

**COLÉGIO ESTADUAL DOM JOÃO BECKER**  
**TÉCNICO EM INFORMÁTICA**

DANIEL BUENO MOREIRA

**APOSTILA HTML**

PORTO ALEGRE-RS

2023

## Sumário

<b>Introdução à linguagem HTML .....</b>	<b>3</b>
<b>Edição de documentos HTML.....</b>	<b>4</b>
<b>Estrutura básica de uma página HTML .....</b>	<b>4</b>
<b>Tags do HTML.....</b>	<b>6</b>
<b>Lista de tags HTML .....</b>	<b>6</b>
<b>Cabeçalhos do HTML .....</b>	<b>8</b>
<b>Parágrafos no HTML .....</b>	<b>10</b>
<b>Tabelas no HTML.....</b>	<b>11</b>
<b>Publicando seu site .....</b>	<b>15</b>

## Introdução à linguagem HTML

A Linguagem de Marcação de Hipertexto (HTML) é uma linguagem de computador que compõe a maior parte das páginas da internet e dos aplicativos online. Um hipertexto é um texto usado para fazer referência a outros textos, enquanto uma linguagem de marcação é composta por uma série de marcações que dizem para os servidores da web qual é o estilo e a estrutura de um documento.

O HTML não é considerado uma linguagem de programação, já que ele não pode criar funcionalidades dinâmicas. Ao invés disso, com o HTML, os usuários podem criar e estruturar seções, parágrafos e links usando elementos, tags e atributos.

*Confira abaixo alguns dos usos mais comuns para o HTML:*

- **Desenvolvimento web.** Os desenvolvedores usam códigos HTML para projetar como um navegador vai exibir os elementos das páginas, como textos, hiperlinks e arquivos de mídia.
- **Navegação na internet.** Os usuários podem navegar facilmente e inserir links entre páginas e sites relacionados, já que o HTML é amplamente usado para incorporar hiperlinks.
- **Documentação.** O HTML torna possível a organização e a formatação de documentos, de maneira similar ao Microsoft Word.

Também vale notar que o HTML agora é considerado um padrão oficial da internet. O *World Wide Web Consortium (W3C)* mantém e desenvolve especificações do HTML, além de providenciar atualizações regulares.

## Edição de documentos HTML

Ao acessar uma página web através de um navegador, ele é capaz de interpretar o código HTML e renderizá-lo de forma compreensível para o usuário final, exibindo textos, botões etc. com as configurações definidas por meio das diversas tags que essa linguagem dispõe.

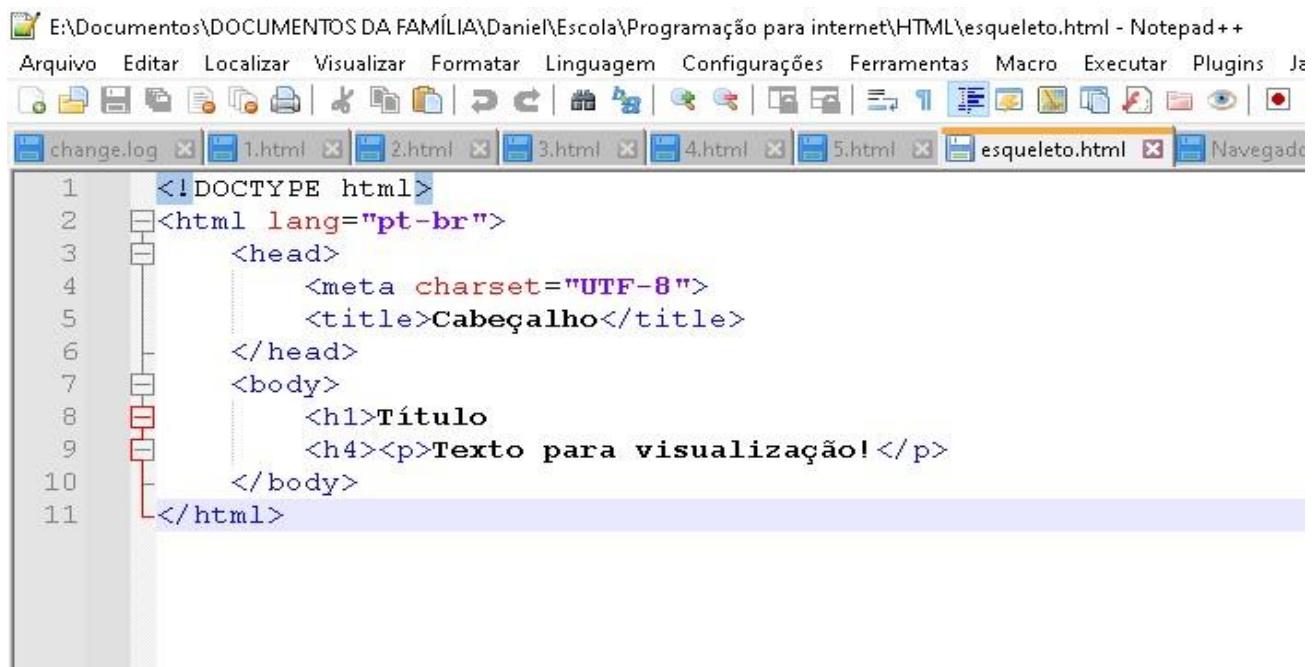
Atualmente a **HTML** encontra-se na versão 5 e é padronizada pelo W3C (World Wide Web Consortium), uma organização internacional responsável por estabelecer padrões para a internet, como a linguagem XML, CSS e o SOAP.

### Estrutura básica de uma página HTML

A **estrutura básica de uma página HTML** pode ser vista na **Listagem 1**, na qual podemos ver as principais tags que são necessárias para que o documento seja corretamente interpretado pelos browsers.

**Listagem 1.** Estrutura básica de uma página HTML (esqueleto):

*(executar em Bloco de Notas – recomendável o programa Notepad++ ou Visual studio)*



```

1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <title>Cabeçalho</title>
6  </head>
7  <body>
8      <h1>Título
9      <h4><p>Texto para visualização!</p>
10 </body>
11 </html>
  
```

**Linha 1:** a instrução *DOCTYPE* deve ser sempre a primeira a aparecer em uma página HTML para indicar ao browser qual versão da linguagem usada. Nesse caso, estamos trabalhando com a HTML5, versão na qual a declaração do DOCTYPE é bastante simples, como podemos ver na listagem;

**Linhas 2 e 14:** abertura e fechamento da tag *html*, que delimita o documento. Sendo assim, todas as demais tags da página devem estar nesse espaço;

**Linhas 3 e 6:** abertura e fechamento da tag *head*, que define o cabeçalho do documento. O conteúdo nesse espaço não é visível no browser, mas contém instruções sobre seu conteúdo e comportamento. Dentro dessa tag, por exemplo, podem ser inseridas folhas de estilo e scripts;

**Linha 4:** a tag *meta*, nesse caso, especifica qual conjunto de caracteres (character set ou charset) será usado para renderizar o texto da página. O UTF-8 contém todos os caracteres dos padrões Unicode e ASCII, sendo, portanto, o mais utilizado em páginas web. A mesma tag *meta*, porém com outros atributos, pode ser utilizada para outros fins, como na SEO (Search Engine Optimization);

**Linha 5:** a tag *title* define o título da página, aquele que aparece na janela/aba do navegador;

**Linhas 8 e 13:** abertura e fechamento da tag *body*, marcando o espaço no qual deve estar contido o conteúdo visual da página. As demais tags que representam texto, botões etc. devem ser adicionadas nesse intervalo;

**Linha 9 a 12:** Conteúdo da nossa página HTML.

Para **desenvolver páginas com HTML** basicamente precisamos de um editor de texto, como o Bloco de Notas do Windows, Nano e Emacs no Linux, entre vários outros. Há, ainda, editores com opções avançadas, como recursos de syntax highlight e autocomplete, como Sublime Text, Atom, Brackets e Visual Studio Code, que podem ser usados para **editar documentos HTML**.

Independentemente do editor utilizado, podemos simplesmente copiar o código da **Listagem 1** para um deles e salvar o arquivo com extensão *.html*. Em seguida, podemos abrir esse arquivo em um browser.

## Tags do HTML

Um documento HTML é composto por tags, as quais possuem um nome e aparecem entre os sinais `<` e `>`, como vimos na Listagem 1, por exemplo, em `<html>` e `<head>`. Naquele exemplo também vimos que algumas tags precisam ser abertas e fechadas, como em `<body> </body>`. Nesse caso, a tag de fechamento deve conter a barra `/` antes do nome. Outras, porém, não precisam ser fechadas, como a tag `<meta>`. Nesses casos, a adição da barra `/` no final da própria tag é opcional. Exemplo:

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

Outro aspecto importante da linguagem é que ela é case *insensitive*, ou seja, não leva em consideração a diferença entre letras maiúsculas e minúsculas. No entanto, o uso apenas de letras minúsculas tem sido utilizado como padrão pelos desenvolvedores.

Além dessas características, as tags também possuem atributos, como vemos na **Linha 4 da Listagem 1**, na qual a tag meta possui o atributo `charset="UTF-8"`. Essas propriedades definem algumas características adicionais de cada tag e em alguns casos são obrigatórias. Seus valores devem aparecer entre aspas duplas, como no exemplo acima, ou em aspas simples, caso o próprio valor contenha aspas duplas.

### Lista de tags HTML:

- **<!DOCTYPE>**: Define o tipo de documento e a versão do HTML que a página está utilizando.
- **<html>**: Elemento raiz que envolve todo o conteúdo da página.
- **<head>**: Contém metadados sobre a página, como o título, links para folhas de estilo e scripts.
- **<title>**: Define o título da página, que é exibido na barra de título do navegador.
- **<meta>**: Fornece metadados sobre a página, como descrição, palavras-chave e codificação de caracteres.
- **<link>**: Conecta a página a uma folha de estilo externa.
- **<script>**: Permite incluir scripts, como JavaScript, na página.
- **<body>**: Contém o conteúdo visível da página, como texto, imagens e outros elementos.
- **<h1>, <h2>, ..., <h6>**: Cabeçalhos de diferentes níveis, usados para títulos e subtítulos.
- **<p>**: Define um parágrafo de texto.

- **<a>**: Cria um link para outra página ou recurso.
- **<img>**: Insere uma imagem na página.
- **<ul>**: Cria uma lista não ordenada.
- **<ol>**: Cria uma lista ordenada.
- **<li>**: Elemento de lista, dentro de **<ul>** ou **<ol>**.
- **<table>**: Cria uma tabela.
- **<tr>**: Define uma linha em uma tabela.
- **<td>**: Define uma célula de dados em uma tabela.
- **<form>**: Cria um formulário para entrada de dados.
- **<input>**: Cria um campo de entrada, como caixa de texto, botão ou caixa de seleção.
- **<textarea>**: Cria uma área de texto para entrada de múltiplas linhas.
- **<button>**: Cria um botão clicável.
- **<select>**: Cria um menu suspenso para seleção de uma opção.
- **<option>**: Define uma opção dentro de um elemento **<select>**.
- **<label>**: Rótulo associado a um elemento de entrada.
- **<fieldset>**: Agrupa elementos de formulário relacionados.
- **<legend>**: Título descritivo para um grupo de elementos de formulário.
- **<div>**: Elemento de divisão usado para agrupar e estilizar conteúdo.
- **<span>**: Elemento inline usado para estilizar um trecho de texto ou conteúdo.
- **<hr>**: Cria uma linha horizontal para separar conteúdo.
- **<br>**: Insere uma quebra de linha.
- **<em>**: Enfatiza o texto usando itálico.
- **<strong>**: Enfatiza o texto usando negrito.
- **<abbr>**: Define uma abreviação ou sigla.
- **<blockquote>**: Citação em bloco.
- **<cite>**: Citação de trabalho criativo, como um livro ou filme.
- **<sup>**: Texto sobrescrito (superior).
- **<sub>**: Texto subscrito (inferior).
- **<iframe>**: Incorpora um documento ou recurso externo na página.
- **<nav>**: Define uma seção de navegação.
- **<header>**: Define o cabeçalho de uma seção ou página.
- **<footer>**: Define o rodapé de uma seção ou página.
- **<article>**: Define um conteúdo independente e autossuficiente, como uma postagem de blog.
- **<section>**: Define uma seção de conteúdo temático.
- **<aside>**: Define um conteúdo relacionado ou auxiliar, muitas vezes usado para barras laterais.

- **<details>**: Cria um elemento de detalhes que pode ser aberto para revelar mais informações.
- **<summary>**: Fornece um título para o elemento <details>.
- **<mark>**: Realça o texto para destaque.
- **<time>**: Representa uma data ou hora.
- **<meter>**: Cria um medidor dentro de um intervalo específico.
- **<progress>**: Exibe o progresso de uma tarefa.
- **<figcaption>**: Define uma legenda para elementos como <figure>.
- **<figure>**: Representa conteúdo independente, como imagens com legendas.
- **<datalist>**: Fornece uma lista de opções pré-definidas para elementos de entrada.
- **<audio>**: Incorpora conteúdo de áudio.
- **<video>**: Incorpora conteúdo de vídeo.
- **<source>**: Especifica múltiplas fontes de mídia para elementos <audio> e <video>.
- **<canvas>**: Cria uma área para desenho com JavaScript.
- **<details>**: Cria um elemento que pode ser aberto para mostrar detalhes adicionais.
- **<summary>**: Fornece um título para o elemento <details>.
- **<template>**: Define um conteúdo de modelo a ser clonado.
- **<dialog>**: Cria um diálogo ou caixa de mensagem.
- **<wbr>**: Indica um local onde uma possível quebra de linha pode ocorrer.
- **<bdo>**: Controla a direção da renderização do texto.
- **<ruby>**: Define anotações de pronúncia ou notas explicativas.
- **<rt>**: Define o texto da anotação de pronúncia em um elemento <ruby>.
- **<rp>**: Fornece uma representação de texto em navegadores que não suportam <ruby>.

## Cabeçalhos do HTML

Cabeçalhos são normalmente utilizados para identificar páginas e seções e possuem aparência diferenciada do restante do texto. No HTML há seis níveis de cabeçalhos/títulos que podem ser utilizados por meio das tags **h1**, **h2**, **h3**, **h4**, **h5** e **h6**, sendo **h1** o maior/mais relevante e **h6** o menor/menos relevante.

*De acordo com as regras de SEO, é recomendado que uma página possua apenas uma tag **<h1>** que indique seu assunto, pois essa tag informa aos motores de busca qual*



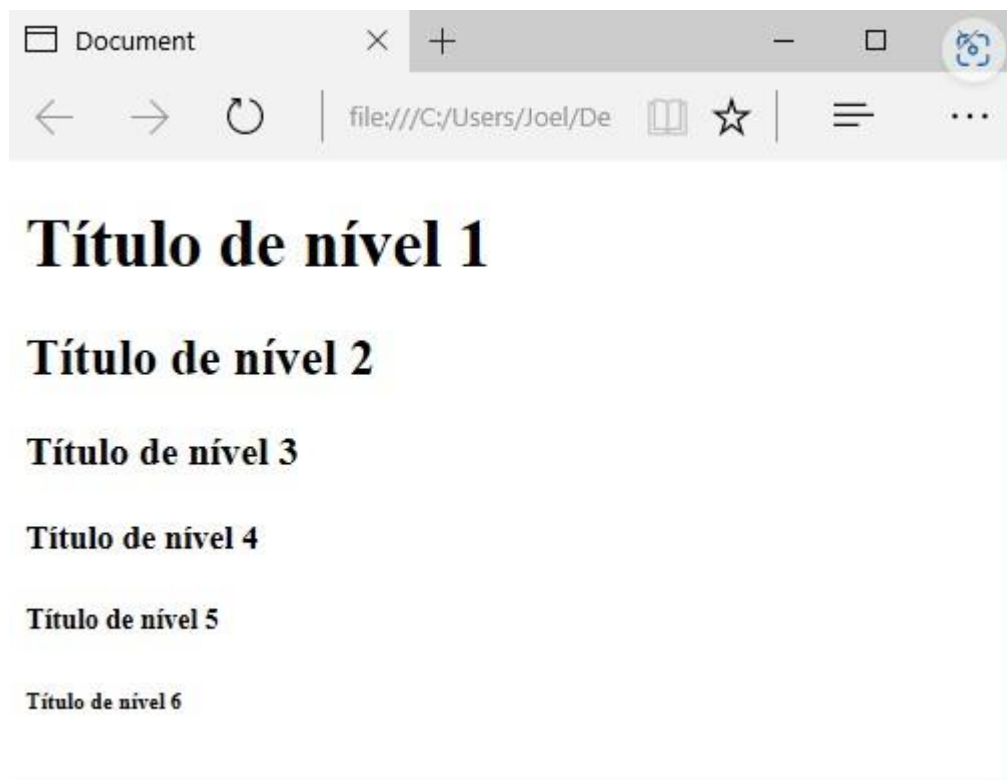
*sua principal palavra-chave.*

Na **Listagem 2** podemos ver como utilizar essas tags, e na **Figura 1** temos o resultado desse código. Para testar esse código, basta copiá-lo para o espaço indicado entre as **Linhas 9 e 12** da **Listagem 1** e salvar o documento com a extensão `.html`.

**Listagem 2** – Diferentes níveis de cabeçalho:

```
1 <h1>Título de nível 1</h1>
2 <h2>Título de nível 2</h2>
3 <h3>Título de nível 3</h3>
4 <h4>Título de nível 4</h4>
5 <h5>Título de nível 5</h5>
6 <h6>Título de nível 6</h6>
```

**Figura 1** – Diferentes níveis de cabeçalho:



## Parágrafos no HTML

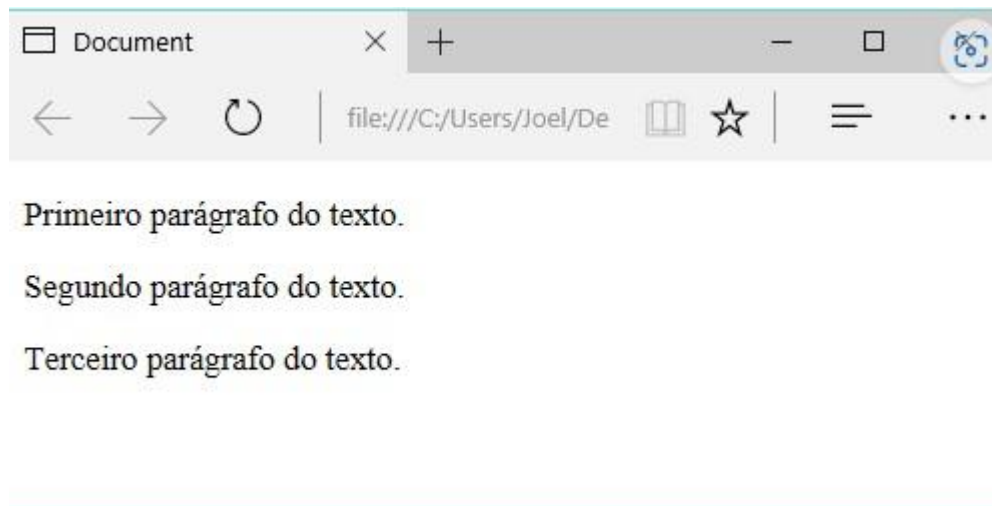
Parágrafos de texto são gerados na HTML por meio das tags `<p>` `</p>`. Esse é um exemplo de tag cuja disposição na tela se dá em forma de bloco, ou seja, um parágrafo é posto sempre abaixo do outro.

O código da **Listagem 3** adiciona três parágrafos, que podem ser visualizados na **Figura 2**.

**Listagem 3** – adicionando parágrafos:

```
1 <p>Primeiro parágrafo do texto.</p>
2 <p>Segundo parágrafo do texto.</p>
3 <p>Terceiro parágrafo do texto.</p>
```

**Figura 2** – Parágrafos visualizados na página:



*Para mais informações sobre Imagens no HTML, Links no HTML, Tabelas no HTML, Listas no HTML, Formulários no HTML, Formatação de texto, Áudio e Vídeo no HTML*

acessar esse link: [HTML básico: Códigos HTML para Iniciantes \(devmedia.com.br\)](http://devmedia.com.br/html-basico-codigos-html-para-iniciantes)

## Tabelas no HTML

Uma tabela em HTML é um elemento que permite organizar e apresentar dados em linhas e colunas. Para criar uma tabela em HTML, você precisa usar as seguintes tags:

- A tag `<table>` indica o início e o fim da tabela.
- A tag `<tr>` define uma linha da tabela.
- A tag `<td>` define uma célula da tabela.
- A tag `<th>` define uma célula de cabeçalho da tabela, que geralmente tem um texto em negrito e centralizado.
- A tag `<caption>` define um título para a tabela, que geralmente aparece acima da tabela.

Veja um exemplo de uma tabela simples em HTML e ao lado como ela ficaria no navegador:

```
<table border="1">
  <caption>Tabela de frutas</caption>
  <tr>
    <th>Nome</th>
    <th>Cor</th>
    <th>Preço</th>
  </tr>
  <tr>
    <td>Maçã</td>
    <td>Vermelha</td>
    <td>R$ 2,00</td>
  </tr>
  <tr>
    <td>Banana</td>
    <td>Amarela</td>
    <td>R$ 1,50</td>
  </tr>
  <tr>
    <td>Laranja</td>
    <td>Laranja</td>
    <td>R$ 3,00</td>
  </tr>
</table>
```

Tabela de frutas

Nome	Cor	Preço
Maçã	Vermelha	R\$ 2,00
Banana	Amarela	R\$ 1,50
Laranja	Laranja	R\$ 3,00

```

        <td>Amarela</td>
        <td>R$ 1,50</td>
    </tr>
    <tr>
        <td>Laranja</td>
        <td>Laranja</td>
        <td>R$ 3,00</td>
    </tr>
</table>

```

Para fazer um título compreender mais de uma coluna ou linha, você pode usar os atributos *colspan* e *rowspan* nas tags *<th>* ou *<td>*. Esses atributos indicam quantas colunas ou linhas a célula deve ocupar. Por exemplo, se você quiser fazer um título que ocupe duas colunas, você pode fazer assim:

```

<table border="1">
  <caption>Tabela de frutas</caption>
  <tr>
    <th colspan="2">Nome e cor</th>
    <th>Preço</th>
  </tr>
  <tr>
    <td>Maçã</td>
    <td>Vermelha</td>
    <td>R$ 2,00</td>
  </tr>
  <tr>
    <td>Banana</td>
    <td>Amarela</td>
    <td>R$ 1,50</td>
  </tr>
  <tr>
    <td>Laranja</td>
    <td>Laranja</td>
    <td>R$ 3,00</td>
  </tr>
</table>

```

Tabela de frutas

Nome e cor		Preço
Maçã	Vermelha	R\$ 2,00
Banana	Amarela	R\$ 1,50
Laranja	Laranja	R\$ 3,00

```

<tr>
  <td>Laranja</td>
  <td>Laranja</td>
  <td>R$ 3,00</td>
</tr>
</table>

```

Para fazer uma tabela sem bordas, você pode usar o atributo *border* na tag `<table>` e definir seu valor como **0**. Por exemplo:

```

<table border="0">
  <caption>Tabela de frutas</caption>
  <tr>
    <th>Nome</th>
    <th>Cor</th>
    <th>Preço</th>
  </tr>
  <tr>
    <td>Maçã</td>
    <td>Vermelha</td>
    <td>R$ 2,00</td>
  </tr>
  <tr>
    <td>Banana</td>
    <td>Amarela</td>
    <td>R$ 1,50</td>
  </tr>
  <tr>
    <td>Laranja</td>
    <td>Laranja</td>
    <td>R$ 3,00</td>
  </tr>
</table>

```

Tabela de frutas

Nome	Cor	Preço
Maçã	Vermelha	R\$ 2,00
Banana	Amarela	R\$ 1,50
Laranja	Laranja	R\$ 3,00

```

</tr>
</table>

```

Para fazer extensões de tabelas, você pode usar as tags `<thead>`, `<tbody>` e `<tfoot>` para agrupar as células de cabeçalho, corpo e rodapé da tabela, respectivamente. Essas tags podem ser úteis para aplicar estilos diferentes a cada parte da tabela, ou para facilitar a navegação por leitores de tela. Por exemplo:

```

<table border="1">
  <caption>Tabela de frutas</caption>
  <thead>
    <tr>
      <th>Nome</th>
      <th>Cor</th>
      <th>Preço</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Maçã</td>
      <td>Vermelha</td>
      <td>R$ 2,00</td>
    </tr>
    <tr>
      <td>Banana</td>
      <td>Amarela</td>
      <td>R$ 1,50</td>
    </tr>
    <tr>
      <td>Laranja</td>
      <td>Laranja</td>
      <td>R$ 3,00</td>
    </tr>
    <tr>
      <td>Total</td>
      <td colspan="2">R$ 6,50</td>
    </tr>
  </tbody>
</table>

```

Nome	Cor	Preço
Maçã	Vermelha	R\$ 2,00
Banana	Amarela	R\$ 1,50
Laranja	Laranja	R\$ 3,00
Total	R\$ 6,50	

```

        <td>R$ 3,00</td>
    </tr>
</tbody>
<tfoot>
    <tr>
        <td colspan="2">Total</td>
        <td>R$ 6,50</td>
    </tr>
</tfoot>
</table>

```

Espero que isso tenha te ajudado a entender melhor como fazer tabelas em HTML.

## Publicando seu site

Uma vez que você tenha acabado de escrever seu código e organizar os arquivos que compõem seu site, você precisa disponibilizar tudo isso online para que as pessoas possam achá-lo. Este artigo mostra como colocar online seu código de amostra simples com o mínimo de esforço.

### Quais são as opções?

Publicar um site não é um tópico simples, principalmente porque há muitas maneiras diferentes de fazê-lo. Nesse artigo, nós não pretendemos documentar todos os métodos possíveis. Em vez disso, nós discutiremos os prós e contras de três estratégias do ponto de vista de um iniciante, e então o guiaremos por um método que irá funcionar por enquanto.

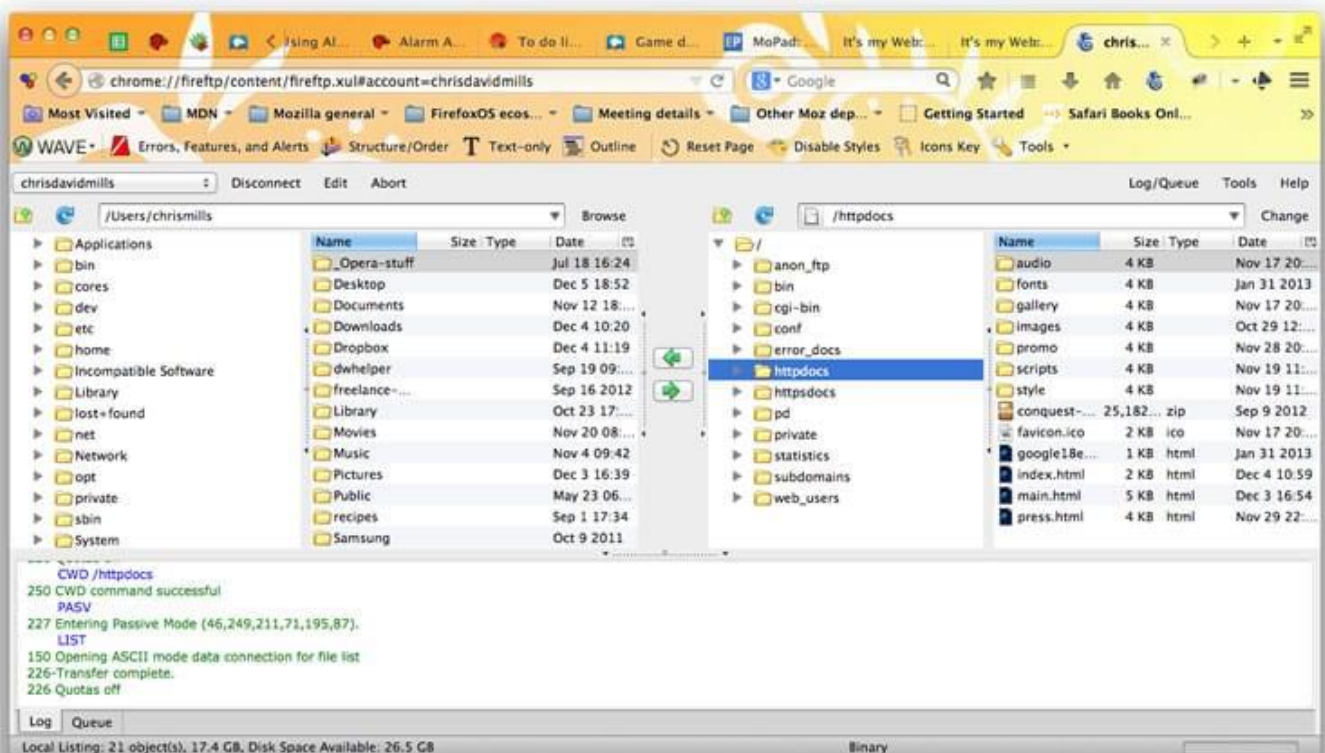
## Conseguindo uma hospedagem e um nome de domínio

Para ter mais controle sobre o conteúdo e a aparência do site, a maioria das pessoas escolhe comprar hospedagem na web e um nome de domínio:

- Hospedagem é um espaço para arquivos alugado no servidor de uma companhia de hospedagem. O servidor disponibiliza o conteúdo do site para os usuários que o visitam.
- Um nome de domínio é o endereço único onde pessoas encontram seu site, como *http://www.mozilla.org*, ou *http://www.bbc.co.uk*. Você aluga seu nome de domínio por tantos anos quanto desejar por meio de um registrador de domínio.

Muitos sites profissionais ficam online dessa maneira.

Você vai precisar ainda de um programa de [Protocolo de Transferência de Arquivos \(FTP\) \(en-US\)](#) para transferir os arquivos do site ao servidor. Programas FTP variam, mas geralmente você tem que logar no servidor usando detalhes fornecidos por sua empresa de hospedagem (ex. nome de usuário, senha, nome de host). Então, o programa mostra para você os arquivos locais e do servidor em duas janelas, para que você possa transferi-los ao servidor e de volta para o computador:





## Dicas para encontrar hospedagem e domínios

- Nós não promovemos empresas de hospedagem comercial ou para registro de nome de domínio. Para encontrar empresas de hospedagem e registradores de domínio, basta procurar por "hospedagem web" e "nomes de domínio". Todos os registradores terão um recurso para permitir que você verifique se o nome de domínio desejado está disponível.
- Seu [provedor de serviço de internet \(en-US\)](#) local ou empresarial pode fornecer uma hospedagem limitada para um site pequeno. O conjunto de recursos disponíveis será limitado, mas pode ser perfeito para seus primeiros experimentos — entre em contato com eles e pergunte!
- Há alguns serviços disponíveis gratuitamente, como [Neocities](#), [Blogspot](#) e [Wordpress](#). Às vezes, você recebe o que paga, mas às vezes esses recursos são bons o suficiente para suas experiências iniciais.
- Muitas empresas fornecem hospedagem e registro de domínio.

## Usando uma ferramenta online como Github ou Google App Engine

Algumas ferramentas permitem que você publique seu site online:

- [Github](#) é um site de "codificação social". Ele permite que você faça upload de repositórios de código para armazenamento no **sistema de controle de versão Git**. Você pode então colaborar em projetos de código, e o sistema é open-source por padrão, o que significa que todo mundo pode encontrar seu código Github, usá-lo, aprender com ele e se aprimorar. O Github tem um recurso muito útil chamado [GitHub Pages](#), que permite expor o código do site ao vivo na web.
- O [Google App Engine](#) é uma plataforma poderosa que permite criar e executar aplicativos na infraestrutura do Google, seja para criar um aplicativo da Web de várias camadas a partir do zero ou para hospedar um site estático. Veja [Como você hospeda seu website no Google App Engine?](#) Para maiores informações.

Diferente da maioria das hospedagens, essas ferramentas geralmente são gratuitas, mas você tem um conjunto limitado de recursos.

## Conclusão

Particularmente eu, Daniel Bueno Moreira, penso que só praticando e colocando em prática a HTML, que realmente qualquer pessoa interessada no assunto, vai aprender. Não é com “decoreba” ou apostilas comuns por aí.

Tentei fazer uma apostila que apareça imagens e como que vai funcionar na prática, pois sem prática, não vai aprender a HTML.

Como eu, aprendi mais na prática (nem mexi nessa apostila enquanto estava fazendo as minhas HTMLs, tive mais capricho nas páginas que fiz. Um ótimo exemplo disso, que vai servir como complemento dessa apostila, é esse link aqui:

[Ajuda na HTML e nas CSS \(danielbmoreira.github.io\)](https://danielbmoreira.github.io)

Fiz uma apostila interativa com músicas ao lado, em forma de rádio, pra pessoa praticar enquanto ouve algumas músicas.

É minha forma de pensar, não é decorando (porque nada na vida se aprende apenas decorando, tem que entender do assunto). Isso que é realmente aprender: ENTENDER o que está aprendendo, não decorando.

Opinião minha.

Obrigado.

Assinado: **Daniel Bueno Moreira**