



Sumário

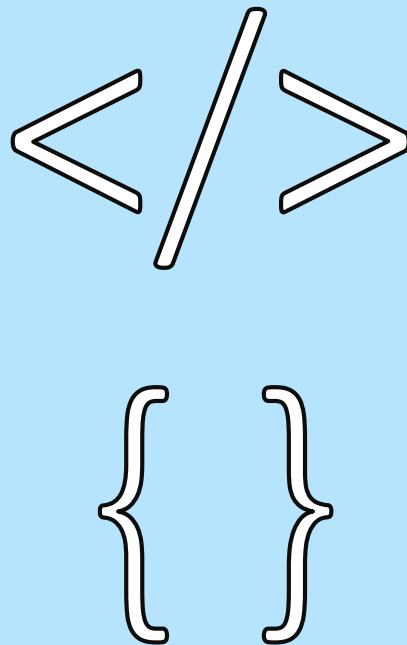
Objetivo	4
Dica de leitura	5
O que você irá construir	6
Arquitetura Básica da Web – Cliente/servidor	7
Sua primeira página Web	8
O que é HTML, CSS e JavaScript?	9
Primeiros Passos com HTML	11
Estrutura básica do HTML	14
Elementos HTML	15
Desenvolvimento HTML	17

Primeiros passos com o CSS	24
Como aplicar o CSS no HTML	26
Sintaxe do CSS	28
Inserção do CSS em nosso projeto	30
Desenvolvendo o CSS	33
Pausa para falar de cores	39
Finalizando o CSS	44
E o JavaScript?	46
Autores	47
Sobre a TargetTrust	48
Sobre a Becode	49

Objetivo

A proposta deste ebook é clara, fazer você dar os seus primeiros passos com HTML e CSS. Duas tecnologias que são essenciais no mercado de TI e que formam aquele pacote de conhecimentos que você precisa ter, independente do rumo que sua carreira irá tomar dentro do mercado de trabalho.

Em outras palavras, saber o mínimo de HTML e CSS não faz mal a ninguém e, para muitos profissionais, são habilidades imprescindíveis para uma carreira de sucesso.

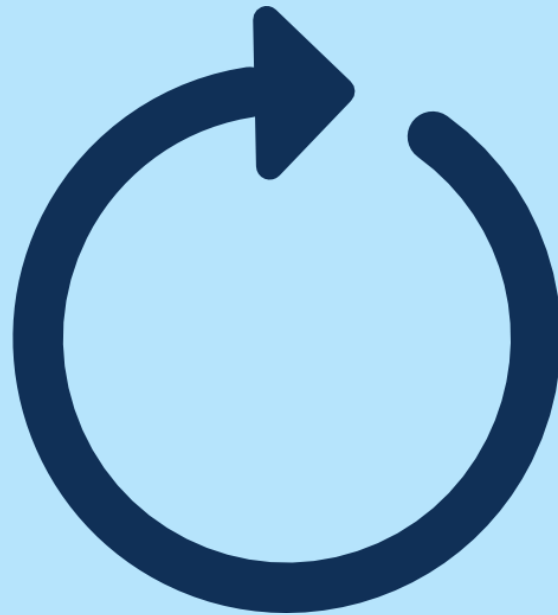


Dica de leitura

Este ebook pode ser acessado de qualquer dispositivo. Contudo, o seu aprendizado será muito mais completo se você seguir os passos desse tutorial em um **desktop ou notebook**, pois você precisará interagir com os códigos HTML e CSS. Sim, para aprender, você irá precisar praticar!

Sempre que você realizar uma alteração no código, **salve o documento .html ou .css e atualize o seu navegador**. Assim, você conseguirá identificar em tempo real, todas as modificações que você estará fazendo em sua página.

Para lembrá-lo, sempre que o ícone ao lado aparecer nesse Ebook, repita esse processo (**salvar doc + atualizar browser**).



O que você irá construir

Ao final desse ebook, você terá construído uma página web do zero, idêntica a essa aqui:

► [Página Demo – Guia HTML e CSS](#)

Sem utilizar templates ou qualquer outro conhecimento que não esteja presente nesse ebook. Resumindo, você irá aprender a teoria e aplicar na prática!

IMPORTANTE: nessa página você encontrará um botão para download. Faça-o para ter acesso aos códigos que utilizaremos ao longo desse ebook.



Arquitetura Básica da Web - cliente/servidor

Antes de começarmos a criar a nossa página, é importante que você entenda um pouco sobre a arquitetura básica da Web. Então, vamos lá.

Quando você digita “<https://becode.com.br>” em seu navegador web (Chrome, Firefox, Safari, etc), você está solicitando ao servidor da **becode**, a página inicial, também conhecida como a página index de um website. O servidor irá processar a sua requisição e disponibilizará uma espécie de “pacote de arquivos” para o seu navegador processar e exibir o conteúdo. Os arquivos neste pacote são, grosseiramente falando, os arquivos **HTML**, **CSS** e **JavaScript** da página em questão.



Sua primeira página Web

Agora, imagine que você foi contratado para desenvolver o HTML e o CSS de uma página web. Suponha que já levantamos todos os requisitos e, portanto, definimos o layout desta página e o resultado final desta página será exatamente como esse:

Obs: caso esse fosse um projeto real, provavelmente, você receberia um wireframe profissional, criado por um web designer, contendo todo a representação visual do website, layout de páginas e interações entre elas. Contudo, para o nosso propósito didático, isso não será necessário.

Esta página contém muitos elementos que, de certa forma, não são triviais quando se está aprendendo sobre HTML e CSS. Então, caso você esteja com a sensação de que não sabe por onde começar, não desanime, vamos por partes!



O que é HTML, CSS e JavaScript

De uma forma breve e resumida:

- ▶ **HTML:** linguagem de marcação que define a estrutura de sua página, é através dele que você definirá os elementos que estarão presentes em sua página, utilizando tags html;
- ▶ **CSS:** tecnologia responsável por estilizar uma página, ou seja, define cores, tamanho de elementos, alinhamento, espaçamentos, margens, efeitos e tudo aquilo que irá proporcionar uma identidade visual para a sua página;
- ▶ **JavaScript:** Por fim, o JavaScript (JS), linguagem de programação que define o comportamento de sua página, ou seja, é uma tecnologia interpretada que pode controlar o seu navegador e os elementos HTML de sua página. Elementos que se movem, alertas e outros tipos de comportamento existentes em páginas web.

Entendido a diferença entre essas tecnologias, podemos começar o desenvolvimento do nosso pequeno projeto. Lembrando, nesse ebook, iremos dar os primeiros passos com HTML e CSS. O JavaScript, por ser uma linguagem de programação, exige a compreensão de padrões lógicos de programação. Em outras palavras, para aprender JavaScript, é fundamental que o indivíduo já domine alguns conceitos de [Lógica de Programação](#).

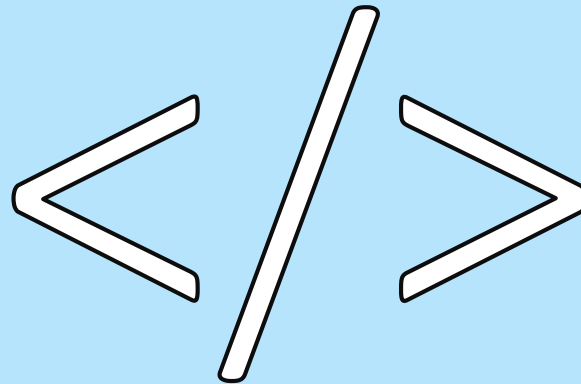
Como desenvolver a estrutura da página com HTML

Caso você nunca tenha tido contato com o HTML, saiba que ele é um arquivo como qualquer outro. Imagine um arquivo **.pdf** que é exibido pelo adobe reader, analogamente, o documento html é um arquivo **.html** com a diferença de que, ao invés de exibirmos o seu conteúdo com o adobe reader, teremos que utilizar um navegador, seja ele o **Chrome, Firefox, IE, Safari** ou qualquer outro.

Vamos começar pelo básico! Para você criar ou editar um documento html, você precisa de um editor de texto que interprete seu código HTML. Recomendamos a utilização do [sublime text](#), mas você pode escolher outros. Até o bloco de notas padrão do Windows irá servir, você quem decide!

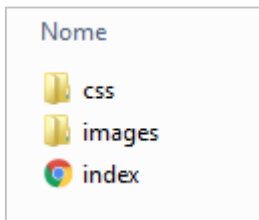


Primeiros passos com HTML



Realizado o download e instalado o seu editor de texto de sua preferência. Crie um novo documento e salve-o com o nome **index.html**, coloque este arquivo dentro de uma pasta chamada “Projeto HTML e CSS”, aproveite também para já criar uma pasta chamada **images**, para colocarmos as futuras imagens desta página.

A sua organização deve ser similar a isso:



HTML5

O HTML5 adicionou novas tags para a criação de páginas web. Essas tags auxiliam principalmente os servidores de busca a encontrar conteúdos com maior precisão, dentro dos diversos sites na web. Além, é claro, de deixar o seu código HTML mais organizado e compreensível para outros programadores que possivelmente poderão dar manutenção ao seu código em algum momento futuro.

Dito isso, é óbvio que queremos ter páginas bem estruturadas para que google, bing ou yahoo encontrem nosso site quando um usuário estiver buscando por algum assunto relacionado, bem como, ter um site organizado facilitando a manutenção realizada por outros profissionais, que não somente você.

Vamos aos nossos primeiros exemplos. Na próxima página, temos a estrutura básica de um documento HTML:

IMPORTANTE: antes de copiar os códigos, saiba que você tem acesso ao HTML, CSS e Imagens da página através [deste link \(download\)](#).



Estrutura básica do HTML

A construção de uma página HTML segue uma estrutura básica, conforme ilustrado no código ao lado.

```
<!DOCTYPE HTML>
```

```
<html>
```

```
  <head>
```

```
    ....
```

```
  </head>
```

```
  <body>
```

```
    ....
```

```
  </body>
```

```
</html>
```

Informações do documento, título da página; meta tags; links para documentos .css e .js e entre outros...

Conteúdo da página

IMPORTANTE: lembre-se de utilizar o `<meta charset="utf-8"/>`, pois assim o seu documento poderá receber acentuação, entre outros caracteres especiais da região. No nosso caso, Brasil.

```
<!DOCTYPE HTML>
```

```
<html>
```

```
  <head>
```

```
    <title>Guia de HTML e CSS</title>
```

```
    <meta charset="utf-8" />
```

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

```
  </head>
```

```
  <body>
```

```
    <header>
```

```
      ...
```

```
    </header>
```

```
    <section>
```

```
      ...
```

```
    </section>
```

```
    <footer>
```

```
      ...
```

```
    </footer>
```

```
  </body>
```

```
</html>
```



Elementos HTML

Primeiramente, temos o comando DOCTYPE, ele não é uma tag html e é utilizado para o seu navegador identificar a versão do seu documento html. Em nosso caso, estamos utilizando a versão 5. Caso você deseje pesquisar mais sobre o **doctype**, verá que, para outras versões, a forma como este comando deve ser escrito muda um pouco.

Logo após, temos as tags de nosso documento, é através delas que estruturamos um arquivo html. Note que para cada tag, temos sua abertura <> e seu fechamento </>:

- ▶ **<html> </html>**: tag que engloba todo o documento html, repare que ela começa bem no início e é a última tag a ser fechada.

```
<!DOCTYPE HTML>
```

```
<html>
  <head>
    <meta charset="utf-8" />
    <title>Becode | Cursos Online</title>
  </head>
  <body>
    <header>
      Cabeçalho do conteúdo
    </header>

    <nav>
      <a href="pagina1.html">Menu 1</a>
      <a href="pagina2.html">Menu 2</a>
      <a href="pagina3.html">Menu 3</a>
    </nav>

    <section>
      <h1>Titulo Geral</h1>
      <article>
        Texto sobre o assunto
      </article>
    </section>

    <footer>
      Informações finais que ficam no rodapé
    </footer>
  </body>
</html>
```

- ▶ **<head> </head>**: essa tag é utilizada para colocar informações a respeito de nosso documento html. Aqui não exibimos conteúdo para o usuário final, mas temos informações que serão utilizadas pelo nosso documento. Repare que a tag **<title>** não exibe nenhum conteúdo no corpo da sua página, mas sim é o título que é mostrado na aba de seu navegador, ou seja, é uma informação sobre o seu documento html.
- ▶ **<body> </body>**: a partir dessa tag que de fato iremos exibir conteúdo aos visitantes de nosso site, ela é utilizada para indicar o “corpo” do seu documento, ou seja, o seu conteúdo.
- ▶ **<header> </header>**: cuidado para não confundir com a tag **<head>**, o header tem a função de indicar que determinado conteúdo é um cabeçalho. Isso facilita o trabalho dos buscadores.
- ▶ **<nav> </nav>**: essa tag indica que temos um bloco de navegação. Como por exemplo, um menu que contém os links para as principais páginas do seu site.
- ▶ ** **: é utilizada para criação de hyperlinks em páginas web. Note que temos um parâmetro nesta tag chamado de “href”. É através dele que informamos o endereço da url de destino a que se refere a tag **<a>**. Entre a abertura da tag e o seu fechamento, colocamos o texto ou o elemento que queremos que receba o hyperlink. Este elemento também pode ser uma imagem ou outros elementos html.
- ▶ **<section> </section>**: define seções em seu documento, estas seções podem ser consideradas seções ou campos independentes.
- ▶ **<article> </article>**: utilizada para indicar que temos um bloco de conteúdo que trata-se de um post, comentário ou artigo. Essa tag também ajuda buscadores a identificar qual é o tipo de conteúdo daquele elemento. É uma tag muito comum em blogs, por exemplo.
- ▶ **<footer> </footer>**: assim como a tag **<header>** que indica cabeçalho, existe a tag **<footer>** que indica a existência de um rodapé.

Desenvolvimento HTML

Agora que você já conhece sobre cada uma das principais tags html e a estrutura básica do HTML, podemos seguir estruturando o nosso documento. Começamos pelo cabeçalho de nossa página, de acordo com o que queremos desenvolver, teremos algo similar a isso:

```
<header id="header">

  <nav>
    <a href="https://becode.com.br/cursos/">Cursos</a>
    <a href="https://becode.com.br/metodologia/">Metodologia</a>
    <a href="https://becode.com.br/sobre/">Sobre</a>
  </nav>

  <div class="inner">
    <h1>Guia HTML e CSS</h1>
    <h3>Página desenvolvida especificamente para este Guia</h3>
  </div>
</header>
```

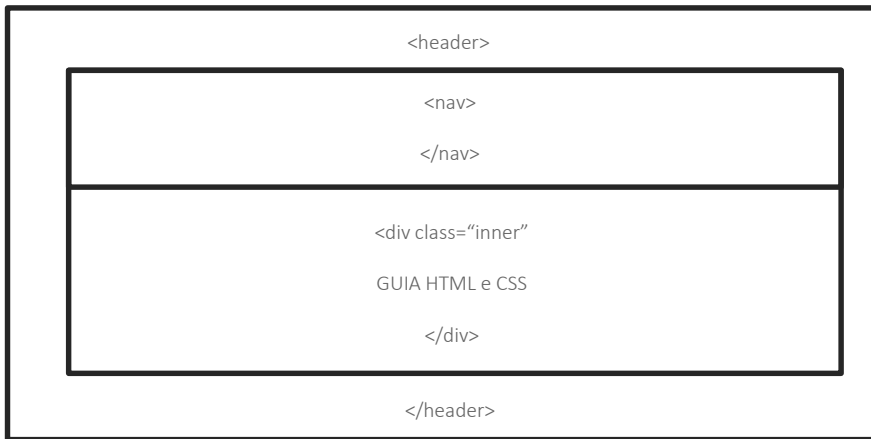


Repare que estruturamos o nosso cabeçalho de acordo com o modelo que vimos no começo deste ebook, onde tínhamos uma navegação dentro do cabeçalho e seu respectivo conteúdo. Aqui adicionamos os parâmetros **id** e **class** em nossas tags, eles são utilizados para selecionarmos elementos posteriormente com o CSS ou JavaScript.

A diferença entre eles é que IDs são únicos em um documento HTML, ou seja, não teremos duas tags distintas com o mesmo ID, já as classes não são únicas, podendo haver mais de uma tag com a mesma class atribuída no documento HTML.

Definido o header de nosso documento, precisamos pensar em como será o conteúdo abaixo desse cabeçalho. Desta forma, iremos estruturar a próxima seção da seguinte forma:

Apresentação visual do header



HTML do Conteúdo

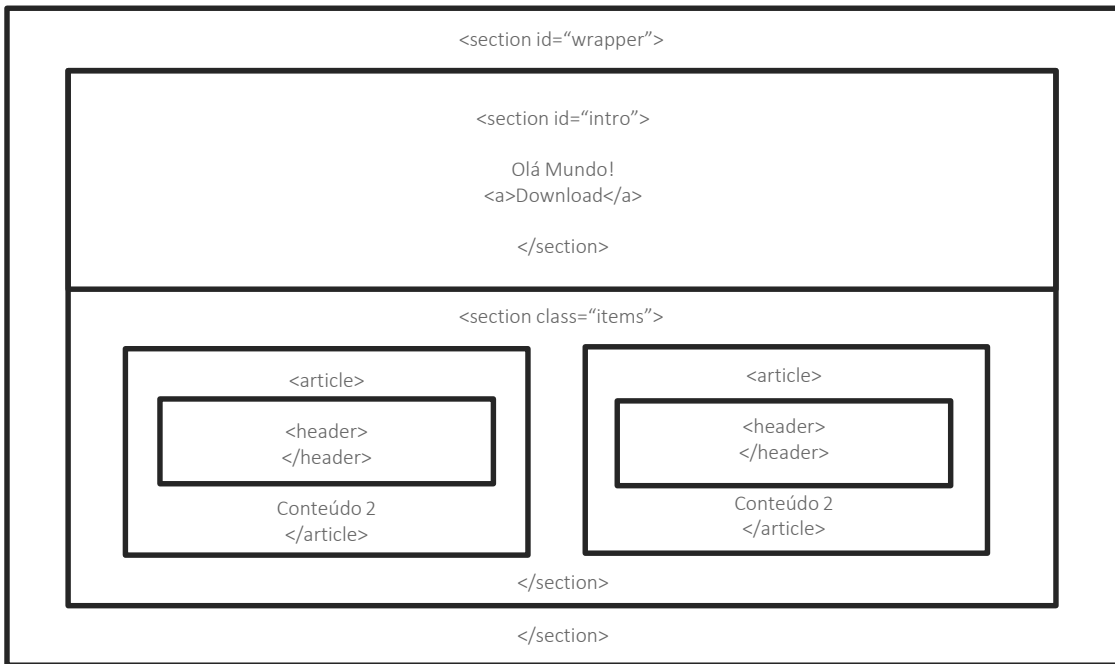
```
<section id="wrapper">
  <section id="intro">
    <h1>Olá Mundo!</h1>
    <p>Seja bem-vindo a página demonstrativa do guia de HTML e CSS da Bencode e da TargetTrust.
    Clique no botão abaixo para realizar o download dos códigos dessa página.</p>
    <ul class="actions">
      <li><a href="https://d2tycqyw09ngo1.cloudfront.net/be-content/uploads/2016/08/24185005/codigos.zip"
      class="button big"> Download </a></li>
    </ul>
  </section>
  <section class="items">
    <article class="item">
      <header>
        <a href="https://becode.com.br/top-sites-para-quem-esta-aprendendo-a-programar/">
          </a>
      </header>
      <p>Top 48 sites para quem está aprendendo a programar!</p>
      <ul class="actions">
        <li><a href="https://becode.com.br/top-sites-para-quem-esta-aprendendo-a-programar/"
        | class="button">Veja mais</a></li>
      </ul>
    </article>
    <article class="item">
      <header>
        <a href="https://becode.com.br/curso/html5-css3-e-javascript/"></a>
      </header>
      <p>Curso Online de HTML5, CSS3 e JavaScript</p>
      <ul class="actions">
        <li><a href="https://becode.com.br/curso/html5-css3-e-javascript/" class="button">Veja mais</a></li>
      </ul>
    </article>
  </section>
</section>
```



Apresentação visual do conteúdo

Perceba que nessa segunda seção, adicionamos mais elementos e, portanto, temos uma estrutura mais robusta, com uma seção que empacotará todo o nosso conteúdo (id 'wrapper'), utilizaremos essa seção para controlar melhor os espaçamentos em nossa página. Dentro dessa seção, há uma seção com uma breve introdução e mensagem de boas-vindas, que definimos com o id 'intro'. E, por fim, uma seção com os itens de nosso conteúdo, utilizando a tag `<article>`.

De preferência, opte por termos em inglês para dar nome a classes e ids. Isto é considerado uma boa prática, pois assim, pessoas de diferentes nacionalidades conseguem compreender o seu código. Se você não domina o inglês, procure memorizar os termos mais populares, mas para futuro, procure um treinamento na língua inglesa, pois isso será fundamental para sua carreira.



Antes de encerrarmos essa seção, note que algumas tags novas foram apresentadas, entre elas:

- ▶ ****: utilizado para quando queremos dar início a uma lista, neste caso, uma lista não ordenada. Caso desejássemos que fosse ordenada (1. 2. 3. ... N.), precisaríamos utilizar a tag ** **. Dentro das tags **** ou ****, utilizamos a tags **** para cada novo elemento de nossas listas.
- ▶ **<h1></h1>**: o h1 é uma tag de heading (cabeçalho), assim como o h2, h3, h4, h5 e h6. As tags possuem tamanhos diferentes e relevâncias diferentes. Em outras palavras, o h1 é maior e mais importante que o h6. Organizar a sua página de acordo com estas regras de importância é fundamental para otimizar a sua página para mecanismos de buscas (SEO), como o Google.
- ▶ **<p></p>**: a tag **<p>** é uma das tags mais utilizadas, pois significa parágrafo, ou seja, a cada novo parágrafo você precisará utilizar a tag **<p>**.
- ▶ ****: tag utilizada para quando você precisa inserir uma imagem em sua página. Repare que esta tag é acompanhada do parâmetro "src" que irá indicar o caminho dessa imagem no seu servidor web ou local.

Bom, finalizando a **seção "wrapper"** que agrupa a maior parte de nosso conteúdo, vamos enfim, definir o **nosso rodapé**, em que atribuímos um id com o nome de "footer".

Em nosso exemplo, trata-se de uma seção bem simples, veja abaixo:



Em termos de código:

```
<footer id="footer">
  <p class="copyright">&copy; Todos os direitos reservados
  <a href="https://becode.com.br">Becode</a> e <a href="http://www.targettrust.com.br">TargetTrust.</a>
</p>
</footer>
```



Concluindo o HTML

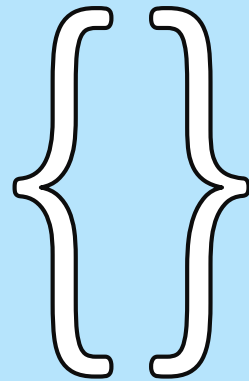
Concluindo este último pedaço de código, finalizamos a estrutura de nossa página web, salve este documento e abra-o com o seu navegador.

Você verá que o resultado não ficou muito bonito, mas não se desespere! Pois ainda não estilizamos a nossa página! Ou seja, está faltando o CSS, o qual veremos em seguida.

IMPORTANTE: finalizando o HTML, sem a presença de regras CSS, você consegue visualizar com exatidão o papel do HTML para aplicações web. Após finalizarmos a etapa de CSS, você também será capaz de ver o mesmo para o CSS. Resumindo, terá total compreensão das diferenças existentes entre o HTML e CSS.



Primeiros passos com CSS



CSS

O CSS é o que dá vida e beleza a páginas e aplicações web, ou seja, uma identidade a nossa aplicação (cores, tamanhos, estilos, efeitos e muito mais). Além disso, ele é responsável por organizar a forma como elementos HTML são ilustrados em nossos monitores, celulares, tablets, TVs, projetores, enfim... até ao imprimirmos uma página HTML, há CSS envolvido. Chamamos isso de responsividade de uma página, ou seja, a capacidade de esta página se adaptar a diferentes dispositivos, com diferentes resoluções e formatos. Tudo isso é feito com o CSS! Sendo assim, devido a crescente importância do mundo mobile nos dias de hoje, você pode imaginar o peso que saber CSS adquiriu, nos últimos anos, na hora de agarrar aquela oportunidade de trabalho.

IMPORTANTE: nesse ebook, não iremos aplicar regras de CSS que irão tratar da responsividade da página, ou seja, a capacidade de adaptação de uma página a diferentes dispositivos, como tablets, celulares, etc. Isto porque não é o foco dessa aula introdutória, sendo um conteúdo mais avançado, com suas particularidades que devem ser tratadas em outro momento, assim que você já possuir uma certa vivência com estilização de páginas HTML.

Bom, é hora de começarmos a construir o CSS de nossa página. Entretanto, antes disso é necessário entender **como aplicamos o CSS** em páginas HTML e a sua sintaxe!

Como aplicar CSS no HTML

O CSS pode ser conectado a páginas HTML de 3 formas:

1. Em um **arquivo .css** separado do arquivo .html. Nesta forma, inserimos o caminho do arquivo CSS no documento HTML, através de uma **tag <link>**, normalmente localizada entre a tag <head>:

```
2
3 <head>
4     <title>Titulo da minha página</title>
5     <link rel="stylesheet" href="caminho/arquivo.css">
6 </head>
```

2. Diretamente no documento .html, utilizando a tag `<style></style>`, entre a tag `<head></head>`. Ex:

```
<head>
  <style type="text/css">
    p {
      color: white;
    }
  </style>
</head>
```

} Regras CSS

3. Em uma tag específica, utilizando o parâmetro *style*. Ex:

```
<p style="color: white;"> Texto </p>
```

} Regras CSS

Neste e-book, iremos utilizar a 1ª forma.

Sintaxe do CSS

O CSS possui a seguinte sintaxe:

```
SELETOR {PROPRIEDADE1:valor; PROPRIEDADE2:valor;...;}
```

OU

```
SELETOR {  
    PROPRIEDADE1:valor;  
    PROPRIEDADE2:valor;  
    ....  
}
```

Obs: ambas as formas apresentam o mesmo resultado, o que muda é a apenas uma questão de boas práticas de indentação presentes no 2º código.

Onde o **SELETOR**, pode ser uma tag, um ID precedido de um sustenido ou uma classe precedida de um ponto. A **PROPRIEDADE** é o que queremos modificar naquele elemento (margem, cor, espaçamento e entre muitas outras possibilidades). Veja dois exemplos abaixo:

```
p {
  color: #000000;
  margin : 30%;
}
```

Selecionamos todas as tags <p> e adicionamos uma cor preta ao texto e uma margem de 30%.

```
#footer {
  color: #000000;
  margin : 30%;
}
```

Nesse caso, aplicamos as mesmas regras do exemplo acima, mas selecionamos o ID="footer" e não a tag <p>.

```
.item {  
color: #000000;  
margin : 30%;  
}
```

De novo, as mesmas regras dos exemplos anteriores, mas agora aplicado a classe `.item`

Inserção do CSS em nosso projeto

Para começar a desenvolver o **CSS de nossa página**, você também poderá utilizar o **Sublime text**, editor de texto recomendado antes. Pensando em organizar o nosso projeto, crie uma pasta com o nome de **“css”**, no mesmo lugar onde você criou o arquivo `index.html`.

Nesta pasta, crie um arquivo com um nome de sua escolha, mas com a extensão **.css**. Por último, adicione a tag `<link>` no arquivo HTML que estávamos trabalhando anteriormente (**index.html**). Esta tag `<link>` deve ser inserida entre a tag `<head>`, como vimos anteriormente, no tópico de aplicação do CSS no HTML.

Você deve ter algo similar a isso em sua tela:

```
<head>
  <title>Guia de HTML e CSS | Página Demonstrativa
  <meta charset="utf-8" />
  <meta name="robots" content="noindex, follow">
  <link rel="stylesheet" href="css/style.css">
</head>
```

Feito isso, abra o **arquivo .css** que você acabou de criar e vamos começar a estilizar nossa página!

IMPORTANTE: sempre que você quiser ver o resultado de uma estilização ou inserção de um novo elemento HTML em sua página. Apenas salve o documento que você está trabalhando e atualize a página do seu navegador.

Neste ebook, não teremos espaço para explicar TODAS as propriedades CSS profundamente, mas conforme iremos construindo o nosso código, serão deixados comentários próximos às propriedades, com uma breve explicação de cada propriedade apresentada. Para melhorar a sua compreensão, abuse do F5 e atualize a sua página a cada nova inserção de elemento ou regra CSS.

IMPORTANTE: tudo que estiver entre os dígitos `/* */` é um comentário no código. Ou seja, não é apresentado para o usuário final daquele website. Serve apenas para explicar para outras pessoas de sua equipe, ou até mesmo você que pode um dia esquecer qual era o seu objetivo ao escrever aquele código: “sobre o que se trata aquele trecho, script, elemento, tag, propriedade, enfim”. Comentários são mais comuns ainda quando trabalhamos com linguagens de programação, onde possuímos scripts complexos que necessitam de uma explicação mais humana para que as pessoas de uma mesma equipe possam trabalhar em conjunto.

Desenvolvendo CSS do body

Vamos começar com a tag `<body>`, pois aqui é onde devemos ter uma visão geral de nossa página. Cor de fundo, margens, espaçamento, fonte do texto, tamanho e padrões da página. Visando aquele layout inicial. Nosso CSS deve ficar da seguinte forma:

```
1
2 /* tag body */
3 body {
4     margin: 0; /* Sem margens */
5     padding: 0; /* Sem espaçamentos */
6     border: 0; /* Sem bordas */
7     background-color: #f4f4f4; /* Cor de fundo */
8     color: #666666; /* Cor de texto */
9     font-family: "Source Sans Pro"; /* Fonte do texto */
10    font-size: 150%; /* Tamanho de fonte */
11    font-weight: 200; /* Espessura de texto */
12    line-height: 1.65em; /* Espaçamento de texto */
13 }
14
```



Perceba que nossa página mudou. A cor de fundo é diferente e já não temos a margem que tínhamos no canto esquerdo da tela. A fonte do texto já não é mais a mesma e o seu tamanho e espessura mudaram.

Note que utilizamos duas unidades de medidas, o `'%` para alterarmos o tamanho da fonte e o `'em'` para alterarmos o espaçamento entre texto. Neste e-book não entraremos em detalhes sobre unidades de medidas, para saber mais sobre isso, sugiro uma rápida visita a esse link [Unidades de medida do CSS](#)

Desenvolvendo CSS do header

Após definir um estilo geral para a nossa página, **estilizando a tag body**, podemos partir para a estilização do cabeçalho de nossa página, o **header**. Lembre-se que no início, quando estávamos falando de HTML, definimos o id **header** para essa estrutura. Sendo assim, agora iremos selecioná-lo e estilizá-lo com as propriedades do CSS.

Perceba que conforme o projeto definido no início deste e-book, desejamos que o cabeçalho da página cubra a tela inteira e possua uma navegação no topo, com um conteúdo centralizado horizontalmente e verticalmente e, ainda, contendo uma imagem de fundo.

Para isso, o seu código deve ficar similar ao seguinte código:

```
16 /* Elemento com id="Header" */
17 #header {
18     background-image: url('../images/banner.jpg'); /* Imagem de fundo */
19     background-size: cover; /* Para a imagem cobrir a tela o quanto for possível */
20     background-color: #444444; /* Cor de fundo caso a imagem 'banner.jpg' não carregue */
21     text-align: center; /* Alinhamento do texto ao centro */
22     min-height: 100vh; /* Tamanho mínimo para cobrir a tela inteira na vertical */
23 }
24
```

Continuando a estilização do nosso header. Abaixo, você encontra uma sintaxe um pouco diferente do que você viu até agora. Isso porque queremos modificar apenas os elementos que estão contidos no **#header** e na classe **.inner** ou tag **h1**. Para que isso ocorra, precisamos escrever os seletores na seguinte ordem:

```
2
3 #header .inner {
4     ....
5 }
6
7 #header h1 {
8     ....
9 }
10
```



Enfim, **#header** (o elemento que está agrupando os outros elementos, seguido do elemento que está contido dentro do elemento agrupador, neste caso, **.inner**, **h1** e **h3**).

Veja abaixo como ficará o seu código:

```
/* Todos os elementos com class="inner", dentro do elemento id="header" */
#header .inner{
  margin: auto; /* Margin auto alinha os elementos 'inner' ao centro */
  padding-top: 13%; /* Espaçamento de 13% do top */
  z-index: 1000; /* Posição relativa ao eixo Z, para ficar na frente de todos os elementos */
}
```

```
/* Todos os H1s dentro do elemento com id="header" */
#header h1 {
  font-size: 2cm; /* Tamanho da fonte do h1 será de 2 centímetros*/
  padding: 0em 0.5em 0em 0.25em; /* Espaçamento: top=0em right=0.5em bottom=0em left=0.25em */
  font-family: "Source Sans Pro", sans-serif; /* duas famílias de fonte, uma titular e uma reserva
  caso o usuário não possua a primeira fonte instalada em seu computador. */
  color: #ffffff; /* Cor do texto em hexadecimal */
  font-weight: 900; /* Espessura da fonte, quanto mais, mais grossa será a fonte */
  line-height: 70px; /* Define 70px para o tamanho da linha dessa tag h1 */
  text-transform: uppercase; /* Trasforma todas as letras do texto em maiúsculas */
}
```



```
/* Todos os H3s dentro do elemento com id="header" */
#header h3 {
  font-family: "Source Sans Pro", sans-serif; /* duas famílias de fonte, uma titular e uma reserva
  caso o usuário não possua a primeira fonte instalada em seu computador. */
  color: #cccccc; /* Cor do texto em hexadecimal */
  font-weight: 400; /* Espessura da fonte, quanto mais, mais grossa será a fonte */
  margin: 0; /* retira qualquer margem padrão da tag */
  padding: 0; /* retira qualquer padding padrão da tag */
}
```

IMPORTANTE: foi selecionado o elemento header através de seu `id` porque a tag `<header>` está declarada em mais campos além do cabeçalho principal. Sendo assim, caso optássemos pela tag `<header>`, iríamos alterar todas as tags e não apenas o cabeçalho principal do topo. Por isso, utilizamos o `id #header` e não a tag `<header>`.



```
/* tag nav dentro do elemento com id="header" */
#header nav {
    background-color: #000000; /* Cor de fundo em hexadecimal */
    width: 100%; /* Largura 100% do tamanho da tela */
    margin: 0 auto; /* Margem igual a 0 acima e abaixo e automático
na horizontal para alinhar o elemento ao centro */
    padding-top: 15px; /* Padding de 15px acima da tag nav */
    padding-bottom: 15px; /* Padding de 15px abaixo da tag nav */
    z-index: 1000; /* Posição relativa ao eixo Z, para ficar na
frente de todos os elementos */
}
```

```
/* tag <a> dentro da tag nav que, por sua vez, está em #header */
#header nav a {
    margin-right: 25px; /* Margem de 25 pixel a direita de
todos os elementos com a tag <a>, contidos em #header */
    color: #cccccc; /* Cor do texto em hexadecimal */
    font-weight: 600; /* Espessura do texto */
}
```



Se você salvar o seu documento **.css** e atualizar o seu navegador, você irá notar que agora temos nosso cabeçalho completo, conforme vimos no começo deste ebook.

Pausa para falar de cores

Provavelmente você tenha reparado que, no lugar do nome de cores (**blue, black, red, etc**), utilizamos códigos como **#000000, #ffffff, #ffc711**, etc. Basicamente, estes códigos dão mais flexibilidade ao desenvolvedor na hora de escolher determinado tom de uma cor. Estes códigos são chamados de códigos hexadecimal. Existem outros tipos de códigos que também substituem cores. Outro tipo muito conhecido é o RGB que também é muito popular, exemplo: **rgb(255, 255, 255) = branco**.

Não se preocupe! Você não precisa decorar todos os códigos, basta pesquisar no google, termos como “tabela hexadecimal”, “tabela RGB”, “tabela pantone” ou ainda, se você preferir uma ferramenta mais completa, acesse o [Adobe Color CC](#). Com estes recursos, você tem acesso a todos os tipos de códigos de cores.

Desenvolvendo CSS do conteúdo

Já definidos o CSS de nosso header e do body de nossa página, atualize a página e aprecie o resultado até agora.

Entretanto, ainda não trabalhamos com o conteúdo de nossa página, isto é, o que virá após o nosso header. Para isso, vamos começar aplicando regras ao nosso empacotador, ou seja, nosso elemento com o **id="wrapper"**. Como ele é apenas um "empacotador", iremos definir a sua largura, sua margem e o espaçamento de seu conteúdo.

```
59 /* Elemento com id="wrapper" */
60 #wrapper {
61     width: 40em; /* Largura de 40em */
62     margin: 0 auto; /* Margem igual a 0 para top e bottom e auto para alinhar o elemento ao centro */
63     padding-top: 0; /* Espaçamento do topo de 0 */
64     padding-right: 5em; /* Espaçamento da direita de 5em */
65     padding-bottom: 2em; /* Espaçamento inferior de 2em */
66     padding-left: 5em; /* Espaçamento da esquerda de 5em */
67
68     /* Os espaçamentos acima poderiam ser escrito da seguinte forma: padding: 0 5em 2em 5em; */
69 }
70
71
```



Interno a este empacotador, temos uma seção para a introdução que deve situar-se no topo de nosso empacotador, estilizada da seguinte maneira:

```
/* Elemento com id="intro" */
#intro {
  padding: 4em 4em 2em 4em; /* Espaçamento: top=4em right=4em bottom=2em left=4em */
  background-color: #ffffff; /* Cor de fundo em hexadecimal */
  border-radius: 10px; /* Borda com 10 pixel de arredondamento */
  text-align: center; /* Texto alinhado ao centro */
  margin-bottom: 4em; /* Margem inferior de 4em */
  margin-top: 2em; /*margem superior de 2em */
}
```

Além do campo de introdução, temos uma seção que contém os itens de nosso conteúdo, onde declaramos uma **classe** chamada *items*.

```
/* Todos as tags com class="items", em nosso caso, a seção com os itens do conteúdo */
.items {
  display: flex; /* Define o tipo de caixa de renderização usada pelo elemento. */
  width: calc(100% + 2em); /* Calculo de largura */
  margin: -2em 0 0 -2em; /* Margem: top=-2em right=0 bottom=0 left=-2em */
}
```



Interno a esta seção, temos cada item específico declarado com uma tag `<article class="item">`, estilizada da seguinte forma:

```
/* Todos os elementos com class="item", declarados nas tags <article>*/
▼ .item {
    padding: 3em 3em 1em 3em; /* Espaçamento: top=3em right=3em bottom=1em left=3em */
    background-color: #ffffff; /*Cor de fundo em hexadecimal*/
    border-radius: 4px; /* Borda com 4 pixel de arredondamento */
    text-align: center; /* Texto alinhado ao centro */
    margin: 2em 0 0 2em; /* margem: top=2em right=0 bottom=0 left=2em */
    width: 50%; /* largura */
}

/* Todas as tags <header> dentro das classes item */
▼ .item header {
    margin: -3em 0 3em -3em; /* Margem: top=-3em right=0 bottom=3em left=-3em */
    width: calc(100% + 6em); /* Calculo de largura */
    height: 10em; /* 10 em de Altura */
    position: relative; /* Tipo de método utilizado para o posicionamento */
    overflow: hidden; /* Esconde elementos no header caso ultrapassem as margens */
}

/* Todas as tags <a> dentro da tag header */
▼ .item header a {
    background-size: cover; /* Para a imagem cobrir o fundo ao máximo */
    background-position: center; /* Centraliza o background (fundo) */
    background-repeat: no-repeat; /* Para não repetir a exibição da imagem no fundo */
    position: absolute; /* Especifica o método utilizado para posicionamento */
    width: 100%; /* Largura */
    height: 100%; /* Altura */
    display: block; /* Define o tipo de caixa de renderização usada pelo elemento. */
}
```



Desenvolvendo CSS do footer

Após o conteúdo de nosso site estar estilizado, temos enfim o footer, que irá receber uma estilização básica, pois, nesse caso não é o foco de nosso trabalho, servindo apenas como campo para atribuir os direitos autorais a TargetTrust e a Bencode.

Tendo isso em mente, aplicamos as seguintes regras para atribuir uma estilização básica.

```
121
122 /* Elemento com id="footer" */
123 ▼ #footer {
124     margin: 4em auto 0 auto; /* Margem: top=4em right=auto bottom=0 left=auto */
125     width: 40em; /* Largura */
126     text-align: center; /* Texto alinhado ao centro */
127     font-size: 0.8em; /* Tamanho da fonte */
128
129 }
130
```



Finalizando CSS

Ao abrirmos a página index no navegador veremos que a nossa página já está praticamente concluída. Faltando apenas os **botões** de nosso conteúdo, **remover a marcação padrão das tags ** e remover a **estilização padrão das tags <a>**.

Repare que em nossa estrutura html, utilizamos uma **classe chamada button** e não uma tag <button>. Através dessa classe (.button), iremos tratar as **tags <a>** para criar um estilo padrão para os nossos **“botões” <a>**. Poderíamos ter utilizado a tag <button> também e tratar a tag da mesma forma. O resultado seria muito similar. Você pode fazer o teste por conta se quiser!

```
/* Elementos com class=button */  
* .button {  
    border-radius: 4px; /* Borda com 4 pixel de arredondamento */  
    display: block; /* Define o tipo de caixa de renderização usada pelo elemento. */  
    font-weight: 400; /* Espessura do texto */  
    padding: 1em; /* Espaçamento de 1em para top, right, bottom e left */  
    text-align: center; /* Texto alinhado ao centro */  
    text-transform: uppercase; /* Transforma todas as letras do texto em maiúsculas */  
    letter-spacing: 0.325em; /* Espaçamento entre as letras do texto */  
    font-size: 0.725em; /* Tamanho da fonte */  
    background-color: transparent; /* Cor de fundo transparente */  
    border: 2px solid #9a9a9a; /* Borda do botão espessura=2px estilo=solid e cor=#666666 */  
    color: #9a9a9a; /* Cor do texto em hexadecimal */  
}
```



E, para finalizarmos, temos a remoção da **marcação das tags ** e a remoção da estilização de texto padrão das tags <a>, utilizando os seguintes comandos:

```
/* Todas as tags <ul> */
ul {
    padding-left: 0; /* Espaçamento da esquerda 0 */
}

/* Todas as tags <li>, dentro das tags <ul> */
ul li {
    display: inline-block; /* Utilizado para
    remover o ponto padrão da tag <li> */
}

/* Todas as tags <a> */
a {
    text-decoration: none; /* Remove a estilização de texto padrão das tags <a> */
}

/* Regras para todas as tags <a> e as tags <a> dentro da navegação,
para quando o usuário passar o mouse por cima do link */

#header nav a:hover, a:hover {
    color: #ffc711; /* Cor do texto em hexadecimal, cor da Becode :) */
}
```



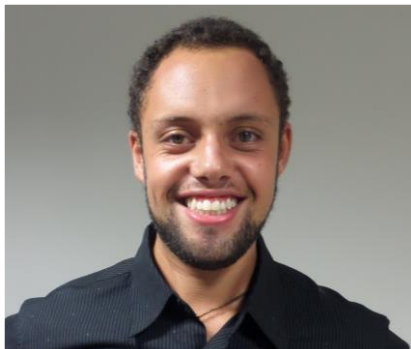
E o JavaScript?

Você deve estar se perguntando onde está o JavaScript de nossa página. Conforme comentei no início deste e-book, o JavaScript é muito importante para páginas web, ele é o responsável pelo comportamento de nossa página e seus elementos.

Imagine que essa página possuísse um slider no banner ou algum formulário em que quiséssemos tratar os seus campos antes de submetê-lo. Nesse caso, provavelmente seria o JavaScript que estaria atuando em cima dessas funcionalidades, para assim, dar comportamento à página.

Contudo, para fins didáticos, iremos deixar a parte de JavaScript para um segundo momento em um próximo conteúdo. Contudo, se você está com pressa para aprender cada vez mais sobre o mundo web, acesse o nosso [curso completo de HTML, CSS e Javascript](#).

Autores



Alysson Zanette
Software Engineer



Erick Scudero
SEO/ UX Designer





Reconhecida como a melhor escola de TI do sul do Brasil, a TargetTrust acumula mais de 20 anos de experiência em treinamentos para profissionais e empresas do mercado de TI e negócios. Veja mais sobre a TargetTrust e [nossos treinamentos](#) presenciais e In Company, acesse o nosso site.

www.targettrust.com.br





A Becode nasceu para ser a sua fonte segura de informações e aprendizado em TI e negócios na Web. E, para atingir esse objetivo, adotamos uma metodologia de ensino única em nossos treinamentos. Desta forma, proporcionando a mesma qualidade e garantia de ensino presente em cursos presenciais, agora também no ambiente online! Conheça mais [sobre a Becode](#) e nossa [metodologia](#).

www.becode.com.br

