**Então, o que realmente é CSS?**[**Edit**](https://developer.mozilla.org/pt-BR/docs/Aprender/Getting_started_with_the_web/CSS_basico$edit#Então_o_que_realmente_é_CSS)

Como o HTML, CSS não é realmente uma linguagem de programação. É uma *linguagem de folhas de estilos* que permite que você aplique estilos seletivamente para elementos em documentos HTML. Por exemplo, para selecionar **todos** os parágrafos de uma página HTML e torne o texto dentro deles vermelho, você vai escrever o CSS:

p {

color: red;

}

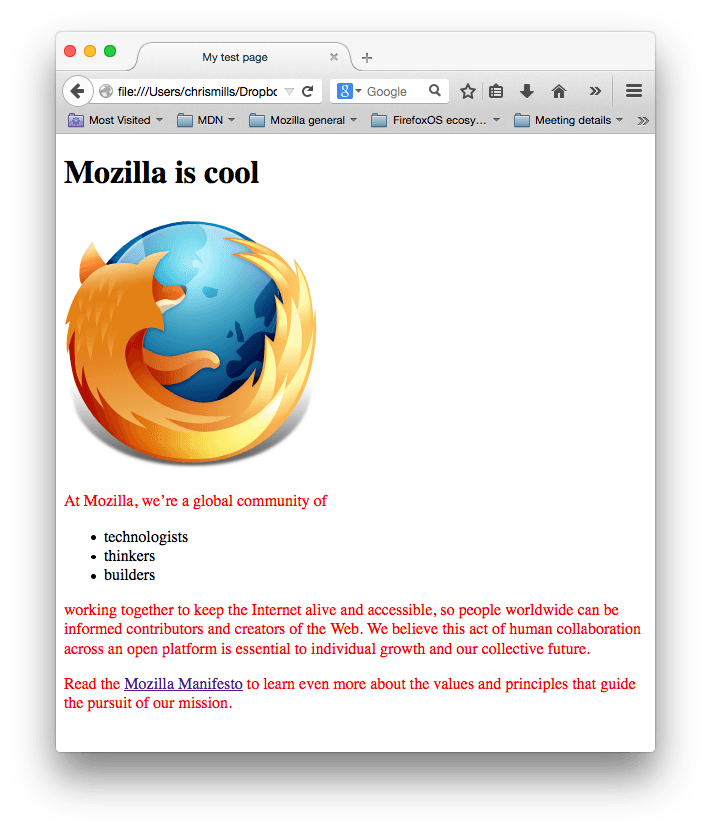
Vamos tentar: cole essas três linhas de CSS em um novo arquivo no seu editor de texto, e salve o arquivo como style.css na sua pasta styles.

Mas nós vamos precisar aplicar o CSS no documento HTML, do contrário, os códigos CSS não vão afetar como o navegador mostra seu documento HTML. (Se você não estiver seguindo nosso projeto, leia [Lidando com arquivos](https://developer.mozilla.org/en-US/Learn/Getting_started_with_the_web/Dealing_with_files) e [HTML básico](https://developer.mozilla.org/en-US/Learn/Getting_started_with_the_web/HTML_basics) e descubra o que você precisa fazer primeiro.)

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

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

1. Salve o index.html e carregue isso no seu browser. Você deve ver algo assim:

Seu parágrafo deve ser vermelho, e mesmo seu texto de parágrafo (não a lista ou as palavras em tamanho grande "Mozilla is cool"). Se isso aconteceu, parabéns, você foi bem-sucedido em escrever seu primeiro CSS!

Anatomia de um comando CSS

Vamos dar uma olhada no CSS abaixo mais detalhadamente:



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

Seletor (Selector)

O nome do elemento HTML no começo da linha de comando. 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 a cor dos elementos p.) Em CSS, você escolhe quais propriedades você quer afetar com sua regra.

Valor da propriedade (Property value)

À direita da propriedade, depois dos dois pontos, nós temos o **valor de propriedade**, para escolher uma de muitas possíveis aparências para uma propriedade (há muitos valores color além do red).

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 linha de comando, 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 pode também selecionar vários tipos de elementos e aplicar uma regra simples para todos. Inclua múltiplos seletores separados por vírgulas. Por exemplo:

p,li,h1 {

color: red;

}

Diferentes tipos de seletor

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

| **Nome do seletor** | **O que ele seleciona** | **Exemplo** |
| --- | --- | --- |
| Seletor de elemento (às vezes, chamado tag ou seletor de tipo) | Todos os elementos HTML de determinada página. | p Seleciona <p> |
| Seletor de ID | O elemento na página com o ID específico (em uma página HTML, você só pode ter um elemento por ID). | #my-id Seleciona <p id="my-id"> ou <a id="my-id"> |
| Seletor de classe | O(s) elemento(s) da página com uma classe específica (múltiplas classes podem aparecer em uma página). | .my-class Seleciona <p class="my-class"> e <a class="my-class"> |
| Seletor de atributo | O(s) elemento(s) na página com o atributo especificado. | img[src] Seleciona <img src="myimage.png"> mas não <img> |
| Seletor de pseudo-classe | O(s) elemento(s) específicado(s), mas somente em um determinado estado. Ex.: com o mouse sobre ele. | a:hover Seleciona <a>, mas somente quando o mouse está em cima do link. |

Há muitos seletores para explorar, e você pode achar uma lista mais detalhada em nosso [Guia de seletores](https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Getting_started/Selectors).

**Fontes e texto**[**Edit**](https://developer.mozilla.org/pt-BR/docs/Aprender/Getting_started_with_the_web/CSS_basico$edit#Fontes_e_texto)

Agora que nós exploramos alguns princípios do CSS, vamos começar adicionando algumas regras e informações no nosso arquivo style.css para nosso exemplo parecer bom. Primeiro, vamos dar uma melhor aparência aos nossos textos e fontes.

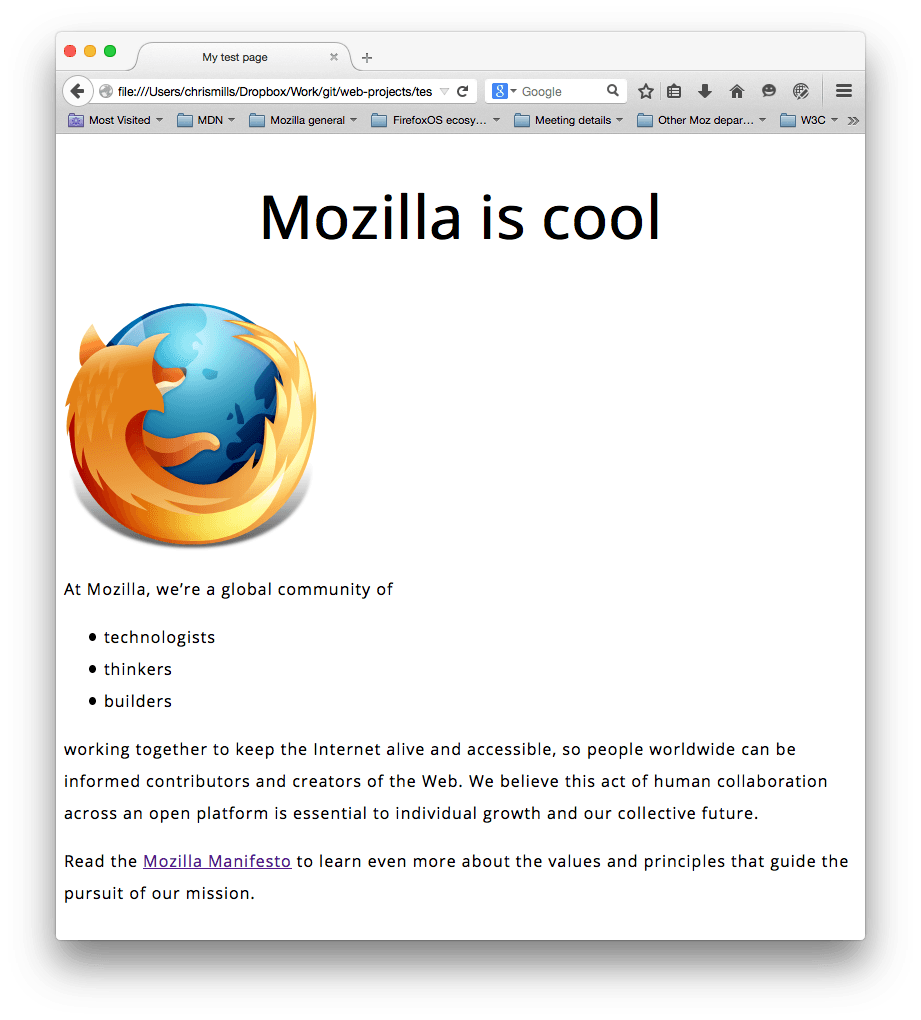
**Nota**: Eu também adicionei um comentário para explicar o que "px" significa. Tudo em um documento CSS entre /\* e \*/ é um **comentário em CSS**, que os browsers ignoram quando rodam o código. É um lugar para você escrever notas que podem te orientar no que você está fazendo.

1. Antes de tudo, volte e encontre a [fonte do Google Fonts](https://developer.mozilla.org/en-US/Learn/Getting_started_with_the_web/What_should_your_web_site_be_like#Font) que você guardou em um lugar seguro. Adicione o elemento <link ... > em algum lugar no cabeçalho do seu arquivo index.html (de novo, entre as tags <head> e </head>). Vai parecer algo assim:
2. <link href='http://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet' type='text/css'>
3. Agora, delete o comando existente no seu arquivo style.css. Foi um bom teste, mas o texto vermelho não parece muito bom.
4. Adicione as seguintes linhas no seu lugar, substituindo a linha "placeholder" com a font-family real que você obteve do Google Fonts. (font-family significa apenas a(s) fonte(s) que você quer usar em seu texto.) Essa regra define primeiro uma fonte base e o tamanho da fonte para toda a página (como <html> é o elemento pai de toda a página, então todos os elementos dentro dele herdam o mesmo font-size e font-family)
5. html {
6. font-size: 10px; /\* px significa 'pixels': o tamanho da fonte base é agora de 10 pixels \*/
7. font-family: placeholder; /\* este deve ser o nome da fonte que você obteve no Google Fonts \*/
8. }

       4. Agora, nós vamos determinar os tamanhos de fontes para os elementos com textos dentro do corpo da nossa HTML ([<h1>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/h1), [<li>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/li), and [<p>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/p)). Nós também vamos centralizar o texto do nosso cabeçalho e determinar a altura da linha e o espaçamento das letras do conteúdo do corpo para fazê-lo mais legível:

1. h1 {
2. font-size: 60px;
3. text-align: center;
4. }
5. p, li {
6. font-size: 16px;
7. line-height: 2;
8. letter-spacing: 1px;
9. }

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



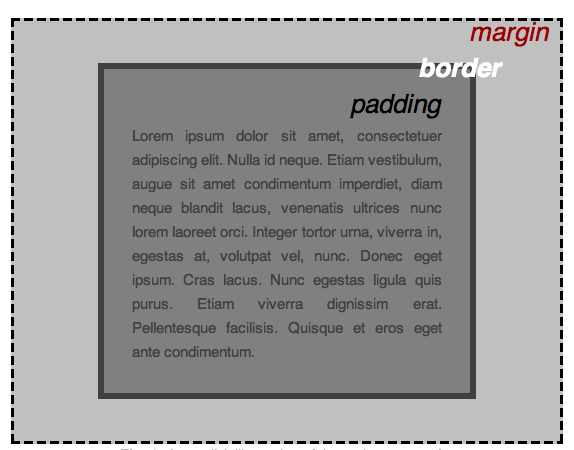
**Caixas, caixas, é tudo sobre caixas**[**Edit**](https://developer.mozilla.org/pt-BR/docs/Aprender/Getting_started_with_the_web/CSS_basico$edit#Caixas_caixas_é_tudo_sobre_caixas)

Uma coisa que você vai notar sobre escrever CSS é que um bocado 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)
* box-shadow: cria uma sombra no texto dentro de um elemento
* display: indica a maneira de dispor um elemento (não se preocupe com isso ainda)

Então, vamos começar e adicionar algum CSS a mais em nossa página! Continue adicionando essas novas regras até o fim da página e não fique com medo de mudar os valores para ver o que aparece.

Mudando a cor da página

html {

background-color: #00539F;

}

Essa regra indica uma cor de fundo para toda a sua página. Mude a cor acima para qualquer cor [que você preferir quando planejar seu site.](https://developer.mozilla.org/en-US/Learn/Getting_started_with_the_web/What_should_your_web_site_be_like#Theme_color)

Resolvendo 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 muitas 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 indica dois valores em uma propriedade como margin ou padding, o primeiro valor afeta o topo do elemento **e** o chão (0, nesse caso), e no segundo valor os lados esquerdo **e** o direito (aqui, auto é um valor especial que divide o espaço horizontal disponível igualmente entre esquerda e direita). Você também pode utilizar um, três ou quatro valores, como documentado [aqui](https://developer.mozilla.org/en-US/docs/Web/CSS/margin#Values).
* background-color: #FF9500; — como antes, isso indica a cor de fundo do elemento. Eu usei um tipo de laranja avermelhado no corpo, para opor ao azul escuro no elemento html. Vá em frente e experimente. Sinta-se livre para usar white ou o que você quiser.
* padding: 0 20px 20px 20px; — nós indicamos 4 valores no padding, para conseguir um pouco de espaço para nosso conteúdo. Dessa vez nós não estamos indicando padding no topo do corpo, e 20 pixels na esquerda, chão e direita. Os valores do padding são topo, direita, chão, esquerda, nessa ordem.
* 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 pode ter notado que há um espaço horrível no topo do corpo. Isso acontece porque os browsers aplicam algumas **estilizações padrão** ao [<h1>](https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/h1) (entre outros), mesmo quando você não aplicou qualquer CSS! Isso pode soar uma má ideia, mas nós queremos ter uma legibilidade básica, mesmo em uma página sem estilos. Para nos livrarmos daquele espaço, vamos sobrescrever o estilo padrão, indicando a margin: 0;.

Agora, nós vamos colocar o topo e o chão do padding do cabeçalho como 20 pixels, e dar ao texto do heading a mesma cor que o fundo do html.

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

* O primeiro valor em pixel indica a **distância horizontal** da sombra com relação ao texto — o quão longe ela está.
* O segundo valor em pixel indica a **distância vertical** da sombra com relação ao texto — o quão para baixo ela está.
* O terceiro valor em pixel indica o **raio de mancha** da sombra — um valor maior vai gerar um borrão maior.
* O quarto valor é a cor de base da sombra.

De novo, tente experimentar com diferentes valores para ver o que aparece.

Centralizando a imagem

img {

display: block;

margin: 0 auto;

}

Finalmente, nós vamos centralizar a imagem para fazê-la parecer melhor. Nós poderiamos usar o truque margin: 0 auto que aprendemos anteriormente para o corpo, mas nós também precisamos fazer algo a mais. O elemento body é **block level**, o que quer dizer que ele ocupa o espaço na página e pode ter margin e outros valores de espaçamento atrelados a ele. Imagens, por outro lado, são elementos **inline**, o que significa que não podem. Então, para aplicar margens a uma imagem, nós temos que dar a ela um comportamento de bloco usando display: block;.

**Nota**: Não se preocupe se você não entender display: block; ainda ou a distinção entre block-level e inline. Você vai entender, conforme estude CSS mais profundamente. Você pode achar mais sobre as diferentes maneiras de dispor valores na nossa [página de referência sobre display](https://developer.mozilla.org/en-US/docs/Web/CSS/display).

**Conclusão**[**Edit**](https://developer.mozilla.org/pt-BR/docs/Aprender/Getting_started_with_the_web/CSS_basico$edit#Conclusão)

Se você seguiu todas as instruções desse artigo, você deve terminar com uma página que pareça algo do tipo (você pode [ver nossa versão aqui](http://mdn.github.io/beginner-html-site-styled/)):

