador mostra seu documento HTML (se você não estiver acompanhando o nosso projeto, leia [Lidando com arquivos](#) e [HTML básico](#) para descobrir o que você precisa fazer primeiro).

1. Abra seu arquivo `index.html` e cole o seguinte código em algum lugar no cabeçalho, ou seja, entre as tags `<head>` e `</head>`:

```
<link href="estilos/estilo.css" rel="stylesheet">
```

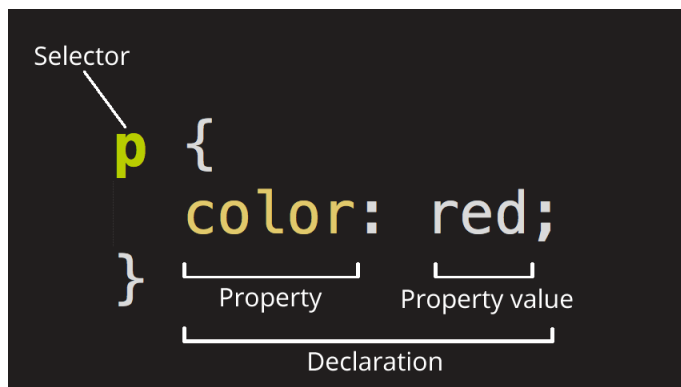
2. Salve o arquivo `index.html` e abra ele no seu navegador. Você deve ver uma página como essa:



Se o texto do seu parágrafo estiver vermelho, parabéns! Você acabou de escrever seu primeiro CSS de sucesso.

Anatomia de um conjunto de regras CSS

Vamos dar uma olhada no CSS acima com mais detalhes:



Toda essa estrutura é chamada de **conjunto de regras** (mas geralmente usamos o termo "regra", por ser mais curto). Note os nomes das partes individuais:

Seletores (Selector)

O nome do elemento HTML no começo do conjunto de regras. Ele seleciona o(s) elemento(s) a serem estilizados (nesse caso, elementos [<p>](#)). Para dar estilo a um outro elemento, é só mudar o seletor.

Declaração (Declaration)

Uma regra simples como `color: red;` especificando quais das **propriedades** do elemento você quer estilizar.

Propriedades (Property)

Forma pela qual você estiliza um elemento HTML. (Nesse caso, `color` é uma propriedade dos elementos [<p>](#).) Em CSS, você escolhe quais propriedades você deseja afetar com sua regra.

Valor da propriedade (Property value)

À direita da propriedade, depois dos dois pontos, nós temos o **valor de propriedade**, que escolhe uma dentre muitas aparências possíveis para uma determinada propriedade (há muitos valores `color(cor)` além do `red(vermelho)`).

Note outras partes importantes da sintaxe:

- Cada linha de comando deve ser envolvida em chaves (`{}`).
- Dentro de cada declaração, você deve usar dois pontos (`:`) para separar a propriedade de seus valores.
- Dentro de cada conjunto de regras, você deve usar um ponto e vírgula (`;`) para separar cada declaração da próxima.

Então para modificar múltiplos valores de propriedades de uma vez, você deve escrevê-los separados por ponto e vírgula, desse modo:

```
p {  
  color: red;
```

```
width: 500px;
border: 1px solid black;
}
```

Selecionando múltiplos elementos

Você também pode selecionar vários tipos de elementos e aplicar um único conjunto de regras a todos eles. Inclua múltiplos seletores separados por vírgulas. Por exemplo:

```
p, li, h1 {
  color: red;
}
```

Diferentes tipos de seletores

Há muitos tipos diferentes de seletores. Abaixo, nós mostramos apenas os **seletores de elementos**, que selecionam todos os elementos de um determinado tipo nos documentos HTML. Mas nós podemos fazer seleções mais específicas que essas. Aqui estão alguns dos tipos mais comuns de seletores:

Nome do seletor	O que ele seleciona	Exemplo
Seletor de elemento (às vezes, chamado tag ou seletor de tipo)	Todos os elementos HTML de determinado tipo.	p Seleciona <p>
Seletor de ID	O elemento na página com o ID especificado. Em uma determinada página HTML, é uma boa prática usar um elemento por ID (e claro, um ID por elemento) mesmo que seja permitido usar o mesmo ID para vários elementos.	#my-id Seleciona <p id="my-id"> ou
Seletor de classe	O(s) elemento(s) na página com a classe especificada (várias instâncias de classe podem aparecer em uma página).	.my-class Seleciona <p class="my-class"> e
Seletor de atributo	O(s) elemento(s) na página com o atributo especificado.	img[src] Seleciona mas não
Seletor de pseudo-classe	O(s) elemento(s) especificado(s), mas somente quando estiver no estado especificado. Ex.: com o mouse sobre ele.	a:hover Seleciona <a>, mas somente quando o mouse está em cima do link.

Há muito mais seletores para explorar e você pode achar uma lista mais detalhada em nosso [Guia de seletores](#).

Fontes e texto

Agora que exploramos algumas noções básicas de CSS, vamos começar a adicionar mais regras e informações no nosso arquivo estilo.css para deixar nosso exemplo bonito. Vamos começar fazendo nossas fontes e textos parecerem um pouco melhores.

1. Primeiro de tudo, volte e encontre a [fonte do Google Fonts](#) que você armazenou em algum lugar seguro. Adicione o elemento `<link>` em algum lugar dentro do cabeçalho no `index.html` (novamente, em qualquer lugar entre as tags `<head>` e `</head>`). Será algo parecido com isto:

```
<link href="http://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet">
```

Esse código vincula sua página a uma folha de estilo que baixa a família de fontes Open Sans junto com sua página web e permite que você a defina em seus elementos HTML usando sua própria folha de estilos.

2. Em seguida, exclua a regra existente no seu arquivo `estilo.css`. Foi um bom teste, mas o texto vermelho não parece muito bom.
3. Adicione as seguintes linhas em seu lugar, substituindo a linha do espaço reservado pela linha `font-family` que você obteve do Google Fonts. (`font-family` significa apenas a(s) fonte(s) que você deseja usar para o seu texto.) Esta regra primeiro define uma fonte base global e o tamanho da fonte para a página inteira (já que `<html>` é o elemento pai de toda a página, e todos os elementos dentro dele herdam o mesmo `font-size` e `font-family`):

```
html {font-size: 10px; /* px significa "pixels": o tamanho da fonte base é agora de 10 pixels */
font-family: "Open Sans", sans-serif; /* este deve ser o nome da fonte que você obteve no Google Fonts */
}
```

Nota: Qualquer coisa em um documento CSS entre `/*` e `*/` é um **comentário CSS**, que o navegador ignora quando renderiza o código. Este é um lugar para você escrever notas úteis sobre o que você está fazendo.

4. Agora definiremos tamanhos de fonte para elementos que contêm texto dentro do corpo HTML (`<h1>` (en-US), `` e `<p>`). Também centralizaremos o texto do nosso cabeçalho e definiremos a altura da linha e o espaçamento das letras no conteúdo do corpo para torná-lo um pouco mais legível:

```
h1 {
font-size: 60px;
text-align: center;
}

p, li {
font-size: 16px;
line-height: 2;
letter-spacing: 1px;
}
```

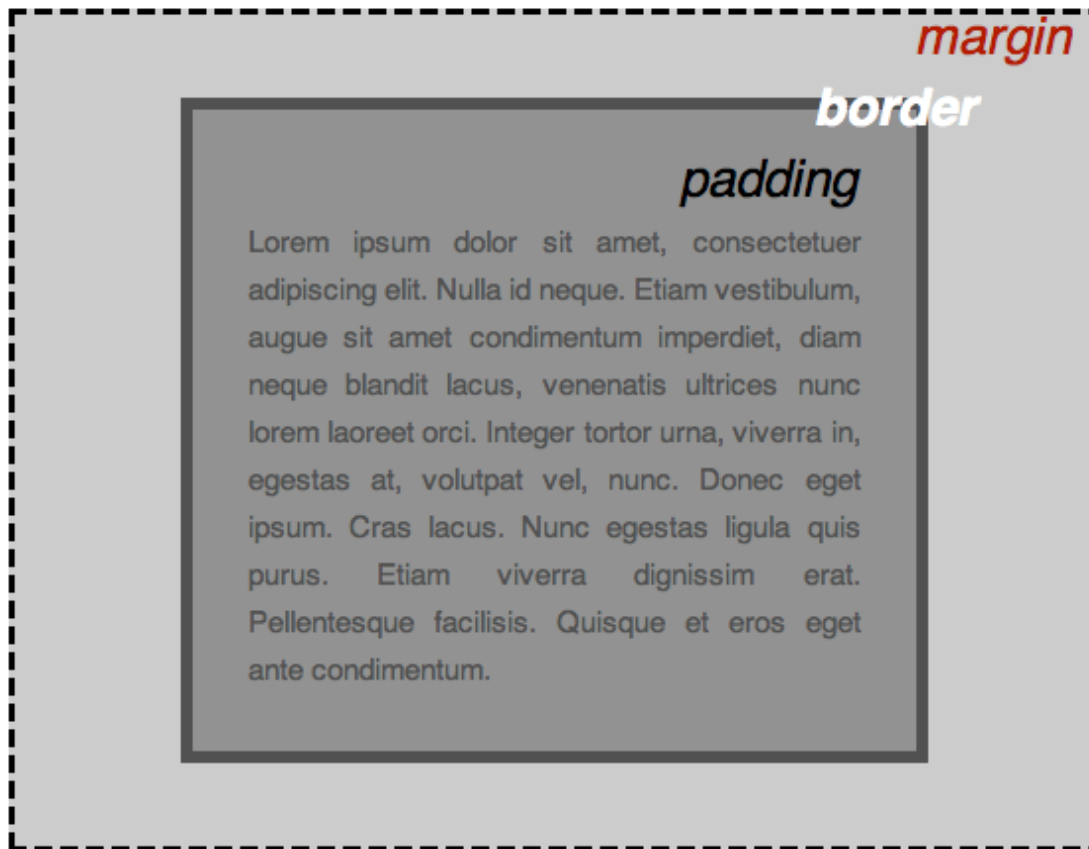
Você pode ajustar esses valores de `px` para o que você desejar, para deixar seu design com a aparência que quiser, mas no geral seu design deve parecer com isso:

Caixas, caixas, é tudo sobre caixas

Uma coisa que você notará sobre escrever CSS é que muito disso é sobre caixas — indicar seu tamanho, cor, posição, etc. Muitos dos elementos HTML da sua página podem ser pensados como caixas umas em cima das outras.

Como esperado, o layout CSS é baseado principalmente no *modelo de caixas*. Cada um dos blocks que ocupam espaço na sua página tem propriedades como essas:

- padding, o espaço ao redor do conteúdo (ex.: ao redor do texto de um parágrafo).
- border, a linha sólida do lado de fora do padding.
- margin, o espaço externo a um elemento.



Nessa seção nós também vamos usar:

- width (largura de um elemento).
- background-color, a cor atrás do conteúdo de um elemento e do padding.
- color, a cor do conteúdo de um elemento (geralmente texto).
- text-shadow: cria uma sombra no texto dentro de um elemento.
- display: define a maneira de dispor um elemento (não se preocupe com isso ainda).

Então, vamos começar e adicionar mais CSS à nossa página! Continue adicionando essas novas regras à parte inferior da página e não tenha medo de experimentar alterações nos valores para ver o que aparece.

Mudando a cor da página

```
html {  
  background-color: #00539F;  
}
```

Essa regra define uma cor de fundo para toda a página. Mude a cor acima para a cor [que você escolheu ao planejar seu site](#).

Separando o corpo

```
body {  
  width: 600px;  
  margin: 0 auto;  
  background-color: #FF9500;  
  padding: 0 20px 20px 20px;  
  border: 5px solid black;  
}
```

Agora para o elemento `<body>`. Há algumas declarações aqui, então vamos passar por elas uma a uma:

- width: 600px; — Isso força o corpo a ter 600 pixels de largura.
- margin: 0 auto; — Quando você define dois valores em uma propriedade como margin ou padding, o primeiro valor afeta a parte superior do elemento e a parte inferior (tornando-os 0 nesse caso), e no segundo valor os lados esquerdo e direito (aqui, auto é um valor especial que divide o espaço horizontal uniformemente entre esquerda e direita). Você também pode utilizar um, três ou quatro valores, como documentado [aqui](#).
- background-color: #FF9500; — como antes, isso define a cor de fundo do elemento. Usamos um tipo de laranja avermelhado no corpo, para opor ao azul escuro no elemento `<html>`, mas fique à vontade para ir em frente e experimentar.
- padding: 0 20px 20px 20px; — temos quatro valores definidos no padding, para criar um pouco de espaço em torno do nosso conteúdo. Dessa vez, estamos definindo sem padding na parte superior do corpo, e 20 pixels no lado esquerdo, na parte inferior e no lado direito. Os valores definem a parte superior, o lado direito, a parte inferior e o lado esquerdo, nessa ordem. Como com a margin, você também pode usar um, dois, ou três valores, conforme documentado na [sintaxe do padding](#).
- border: 5px solid black; — isso simplesmente define uma borda preta sólida de 5 pixels de largura em todos os lados do corpo.

Posicionando e estilizando o título da nossa página principal

```
h1 {
  margin: 0;
  padding: 20px 0;
  color: #00539F;
  text-shadow: 3px 3px 1px black;
}
```

Você deve ter notado que há um espaço horrível no topo do corpo. Isso acontece porque os browsers aplicam algumas **estilizações padrão** ao `<h1>` ([en-US](#)) (entre outros), mesmo quando você não aplicou nenhum CSS! Isso pode soar como uma má ideia, mas queremos ter uma legibilidade básica, mesmo em uma página sem estilos. Para nos livrarmos desse espaço, sobrescrevemos o estilo padrão, definindo margin: 0;.

Em seguida, definimos o padding das partes superior e inferior do cabeçalho para 20 pixels e fizemos o texto do cabeçalho da mesma cor que a cor de fundo do HTML.

Uma propriedade bastante interessante que usaremos aqui é o text-shadow, que aplica uma sombra ao conteúdo de texto do elemento. Seus quatro valores são os seguintes:

- O primeiro valor em pixel define o **deslocamento horizontal** da sombra do texto — até onde ele se move: um valor negativo deve movê-la para a esquerda.
- O segundo valor em pixel define o **deslocamento vertical** da sombra do texto — o quanto ela se move para baixo, neste exemplo; um valor negativo deve movê-la para cima.
- O terceiro valor em pixel define o **raio de desfoque** da sombra — um valor maior significará uma sombra mais borrada.
- O quarto valor define a cor base da sombra.

De novo, tente experimentar com diferentes valores para ver o que você pode criar!

Centralizando a imagem

```
img {
  display: block;
  margin: 0 auto;
}
```

Finalmente, centralizaremos a imagem para melhorar a aparência. Nós poderíamos usar novamente o truque `margin: 0 auto` que aprendemos anteriormente para o corpo, mas também precisamos fazer outra coisa. O elemento `<body>` é **em nível de bloco**, o que significa que ocupa espaço na página e pode ter margens e outros valores de espaçamento aplicados a ele. Imagens, por outro lado, são elementos **em linha**, o que significa que não podem ter margens. Então, para aplicar margens a uma imagem, temos que dar o comportamento de nível de bloco a imagem usando `display: block;`.

Nota: As instruções acima assumem que você está usando uma imagem menor que a largura definida no corpo (600 pixels). Se sua imagem for maior, ela irá transbordar o corpo e vazará para o restante da página. Para corrigir isso, você pode 1) reduzir a largura da imagem usando um [editor gráfico](#) (em inglês) ou 2) dimensionar a imagem usando CSS definindo a propriedade `width` no elemento `` com um valor menor (por exemplo, 400 px;).

Nota: Não se preocupe se você ainda não entender `display: block;` ou a distinção entre em nível de bloco / em linha. Você entenderá ao estudar CSS com mais profundidade.



Bibliografia

https://developer.mozilla.org/pt-BR/docs/Learn/CSS/First_steps/What_is_CSS

https://developer.mozilla.org/pt-BR/docs/Learn/Getting_started_with_the_web/CSS_basics