



## Programação Web Front-End

### Aula 1 - HTML

Profa. Rosangela de Fátima Pereira Marquesone  
romarquesone@utfpr.edu.br

**Proposta:** apresentar uma introdução à linguagem HTML, possibilitando a criação de um código básico em HTML na IDE Visual Studio Code (VS Code) e a visualização do resultado em um navegador.

**Objetivos:** espera-se que após essa aula, você tenha habilidade para compreender os seguintes tópicos:

1. [Compreender o que é HTML](#)
2. [Compreender o que são tags HTML básicas](#)
3. [Criar um arquivo .html no Visual Studio \(VS\) Code](#)
4. [Abrir o arquivo .html em um navegador](#)
5. [Visualizar o código-fonte de uma página em um navegador](#)
6. [Inspecionar a página em um navegador](#)
7. [Utilizar o Live Server no VS Code](#)

#### Dicas de aprendizado:

- Execute todos os passos com atenção, compreendendo o que está sendo realizado;
- Procure não copiar código, para ter a prática de digitar o código desenvolvido;
- Pergunte quando tiver alguma dúvida;
- Mantenha um histórico dos códigos desenvolvidos, seja no github ou em algum outro meio de armazenamento (e-mails, google drive, etc.);
- Tenha curiosidade e explore os recursos apresentados.

---

# Passo 1 - Compreender o que é HTML

A linguagem que o navegador consegue interpretar para exibir conteúdo é o **HyperText Markup Language (HTML)**, ou Linguagem de Marcação de Hipertexto. Sua função é enviar para o navegador (*browser*) informações que definem de que maneira textos, imagens e outros itens deverão ser apresentados. Ou seja, não se trata de uma linguagem de programação, como C, Java e Python, por exemplo, e sim, uma linguagem de marcação, nesse caso de hipertexto. Cabe lembrar que **hipertexto** consiste em conjuntos de elementos (e.g., palavras, imagens, áudios), ligados por conexões, que formam uma rede conectada.

A primeira versão do HTML foi escrita pelo cientista britânico Tim Berners-Lee (criador do conceito **World Wide Web**), criado em 1989, quando ele trabalhava na Organização Europeia para a Pesquisa Nuclear (CERN). Esta primeira versão continha somente **18 tags** HTML. Atualmente existem aproximadamente **140 tags**.

## Curiosidades!

Em comemoração ao desenvolvimento da World Wide Web, um grupo de desenvolvedores e designers reuniu-se no CERN em 2019 para reconstruir o navegador original, permitindo que usuários de todo o mundo experimentassem as origens bastante humildes desta tecnologia transformadora.

Veja como era o navegador por este [link](#).

Conforme é apresentado neste [link](#) da World Wide Web Consortium (W3C), o HTML criado por Tim Berners-Lee foi baseado em Standard Generalized Markup Language (SGML), um padrão aceito internacionalmente para marcar texto em unidades estruturais como parágrafos, itens de lista e assim por diante. Por exemplo, o uso de pares de tags como <TITLE> e </TITLE>, que usamos em HTML, é originado do SGML, que possui exatamente o mesmo comportamento.

Os elementos SGML usados no HTML de Tim Berners-Lee incluíam também: P (parágrafo); H1 a H6 (nível de título 1 até nível de título 6); OL (listas ordenadas); UL (listas não ordenadas); LI (itens de lista) e vários outros. Conforme é destacado no texto, o que o SGML não possui e foi inventado por Tim Berners-Lee são os links de hipertexto, que usam o elemento âncora com o atributo HREF, possibilitando a ligação entre os documentos.

Em 1995 é lançado o HTML 2.0, contendo todos os recursos do HTML 1.0, além de recursos adicionais. Em janeiro de 1997 surge o HTML 3.0 (mais especificamente o HTML 3.2), que adicionou novos recursos de HTML, provendo novas funcionalidades na construção de websites. Foi nesse período também que a W3C tornou o HTML um padrão a ser adotado pelos navegadores.

## Para saber mais!

Você já ouviu falar sobre a **Guerra dos Navegadores** ou Guerra dos *Browsers*? Esse termo refere-se ao período entre 1995 e 1999, no qual houve uma grande competição pelo domínio de mercado de navegadores (*web browsers*), principalmente entre a Netscape e a Microsoft, com o Internet Explorer.

Saiba mais sobre como tudo isso aconteceu neste vídeo:

[A história da guerra dos navegadores - TecMundo](#) (~10 minutos)

Caso queira ver essa história com ainda mais detalhes, você pode assistir a esse vídeo:

[The True Story of the Internet E01 Browser Wars](#) (~43 minutos)

Embora o HTML 3.0 tenha sido considerado um avanço, alguns recursos tornaram as aplicações mais lentas no navegador, necessitando de melhorias. Em 1999, surge então o HTML 4.0, sendo uma versão que aceitava folhas de estilo externas, permitindo desenvolver um arquivo CSS externo, podendo esse também ser incluído no próprio HTML.

Em 2014 foi lançado o **HTML5**, contendo novos recursos muito utilizados atualmente, tais como:

- Suporte ao armazenamento local
- Introdução de novas API's (*Application Programming Interface*)
- Adição de dois elementos de reprodução de mídia:
  - <video>
  - <audio>
- Novos elementos semânticos, contendo uma representação mais explícita da estrutura do documento, tais como:
  - <header>
  - <article>
  - <nav>
  - <section>
  - <footer>

O HTML5 foi originado pelo Web Hypertext Application Technology Working Group ou WHATWG, criado por desenvolvedores de empresas web, tais como Mozilla, Apple e Opera. O grupo existe até hoje, é de acesso livre e pode ser visualizado neste [link](#).

Você também pode consultar as atualizações do HTML5 por esse [link](#) da W3C.

Para dar continuidade ao conhecimento sobre HTML, veremos na sequência algumas informações sobre as **tags HTML**.

---

## Passo 2 - Compreender o que são tags HTML básicas

Uma das maneiras de compreender as tags HTML é organizando-as por meio de suas funções. Por exemplo, há tags referentes a elementos básicos de um documento HTML. A tag de abertura **<html>** e a tag de fechamento **</html>** corresponde ao elemento raiz do documento. Ou seja, todos os outros elementos devem ser descendentes desse elemento, conforme a estrutura a seguir:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
</head>
<body>
</body>
</html>
```

**OBSERVAÇÃO:** O doctype refere-se a uma instrução aos navegadores de que é preciso renderizar a página em um modo padrão (*standard*), seguindo a especificação html atual. Ou seja, a declaração **<!DOCTYPE HTML>** define que o documento é um documento HTML5.

Na tag **<html>** já é possível identificarmos que uma tag pode ter atributos. Nesse caso, incluímos o atributo **lang**, usado para definir o idioma da página HTML. Veremos no decorrer da disciplina muitos outros exemplos de atributos usados em diferentes elementos.

Outros dois elementos básicos são as tags **<head>** e **<body>**. A tag **<head> </head>** refere-se a um elemento no qual inserimos informações sobre o documento que não são mostradas na tela. É utilizada para informar ao navegador as configurações da página. Assim, ela normalmente contém informações como o título da página **<title>**, links para arquivos CSS e links para outros metadados, conforme o exemplo a seguir:

```
<head>
  <title>Minha página</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
```

O elemento **<title>Minha página</title>** define o título da página, que aparece na guia do navegador onde a página é carregada.

As tags **<meta>** são utilizadas para inserir metadados de uma página. O elemento **<meta charset="UTF-8">**, por exemplo, é usado para definir o conjunto de caracteres que o documento deve usar para o UTF-8, incluindo assim um vasto conjunto de caracteres de diferentes idiomas.

Por fim, o elemento **<meta name="viewport" content="width=device-width, initial-scale=1">** é usado para controlar como um site é exibido em um dispositivo, tornando o layout responsivo. Veremos mais sobre isso no decorrer da disciplina.

Além da tag **<head>**, outro elemento básico de um documento HTML é a tag **<body> </body>**. Essa tag contém todo o conteúdo exibido na página, incluindo texto, imagens e vídeos. Um exemplo pode ser visto a seguir:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
<title>Minha página</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
</head>

<body>
<h1>Minha primeira página web</h1>
<p>Aprendendo a utilizar HTML na prática.</p>
</body>
</html>
```

Nesse exemplo, dentro da tag body foram incluídas novas tags, referentes ao conteúdo da página.

Veremos no decorrer do curso inúmeras tags a serem utilizadas, mas, para exemplificar, veja o significado das tags <h1> e <p>:

- <h1></h1>: refere-se a um elemento de cabeçalho. Esses elementos permitem você especificar que determinadas partes do seu conteúdo referem-se a títulos ou subtítulos. Assim, um documento HTML pode conter até 6 níveis de título, indo de <h1> a <h6>.
- <p></p>: são elementos usados para incluir parágrafos de texto, sendo utilizado com muita frequência em uma página.

Um outro fator que pode te auxiliar no desenvolvimento de uma página HTML é saber que ela suporta comentários. Os navegadores ignoram os comentários, assim como ocorre com os compiladores nas demais linguagens de programação.

Para inserir um comentário em seu código HTML, inclua os marcadores “<!--” e “-->” para iniciar e fechar um bloco de comentário, conforme o exemplo a seguir:

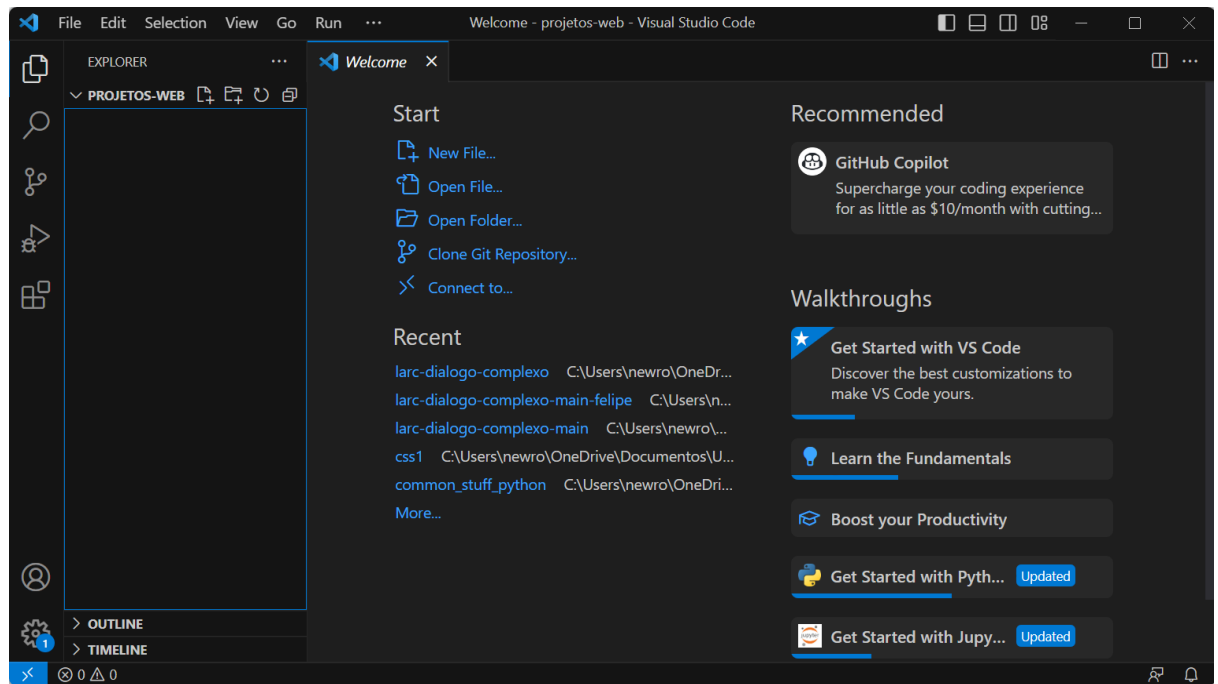
```
<body>
<!-- Esse é um comentário em uma página -->
<h1>Minha primeira página web</h1> <!-- outro comentário -->
<p>Aprendendo a utilizar HTML na prática.</p>
</body>
```

Uma lista com todos os elementos HTML podem ser encontrados no site da W3Schools, por esse [link](#).

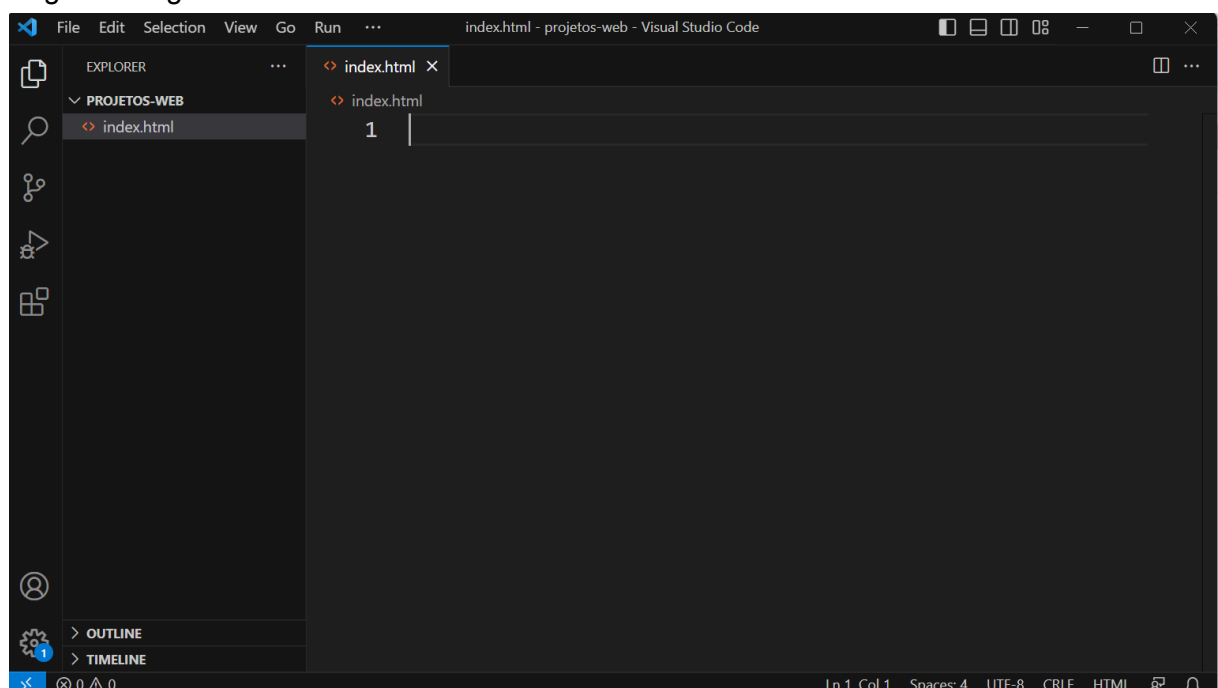
É claro que ainda há muito mais a aprender sobre HTML, porém, a partir das informações apresentadas, você pode dar início ao desenvolvimento da sua primeira página Web, seguindo os passos a seguir. Vamos lá?

## Passo 3 - Criar um arquivo .html no Visual Studio (VS) Code

1. Abra o Windows Explorer e crie uma pasta chamada “projetos-web” dentro do diretório “Documentos”.
2. Abra o VS Code e selecione a opção “File >> Open folder”, selecionando a opção projetos-web. Essa ação deverá gerar um cenário igual à figura a seguir:



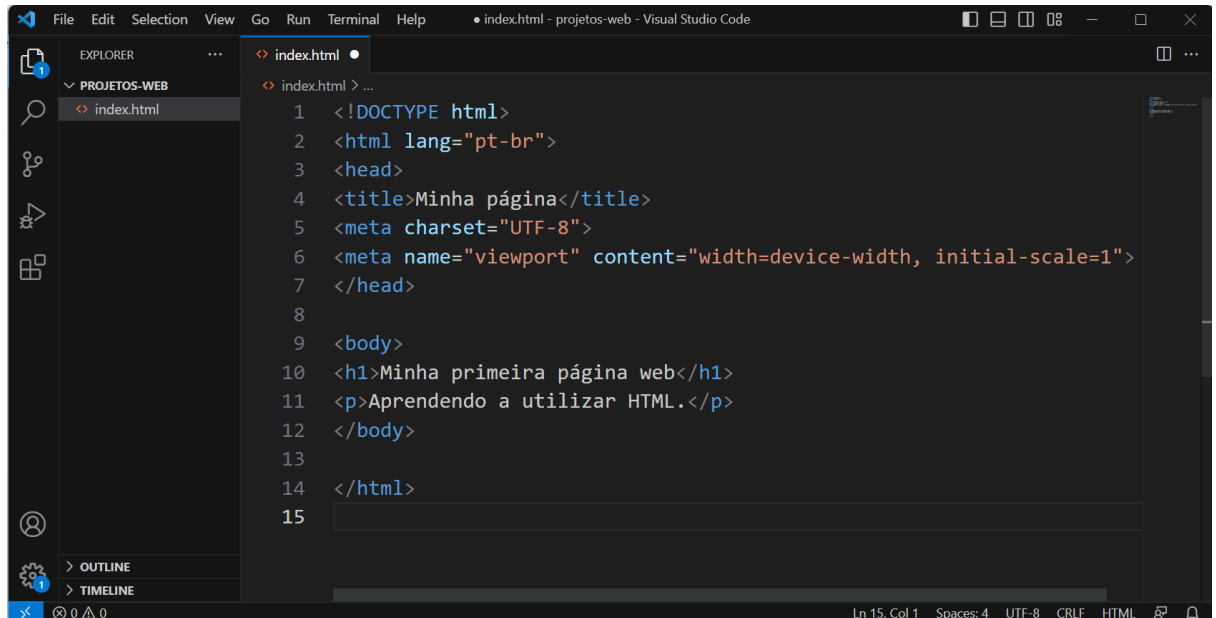
3. Selecione a opção “File >> New file” e crie um arquivo texto com o nome “index.html”, igual a figura a seguir:



## Curiosidades!

O **index.html** é o arquivo padrão (*default*) de um site, porque o navegador reconhece esse nome como padrão. Dessa forma, caso o projeto de seu site tenha mais de uma página, o navegador vai procurar o **index.html** para apresentar como a página inicial.

4. Dentro do arquivo **index.html**, iremos escrever o seguinte código HTML:

A screenshot of the Visual Studio Code editor interface. The Explorer sidebar on the left shows a project named 'PROJETOS-WEB' with a file named 'index.html' selected. The main editor area displays the content of 'index.html'. The code is as follows:

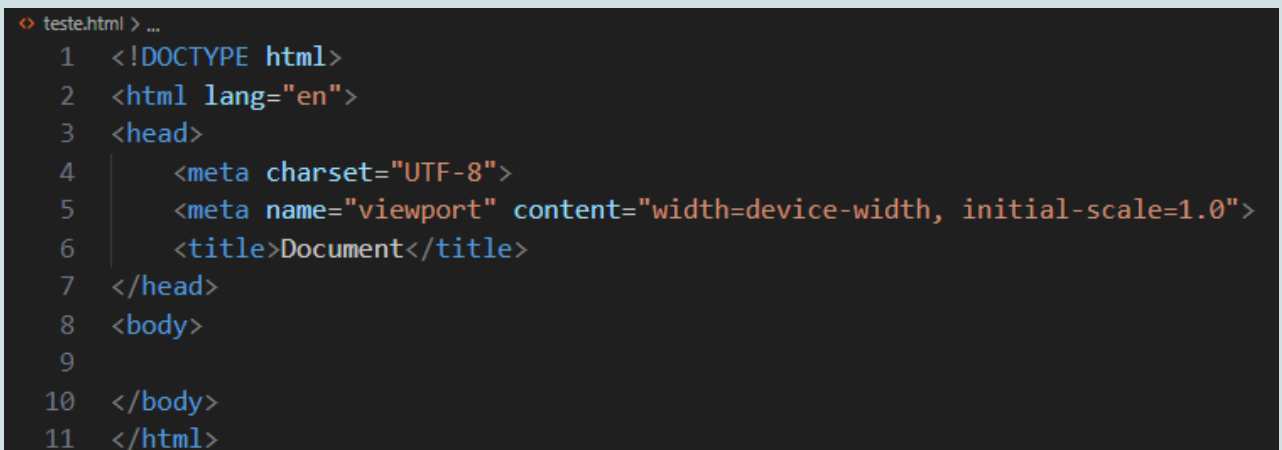
```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4 <title>Minha página</title>
5 <meta charset="UTF-8">
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7 </head>
8
9 <body>
10 <h1>Minha primeira página web</h1>
11 <p>Aprendendo a utilizar HTML.</p>
12 </body>
13
14 </html>
15
```

The status bar at the bottom indicates 'Ln 15, Col 1', 'Spaces: 4', 'UTF-8', 'CRLF', and 'HTML'.

Verifique que o VS Code já reconhece o conteúdo HTML e faz uma formatação de acordo com o tipo de cada conteúdo, facilitando o desenvolvimento.

## Para saber mais!

O VS Code possui um atalho interessante para a construção de arquivos HTML. Ao criar um arquivo com a extensão **.html**, você pode iniciar o arquivo com um ponto de exclamação (!) e, ao clicar na tecla **enter**, ele irá criar um arquivo base contendo as tags essenciais do **<head>** e a criação da tag **<body>**, como a figura abaixo.

A screenshot of the Visual Studio Code editor interface showing a file named 'teste.html'. The code is as follows:

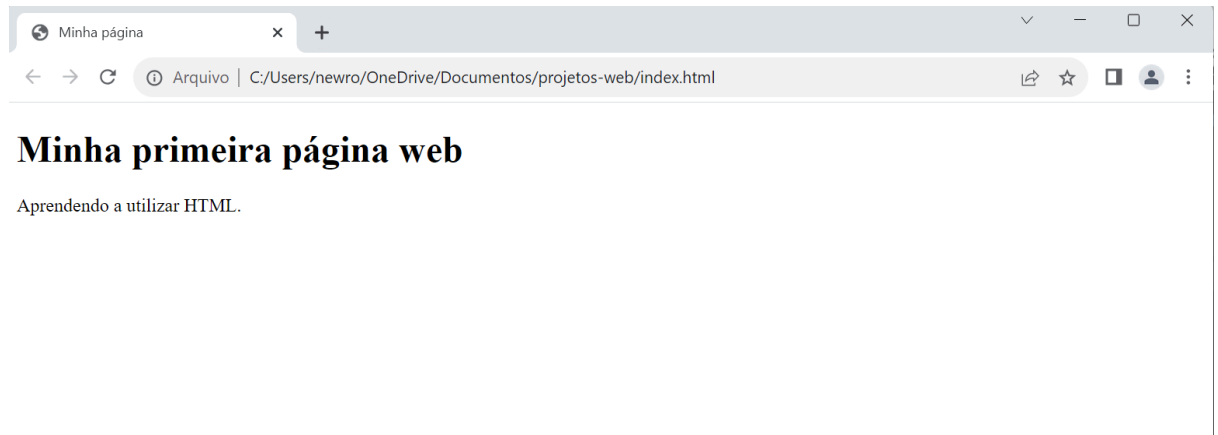
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9
10 </body>
11 </html>
```

Muito bom, não é mesmo?

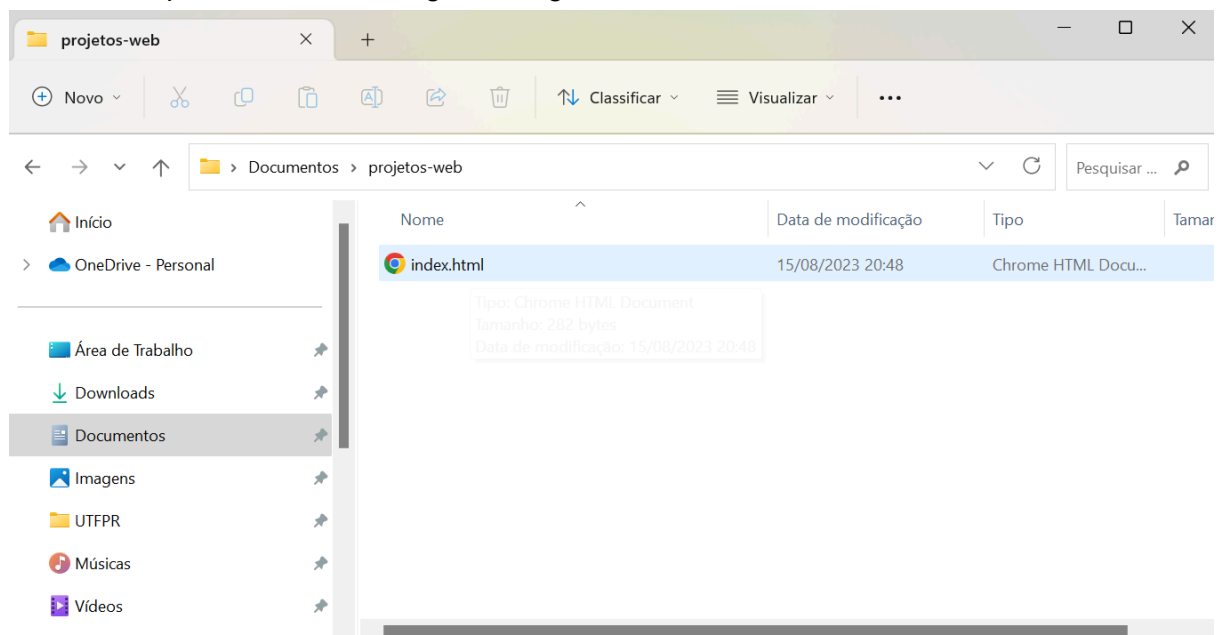
Lembre-se somente de alterar o valor do atributo **lang** para **pt-br**.

## Passo 4 - Abrir o arquivo .html em um navegador

1. Salve o arquivo no VS Code, selecione a opção “Run >> Run Without Debugging” e em seguida selecione a opção “Web App (Chrome)”. Essa ação deverá abrir o navegador Chrome, contendo o conteúdo da página index.html desenvolvida, conforme a figura a seguir:



2. Além dessa opção, você também poderá abrir o arquivo index.html no navegador pelo Windows Explorer, conforme a figura a seguir:

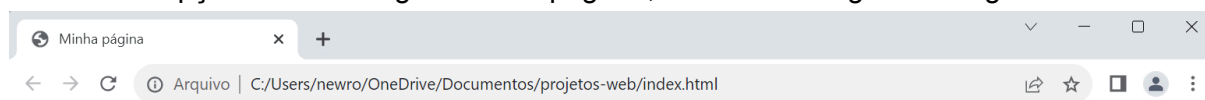


3. Ao fazer alguma alteração, salve o arquivo e o execute novamente, para verificar os resultados.



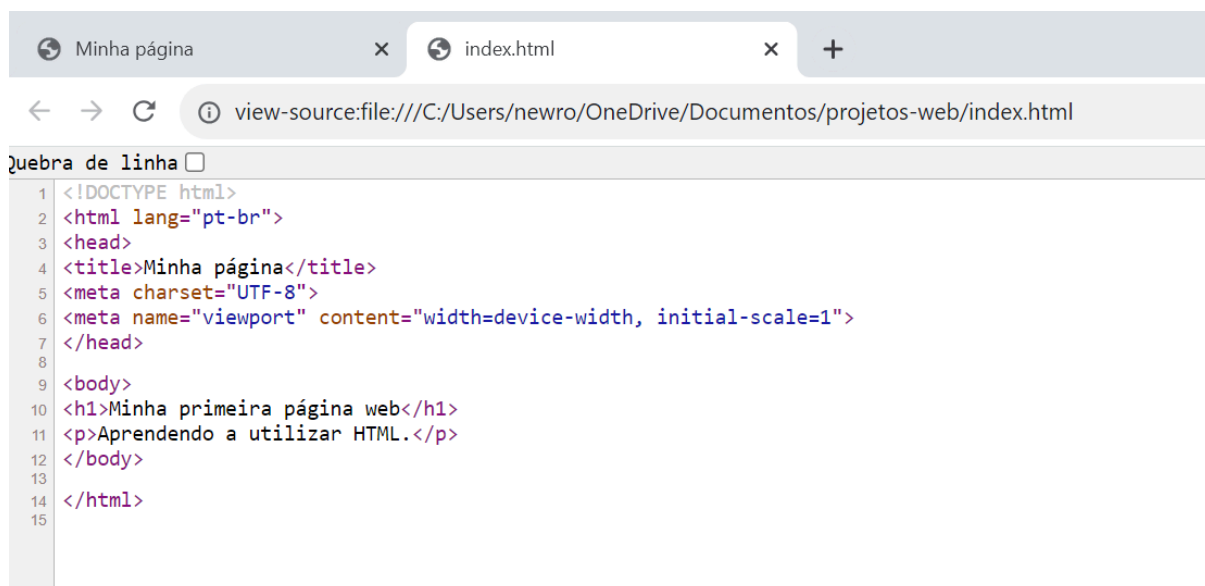
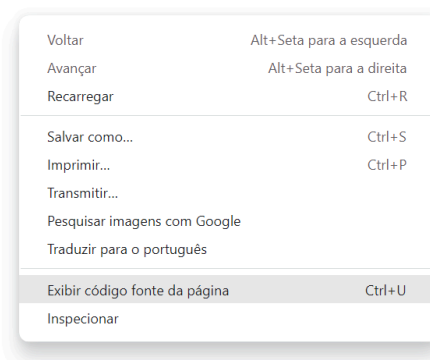
## Passo 5 - Visualizar o código-fonte de uma página em um navegador

1. Um recurso interessante dos navegadores é que eles oferecem a possibilidade de visualizar o código fonte de uma página. Por exemplo, para visualizar o código fonte da página desenvolvida, clique com o botão direito do mouse em algum lugar da sua página e selecione a opção “Exibir código fonte da página”, conforme as figuras a seguir:



### Minha primeira página web

Aprendendo a utilizar HTML.

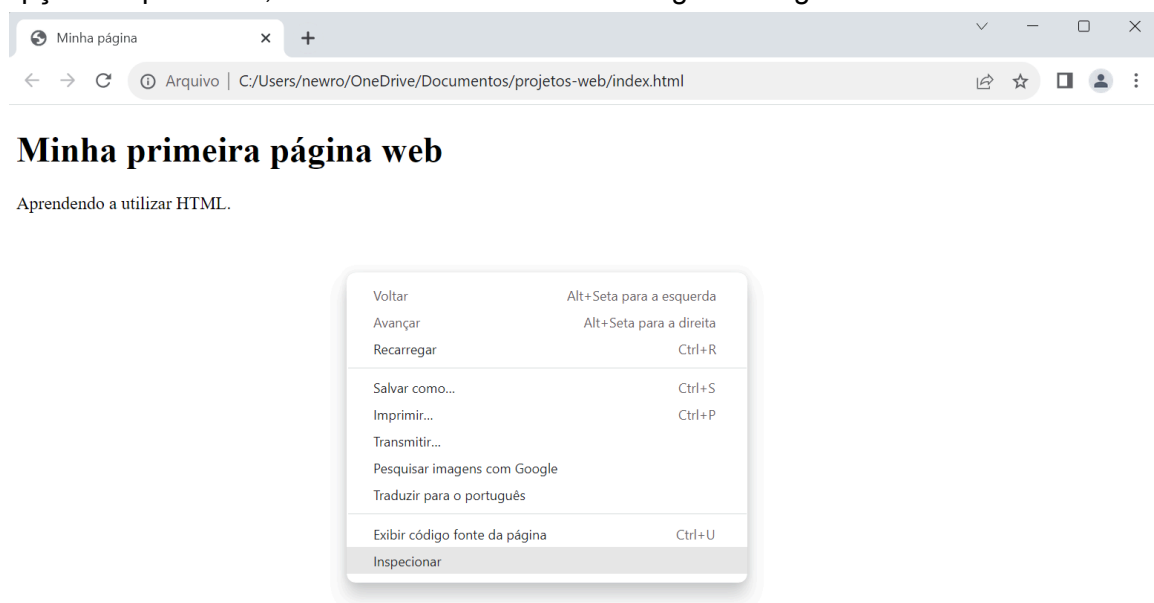


**OBSERVAÇÃO:** note que estamos utilizando neste tutorial o navegador Google Chrome. Caso utilize outro, pode ocorrer algumas variações em relação à forma de acesso. Porém, normalmente os comandos utilizados são parecidos.

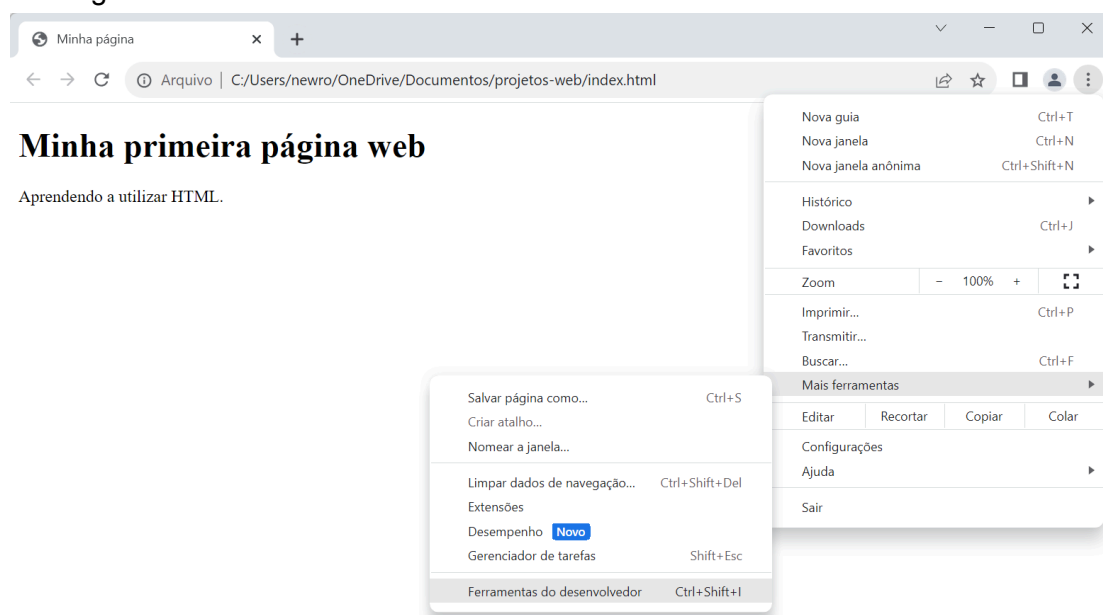
2. Note que é possível utilizar esse recurso não somente nas páginas desenvolvidas por você, mas em qualquer página html que você acesse em seu navegador. Faça o teste em alguma outra página, como, por exemplo, no site Ideia Circular (<https://ideiacircular.com/>).

## Passo 6 - Inspeccionar a página em um navegador

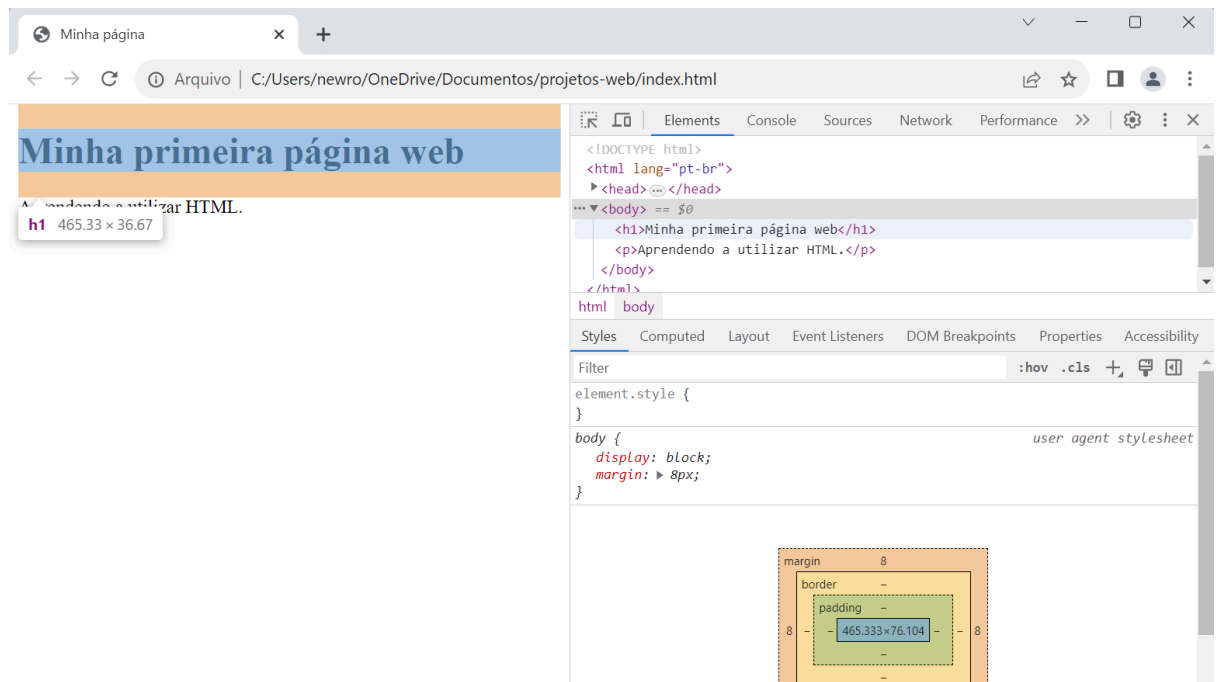
1. Uma outra funcionalidade muito útil é a de inspecionar a página. Nessa opção, o navegador oferece diversos recursos que podem ser utilizados durante o desenvolvimento de uma página web, tais como: inspecionar o conteúdo da página (HTML), os estilos adotados (CSS) e os recursos utilizados (JavaScript), alterar temporariamente o conteúdo da página, visualizar dados armazenados localmente, uso de memória, entre outras opções que veremos no decorrer da disciplina. Esse recurso será muito útil para você diagnosticar um problema no decorrer do desenvolvimento. Mas, por enquanto, vamos somente identificar como é possível inspecionar o conteúdo HTML.
2. Clique em algum lugar da página desenvolvida com o botão direito do mouse e selecione a opção “Inspecionar”, abrindo o conteúdo similar a figura a seguir:



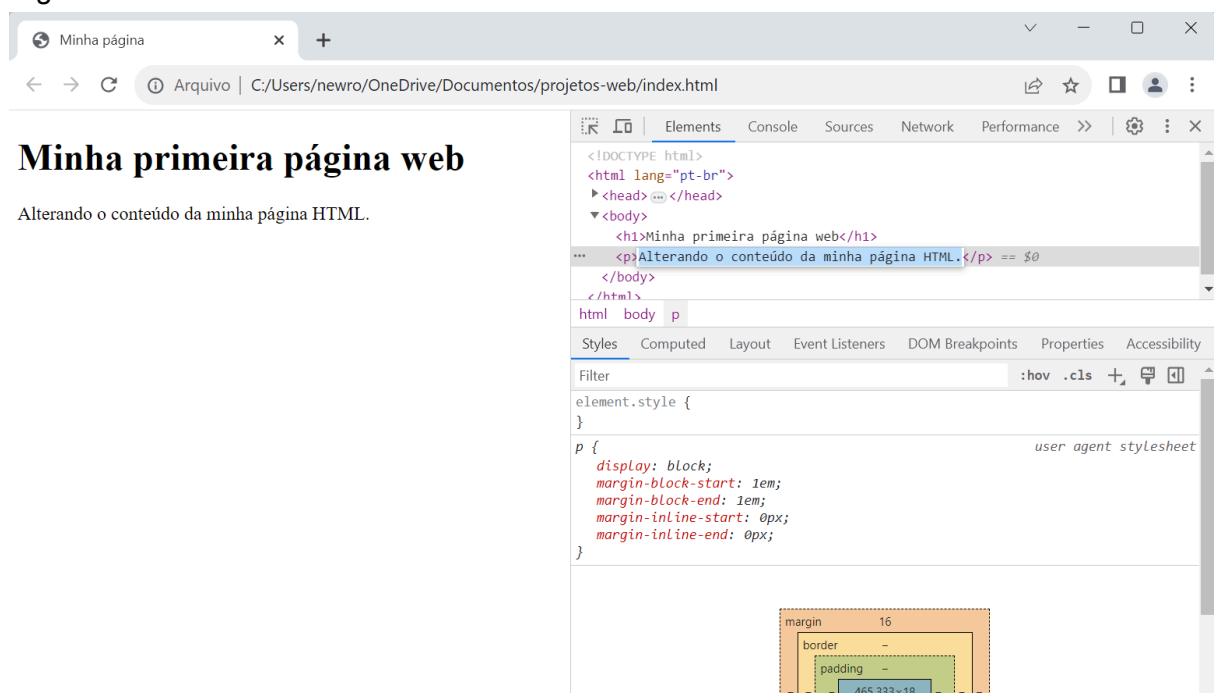
3. Uma outra forma de abrir essa opção é por meio dos 3 pontinhos na lateral direita do navegador, na opção “Mais ferramentas >> Ferramentas do desenvolvedor”, conforme a figura a seguir.



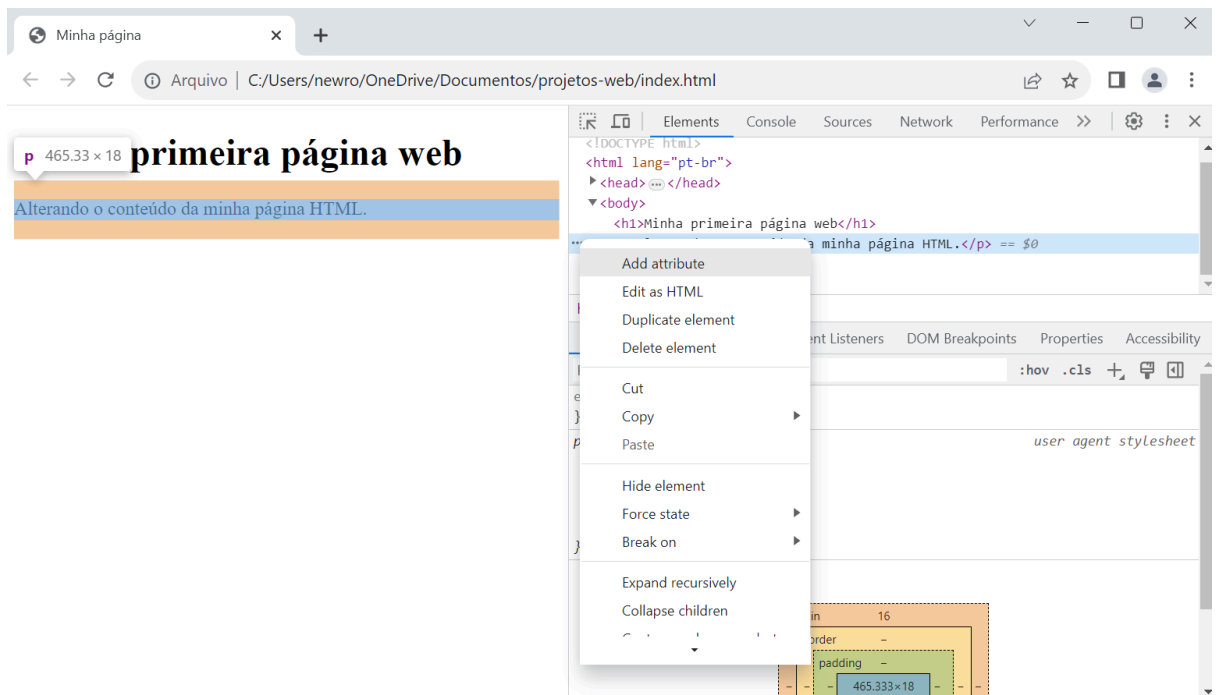
4. Veja que será aberto o conteúdo da página conforme a figura a seguir. Do lado esquerdo, é apresentado o conteúdo da página desenvolvida, contendo informações conforme você passa o mouse sobre os elementos. Ao lado direito estão as ferramentas de inspeção, como, por exemplo, a de identificar as características de cada elemento da página e sua configuração de estilo. Veja que ao passar o mouse no código `<h1>`, é apontado onde esse conteúdo está em sua página, à esquerda.




5. Conforme foi mencionado, você pode simular alterações na página diretamente por esse recurso, incluindo novas tags, alterando textos, etc. Para exemplificar, vamos alterar o conteúdo textual dentro da tag `<p>`, diretamente por esse recurso, conforme a figura a seguir:

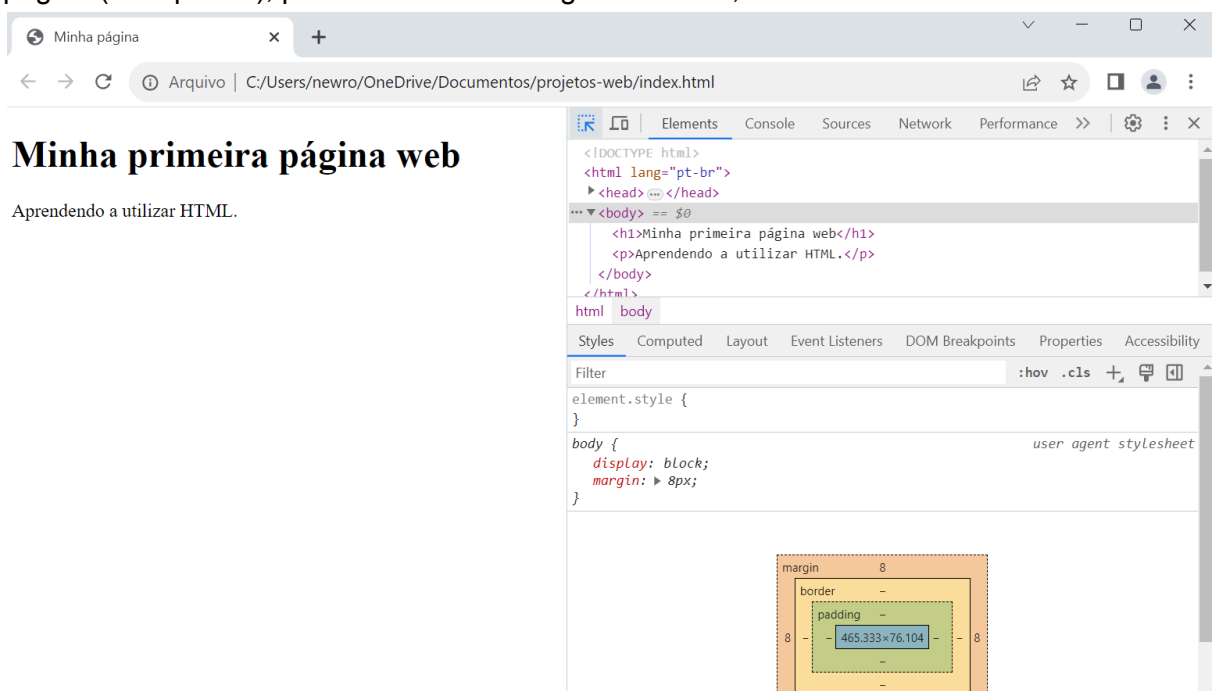



6. Veja que a alteração foi aplicada e já pode ser vista em tempo real, no conteúdo da página à esquerda. Entretanto, conforme foi mencionado, essas alterações são momentâneas e serão descartadas ao fechar a página.
7. Verifique também que é possível fazer outras alterações, tais como criar e editar atributos por meio da seleção dos `...` no início da tag, conforme a figura a seguir.

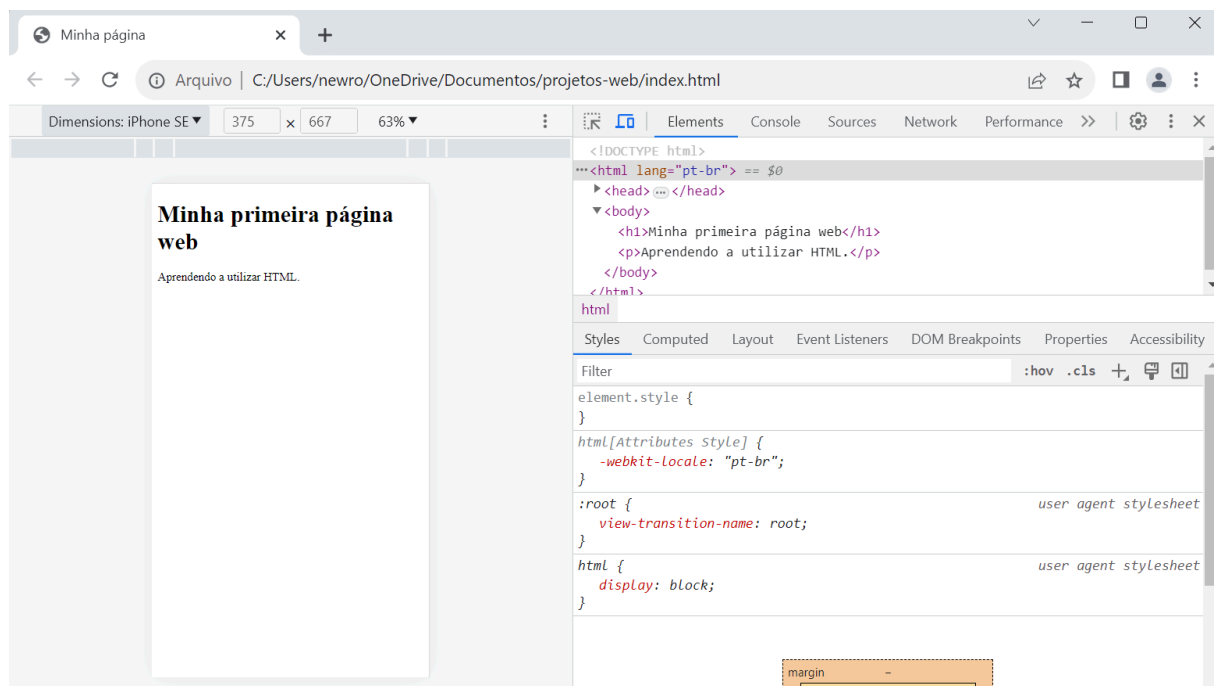


8. Para explorar os recursos, faça alguns testes duplicando o elemento, excluindo e alterando-o. Caso queira voltar ao resultado original, basta atualizar novamente a página.

9. Note também que você pode selecionar a opção  para selecionar o elemento da sua página (à esquerda), para identificar o código-fonte dela, à direita.




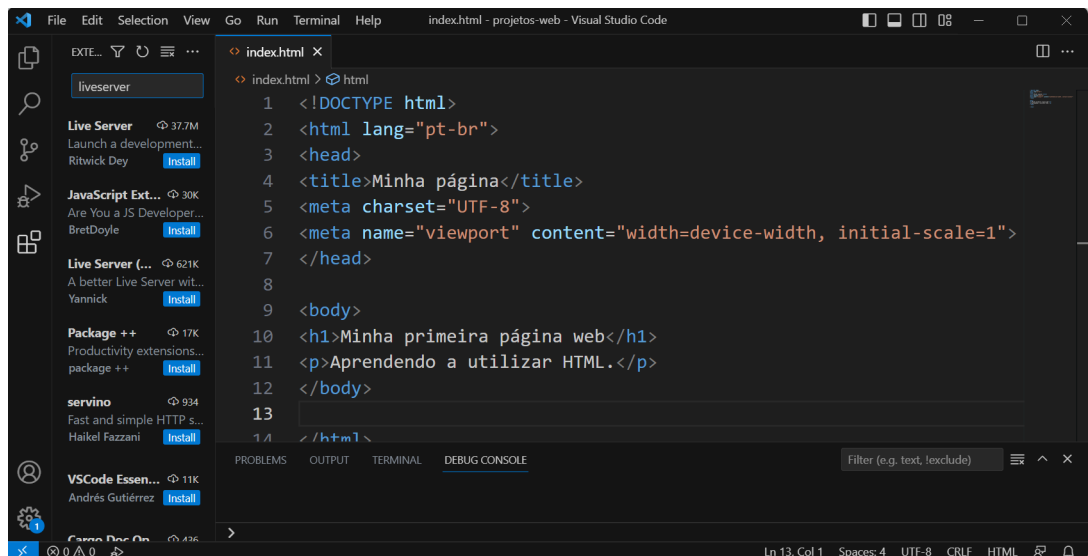
10. Uma outra opção útil pode ser acessada pela opção , que permite a visualização da página em diferentes formatos de tela, permitindo que você verifique se a página está adequada, se adaptando adequadamente a diferentes dispositivos.



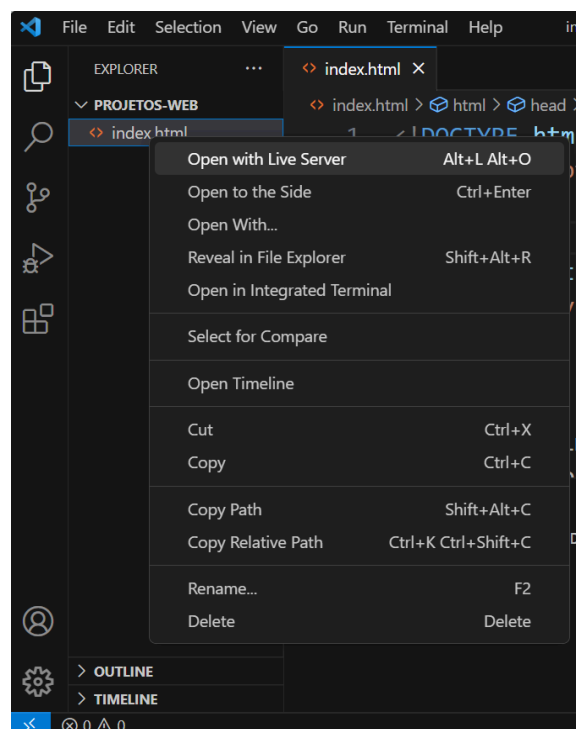
11. Veja que há diversos outros recursos disponíveis nesse utilitário, tais como: visualizar console, rede, desempenho, layout, etc. Caso tenha curiosidade, navegue por essas opções, mas não se preocupe se não compreendê-las agora. Elas serão aprendidas no decorrer da disciplina.

## Passo 7 - Utilizar o Live Server no VS Code

1. Live Server é uma extensão do VS Code, permitindo a criação de um servidor HTTP para a execução de páginas HTML. A partir de seu uso, é possível ter acesso a um serviço chamado *Live Reload*, na qual a alteração de um arquivo é automaticamente aplicada na página que está sendo executada.
2. Acesse novamente o VS Code e selecione o ícone Extensions , na lateral esquerda da IDE. Essa ação irá abrir uma caixa de busca, na qual você deve digitar “live server”, conforme a figura a seguir:



3. Selecione a primeira opção, para instalar o live server em sua máquina.
4. Após instalado, você pode clicar com o botão direito do mouse sobre o arquivo “index.html” e selecionar a opção “Open with Live Server”, conforme a figura a seguir:



5. Veja que essa ação irá abrir a página desenvolvida a partir de um servidor local criado, conforme a figura a seguir. Assim, qualquer mudança realizada no seu código será automaticamente atualizada em sua página.



# Minha primeira página web

Aprendendo a utilizar HTML.

6. Por fim, verifique que o Live Server sempre criará um servidor local no endereço `http://127.0.0.1:5500` ou `http://localhost:5500`. Dessa forma, caso você inicie o Live Server e o navegador não abra a página automaticamente, basta você digitar esse endereço em seu navegador que a página-raiz do projeto (`index.html`) será carregada.
7. Para continuar testando os recursos do HTML, altere o código HTML incluindo 10 tags em sua página. Você pode utilizar algumas dos exemplos a seguir:
- `<h1> </h1>` - Título de nível 1
  - `<h2> </h2>` - Título de nível 2
  - `<h3> </h3>` - Título de nível 3
  - `<h4> </h4>` - Título de nível 4
  - `<h5> </h5>` - Título de nível 5
  - `<h6> </h6>` - Título de nível 6
  - `<p></p>` - tag de texto, incluindo um parágrafo;
  - `<b></b>` - Transforma o conteúdo em negrito;
  - `<img>` - tag para inclusão de uma imagem
    - Exemplo: ``
  - `<footer></footer>` - Insere um rodapé

## Para saber mais!

Gostou de utilizar HTML e do conceito da Web? Caso queira ver mais sobre como Tim Berners-Lee pensou sobre a criação da Internet e também como ele considera que a Internet deveria ser agora, assista este TED Talk:

[Tim Berners-Lee e a próxima Web](#)

---

## Considerações finais

Espero que tenha conseguido realizar todas as atividades descritas, pois elas serão essenciais para os novos conteúdos que serão abordados na disciplina.

Caso tenha curiosidade, além de conversar com a professora, sugiro que vá explorando o conteúdo sobre HTML por meio dos tutoriais disponíveis no site da W3Schools. Para isso, sugiro executar os seguintes exemplos:

- [Básico de HTML](#)
- [Elementos HTML](#)
- [Atributos HTML](#)

Bom estudo!