

HTML 5 BASICS



<Por: Willian Henkel>

O BÁSICO PARA APRENDER HTML5

INTRODUÇÃO

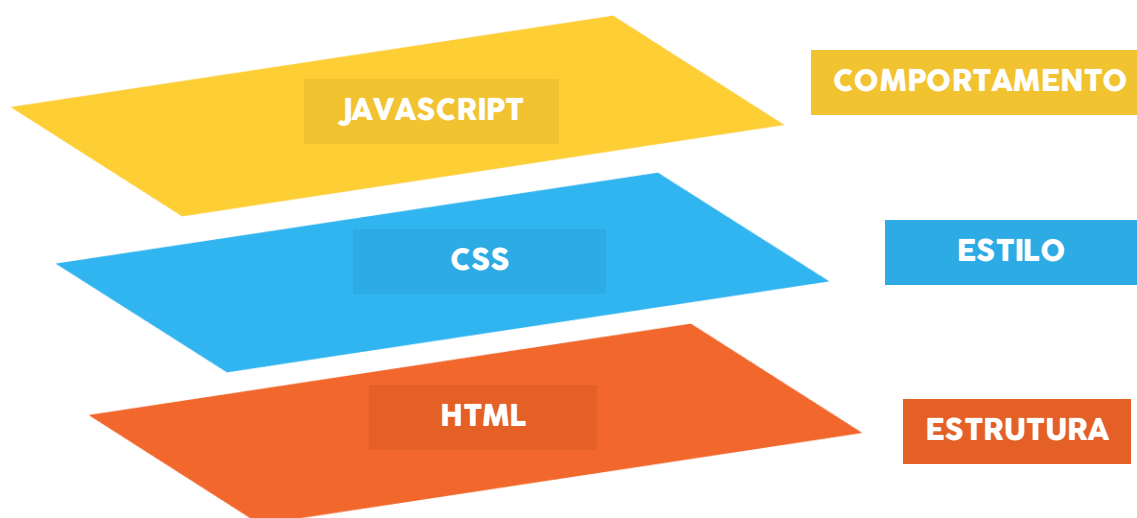
HTML5 BASICS é uma pequena apostila composta por tópicos bem definidos sobre o aprendizado básico de HTML na sua versão atualizada. Na apostila, veremos o que é HTML, principais tags, inserção de *inputs* e atributos.

O QUE É HTML?

O Hyper Text Markup Language (Linguem de Marcação de Texto), é a linguagem utilizada na construção de estrutura de uma página Web. O HTML usa as tags (marcações) para exibição do texto, como `<head>`, `<title>`, `<body>`, entre outros.

O HTML é utilizado com outras tecnologias, como o CSS para estilização da página Web e o Javascript para definir funcionalidades e comportamentos do documento.

Veja abaixo como as tecnologias se relacionam.



Podemos integrar o CSS na página Web de 3 maneiras: Externo, Interno e Inline.

Externo: Usando a tag `<link>` para conectar o documento HTML com CSS;

Interno: Usando a tag `<style>` para atribuir estilos aos elementos dentro do próprio documento HTML;

Inline: Usando o atributo `style` dentro do elemento HTML.

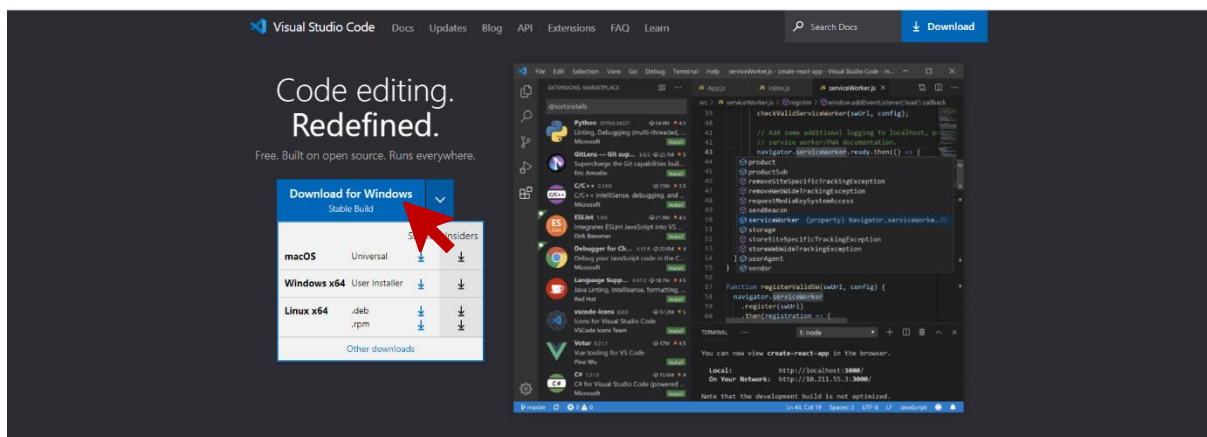
ONDE CRIAR HTML?

Podemos usar algum editor de texto, qualquer local onde conseguimos escrever e salvar o arquivo com a extensão `.html`, irá servir para criarmos um documento em HTML, entretanto já temos alguns editores de texto desenvolvidos para criação de Páginas Web. Temos alguns exemplos no mercado como: *Sublime Text editor*, *Microsoft Visual Studio Code*, *Notepad++*, entre outros. A vantagem de se utilizar esses softwares, são os auxílios que nos trazem na digitação do código, como a função de *Autocomplete*.

Além dessas opções, podemos escrever em editores online, que usamos para testes rápidos. Como por exemplo o *W3C Tryit*, você pode acessar por este [link](#).

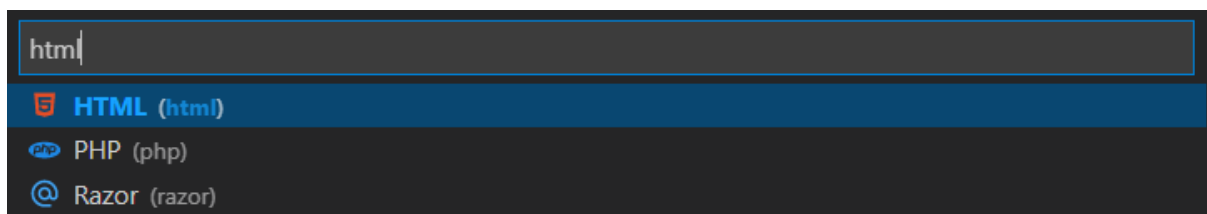
Nesta apostila, usaremos a aplicação *Microsoft VS Code*, que é atualmente o editor de texto mais famoso do mundo.

Para instalarmos o VS Code, podemos acessar o link e clicar em Download for Windows ou MacOS, depende do seu Sistema Operacional.



(link: <https://code.visualstudio.com/>)

Com o VSCode instalado, podemos selecionar a linguagem que gostaríamos de programar, no caso **HTML**.



Temos a opção de trazer a estrutura padrão HTML, utilizando o sinal de "!" no início do código e aperte ENTER ou TAB.



ESTRUTURA PADRÃO HTML

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

DISSECANDO A ESTRUTURA

- <!DOCTYPE html>

A declaração <!DOCTYPE html> representa o tipo do documento, que no caso ajuda o navegador reconhecer que estamos usando a última versão do HTML.

```
<!DOCTYPE html> <!--DECLARAÇÃO HTML5-->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//PT"
"http://www.teste.org/TR/html4.dtd"> <!--DECLARAÇÃO HTML4-->
```

É muito importante declararmos no início da estrutura!

- <html>

A tag <html> contém todos os elementos do documento html. Ele representa a raiz de toda página Web. Também podemos adicionar o atributo lang dentro da tag <html>, para definirmos a língua da página.

```
<html>
```

Ou

```
<html lang="pt-br">
```

- <head>

A tag <head> contém o metadado. O metadado são os dados que compõe o documento HTML, não é mostrado na página Web.

O elemento pode conter a tag <styles> para definirmos os estilos, scrips e tipo de caracteres.

```
<head>
```

- <meta> (metadados)

A tag <meta> define o metadado em um documento HTML. Os metadados não serão mostrados na página Web.

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

O elemento <meta charset="UTF-8">, define qual tipo de caractere será utilizado no documento. No caso, utilizamos "UTF-8" para usar o padrão Unicode, que nos ajuda com acentos.

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```

O elemento <meta http-equiv="X-UA-Compatible" content="IE=edge">, faz com que o documento HTML seja compatível com o navegador Microsoft EDGE e derivados.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

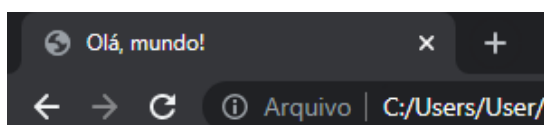
O elemento <meta name="viewport" content="width=device-width, initial-scale=1.0">, define a responsividade da página Web em diferentes dispositivos.

- <title>

```
<title>Olá, mundo!</title>
```

A tag <title>, define o nome do título da página.

Exemplo:



- <Body>

A tag body define o corpo do documento. Contém todos os elementos da Página Web, como: *Títulos, parágrafos, imagens, etc.*

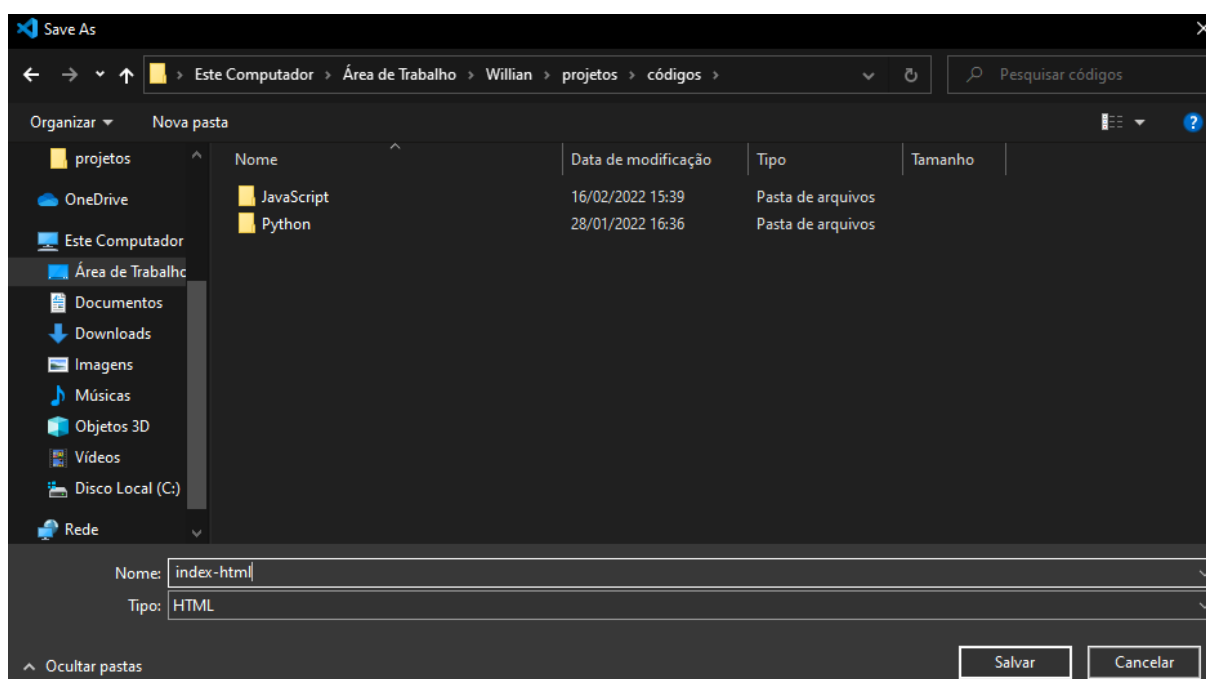
```
<body></body>
```

Exemplo:

```
<body>
  <h1>Isso é um título</h1>
  <p>Isso é um paragrafo</p>
</body>
```

SALVANDO O HTML

Para salvar o documento HTML, podemos ir em FILE (Arquivo) e clicar em Salve As (Salvar como) ou o atalho CTRL + S e colocar no diretório escolhido.



ESCREVENDO A PRIMEIRA PÁGINA

Existem diversas tags no [HTML](#), cada uma tem suas funções. Aqui veremos as principais, mas conseguimos ver todas elas e suas funcionalidades no site W3C, ([clique aqui para acessar](#)).

Vamos iniciar pelo [Headings <h1>](#) e [Paragraph <p>](#), que são elementos de texto.

Há uma regra no [HTML](#) que para abrir uma tag, precisamos iniciar com o sinal de menor do que e fechar com o sinal de maior do que. Exemplo: `<h1>`.

E para fechar a tag, precisamos adicionar um sinal de `/` após o sinal de menor que. Exemplo: `</h1>`. No fim, temos `<tag>conteúdo</tag>`.

TAG	FUNÇÃO
<code><H1></code>	Serve para adicionar um título no seu documento. Tem variáveis como as tags <code><h2></code> , <code><h3></code> , <code><h4></code> ...
<code><p></code>	Serve para adicionar um parágrafo no documento

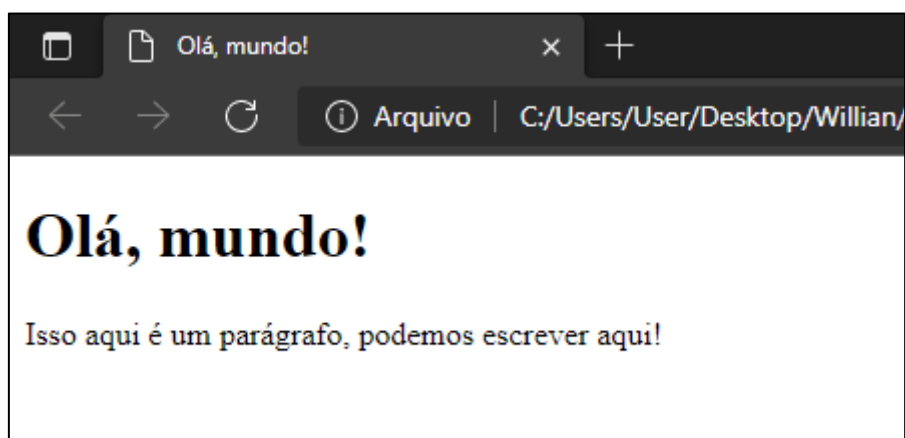
Vamos testar!

```

1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Olá, mundo!</title>
8  </head>
9  <body>
10     <h1>Olá, mundo!</h1>
11     <p>Isso aqui é um parágrafo, podemos escrever aqui!</p>
12 </body>
13 </html>

```

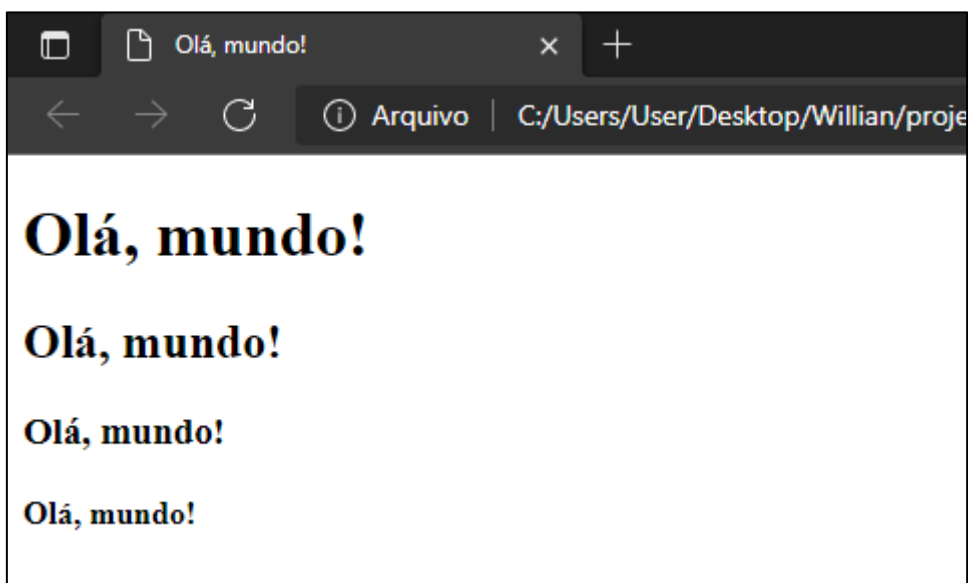
Salve o documento e abra o arquivo `.html`.



Se usarmos as tags *Headings*, temos variações de tamanho:

```
C: > Users > User > Desktop > Willian > projetos > códigos > JavaScript > Untitled-1.html > html > bod
1  <!DOCTYPE html>
2  <html lang="pr-br">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Olá, mundo!</title>
8  </head>
9  <body>
10     <h1>Olá, mundo!</h1>
11     <h2>Olá, mundo!</h2>
12     <h3>Olá, mundo!</h3>
13     <h4>Olá, mundo!</h4>
14
15 </body>
16 </html>
```

Fica:



INPUTS

Além das tags de textos, temos tags de *inputs*, onde o usuário pode digitar valores. Vamos ver os exemplos de tipo de *input* de texto e botão.

```
9  <body>
10     <h1>Inputs</h1>
11     <input type="text">
12     <input type="button" value="Botão">
13
14 </body>
```


Podemos ver que as estruturas de `input` têm valores dentro das tags. Esses valores são chamados de **atributos**.

Todos os elementos podem ter atributos e adicionam informações sobre o elemento.

No caso do elemento `input`, temos o atributo `type`, que adiciona a informação do tipo do `input`. Temos vários tipos de inputs, como o de senha, e-mail e etc. Aqui usamos o tipo `text` e `button`.

Veja também que no `input` do botão, temos o atributo `value` que determina o valor que estará escrito dentro do botão.

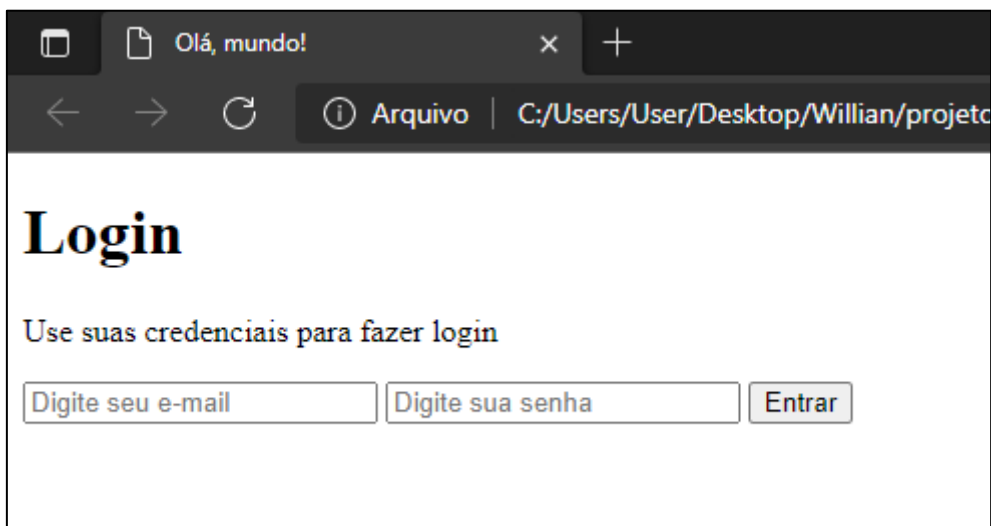
Vamos criar uma tela de login apenas com o html usando o que aprendemos até agora.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Olá, mundo!</title>
</head>
<body>
  <!-- LOGIN -->

  <h1>Login</h1>
  <p>Use suas credenciais para fazer login</p>
  <input type="email" placeholder="Digite seu e-mail">
  <input type="password" placeholder="Digite sua senha">
  <input type="button" value="Entrar">

</body>
</html>
```

Fica:



Note que neste último código, adicionei um comentário informando que a aquela estrutura era de um login.

```
<!--LOGIN-->
```

Os comentários não serão processados pelo navegador no momento que a Página Web carregar. Comentários servem exclusivamente para direcionar o código

Temos também a inclusão do atributo `placeholder`, que coloca um valor que será alterado após o usuário começar a digitar.

Podemos fazer essa estrutura sem o atributo `placeholder` para identificar os inputs para o usuário. Ao invés, utilizamos a tag `<label>` e o atributo `name`.

```
<!DOCTYPE html>
<html lang="pr-br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Olá, mundo!</title>
</head>
<body>
  <!--LOGIN-->

  <h1>Login</h1>
  <p>Use suas credenciais para fazer login</p>

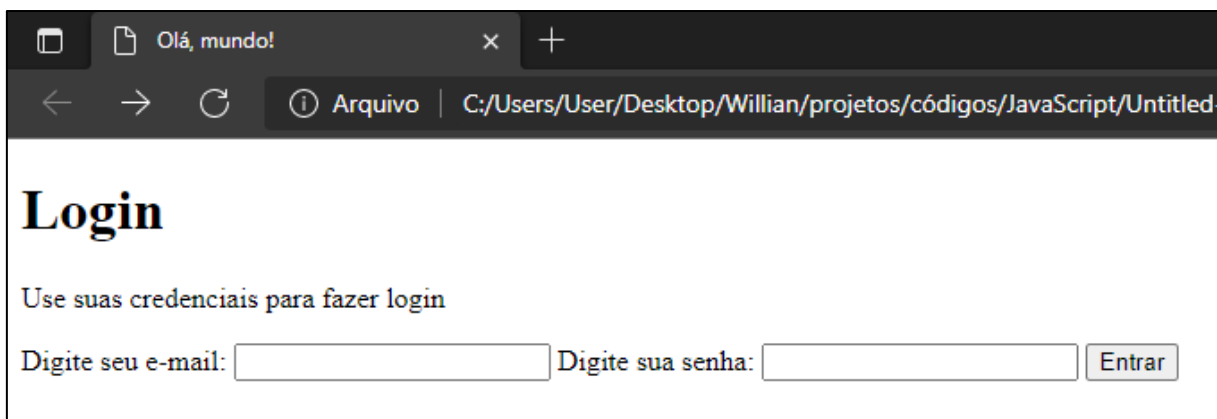
  <!--email-->
  <label for="inputEmail">Digite seu e-mail:</label>
  <input type="email" name="inputEmail">

  <!--senha-->
  <label for="inSenha">Digite sua senha:</label>
  <input type="password" name="inSenha">

  <!--botão-->
  <input type="button" value="Entrar">

</body>
</html>
```

Fica:



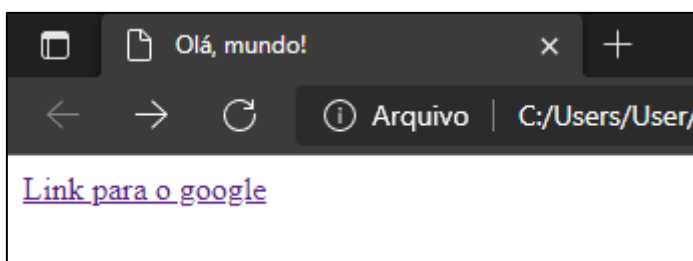
Dentro da tag `<label>`, temos o atributo `for`, que faz uma referência com o atributo `name`, que está dentro da tag do input. Em `name`, criamos o nome de referência do input que deve ser o mesmo no `for`.

- <a>

Podemos referenciar outras páginas usando a tag `<a>`. Exemplo:

```
<a href="https://www.google.com">Link para o google</a>
```

Fica:



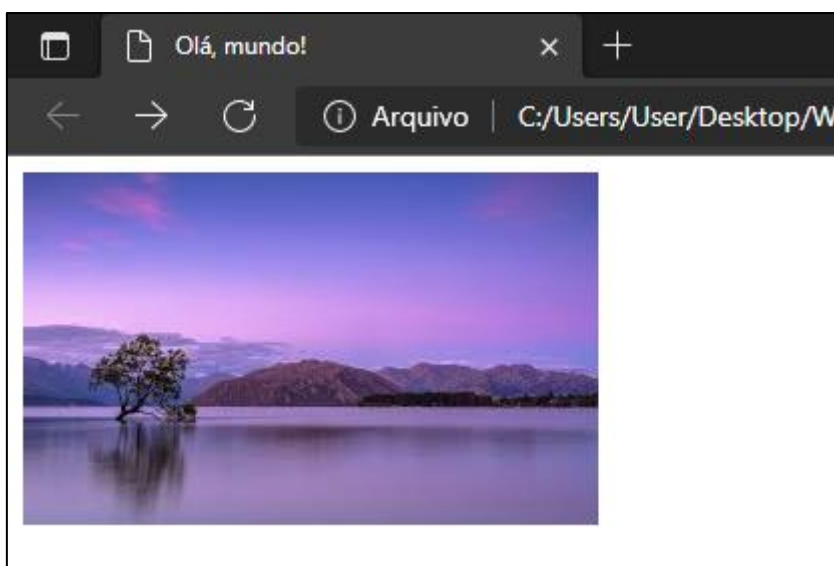
-

Para adicionarmos imagens em nosso documento, podemos usar a tag ``. A tag `` não precisa ser fechada usando ``. Exemplo:

```

```

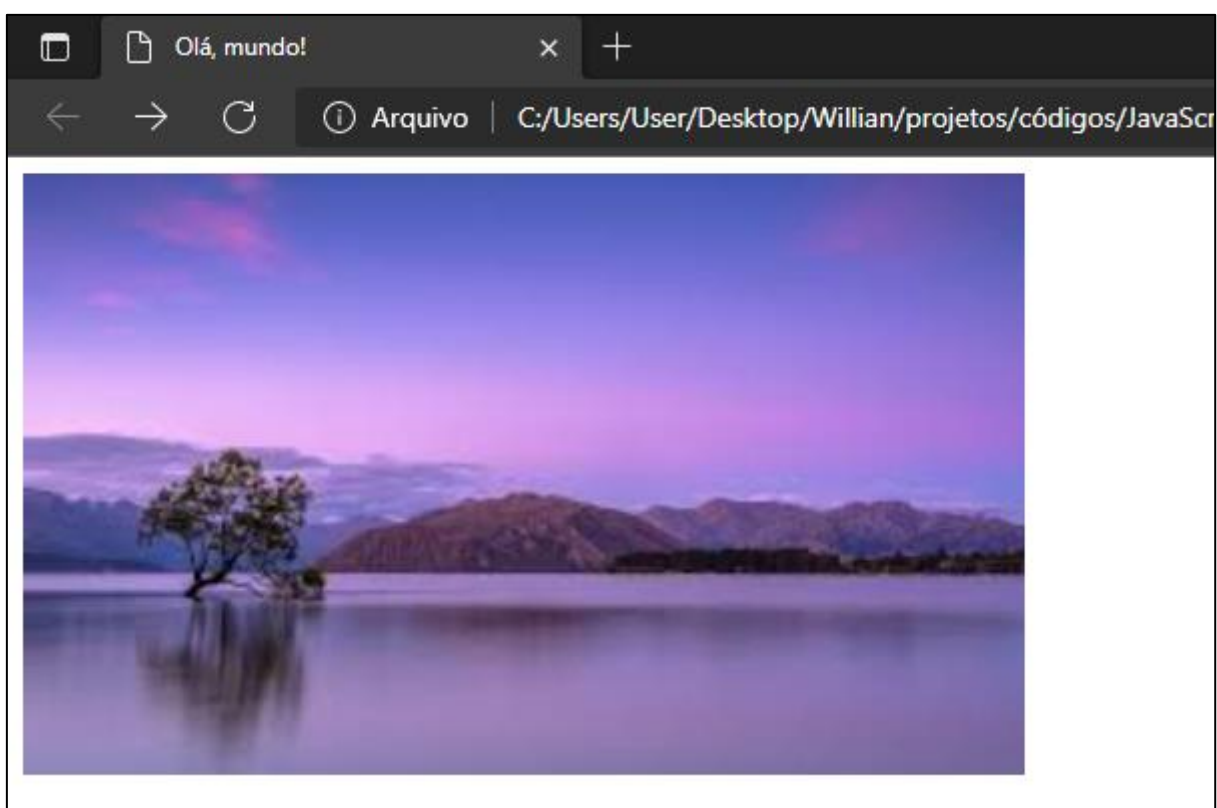
Fica:



Conseguimos alterar as dimensões da nossa imagem usando o atributo `width` para largura e `height` para altura.

```

```



-

Agora, vamos supor que temos um texto grande, mas precisamos quebrar a linha. Para esse caso, usamos a tag
.

```
<p>
  Lorem ipsum dolor sit amet consectetur, adipisicing elit.
  Animi deserunt nulla culpa <br>dolorem excepturi? Dolorem iste,
  modi sequi <br>aspernatur velit, atque distinctio sed earum rerum eos
  corporis, nobis ab. Officiis!
</p>
```

Fica:

