



FUNDAMENTOS DA PROGRAMAÇÃO *WEB*

AULA 1



Prof.^a Margarete Klamas



CONVERSA INICIAL

Pode-se afirmar que HTML é essencial para qualquer linguagem que venha a se especializar futuramente para *web*. Saber HTML é fundamental.

O objetivo dessa abordagem é apresentar as *tags* básicas em HTML, que servem de base para o desenvolvimento de aplicações *web* para os mais variados fins. Ao concluir os cinco tópicos propostos, você será capaz de construir páginas e de fazer alterações em páginas criadas por outros desenvolvedores.

No tópico 1, abordaremos a história da criação do HTML, termos utilizados no mercado e arquitetura cliente/servidor. No tópico 2, iremos mostrar o ambiente de desenvolvimento. Utilizaremos, ao longo do nosso estudo, o Visual Studio Code. No tópico 3, iniciamos com HTML: *tag*, estrutura da página HTML, títulos, parágrafos, quebra de linha, linha horizontal e listas. No tópico 4, estudaremos *links*, imagens e tabelas. No tópico 5, abordaremos formulários.

No desenvolvimento das explicações, você irá encontrar *QR Codes* com vídeos que contêm exemplos, além das videoaulas.

HTML é estrutural, e nesta abordagem não faremos formatações.

TEMA 1 – CONCEITOS INICIAIS

1.1 História

A sigla HTML significa *HyperText Markup Language*. Em português, linguagem de marcação de hipertexto, a linguagem de marcação que abriu um leque de possibilidades e extrapolou as possibilidades inicialmente imaginadas para seu uso e desenvolvimento. HTML é o coração da web.

A essência do HTML é o HyperText (hipertexto), podemos colocar numa página texto, imagens, sons e vídeos, além dos links que permitem que façamos uma leitura não linear.

Pessoas visionárias propuseram, por meio de modelos teóricos, ideias que inspiraram o desenvolvimento do HTML. Vannevar Bush foi uma dessas pessoas, com o Memex, em 1945. Tratava-se de uma máquina que tornava possível fazer *links* entre documentos.



Saiba mais

MEMEX animation – Vannevar Bush's diagrams made real. **Mark Sanderson**, 2009. Disponível em: <<https://www.youtube.com/watch?v=c539cK58ees>>. Acesso em: 7 dez. 2022.

Posteriormente, em 1960, Theodor (Ted) Nelson criou o termo hipertexto/hipermídia. Apresentou o projeto Xanadu, cuja ideia era disponibilizar textos paralelos. Com essa proposta, é possível visualizar documentos e suas conexões precisas, utilizando duas janelas.

Saiba mais

Quer saber mais sobre o projeto do Ted Nelson? Use o QR Code a seguir:
THE XANADU® Parallel Universe – Visibly connected pages and documents for a new kind of writing. Xanadu, S.d. Disponível em: <<https://xanadu.com/xUniverse-D6>>. Acesso em: 7 dez. 2022.

Tim Bernard Lee, em 1990, desenvolveu o HTML, com o objetivo de trocar informações entre pesquisadores. Ele não tinha ideia da dimensão que tomaria a sua criação.

Berners-Lee (citado por Chagas, 2014) afirmava: “Gostei da ideia de que uma informação é realmente definida apenas pelo que está relacionada e como está relacionada. Realmente há pouco mais para o significado. A estrutura é tudo”.

HTML teve como base o SGML (*Standard Generalized Mark-up Language*), um método acordado internacionalmente para marcar texto. O uso de elementos como <title> e </title> foi retirado diretamente do SGML, e o mesmo ocorreu com elementos como parágrafos, títulos, listas e outros. O elemento para *link* e o formato `www.name.name` para endereços de máquina na web foram inventados por Tim.

HTML sofreu oito evoluções até o momento dessa pesquisa. A versão atual é HTML5.

1.2 Termos utilizados atualmente

Nas buscas por vagas, você irá encontrar termos como *front-end*, *back-end* e *full-stack*, mas afinal o que são esses termos?



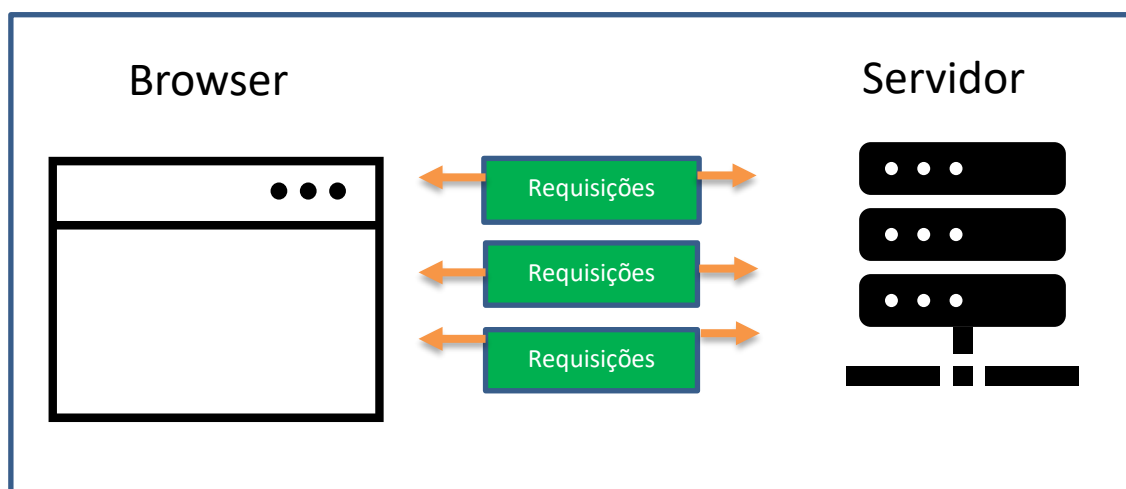
Vamos começar com *front-end*, termo usado para fazer referência à interface gráfica, ou aparência dos aplicativos, a parte que o usuário irá interagir. *O foco dos nossos estudos é front-end*, e abordaremos HTML, CSS e JavaScript.

Back-end é um termo utilizado para fazer referência ao funcionamento das aplicações, como conectar com banco de dados, servidores, lógica de negócio, ou seja, a camada oculta, não vemos, mas ela está lá. *Back-end* pode utilizar diversas linguagens como Java, PHP, .NET, C#, RUBY, JavaScript e outras, fica como sugestão pesquisar.

Full-stack é o termo utilizado quando envolve as tecnologias *front-end* e *back-end*.

1.3 Arquitetura cliente versus servidor

Quando acessamos o endereço de um *site* no *browser*, este vai buscar o endereço do site digitado num servidor de hospedagem. Pode-se dizer que temos dois lados: o *cliente* e o *servidor*. Computadores conectados à *web* são chamados de clientes e servidores. Observe a imagem a seguir:



Cliente pode ser qualquer dispositivo conectado à internet utilizando o browser ou navegador.

Servidores são computadores que armazenados os sites, com seu conjunto de páginas ou aplicativos. Quando o browser de um cliente solicita uma página (HTML), uma cópia dela é baixada do servidor para o dispositivo do cliente. Essa é uma parte da história de ferramentas e funcionamento da internet.

O que permite ocorrer essa comunicação é protocolo HTTP, que significa *Hypertext Transfer Protocol*, e é usado para gerenciar as requisições (cliente) e



as respostas (servidor) pela internet. O HTTP requer que os dados sejam transferidos de um ponto para outro pela rede (Galhego, 2022).

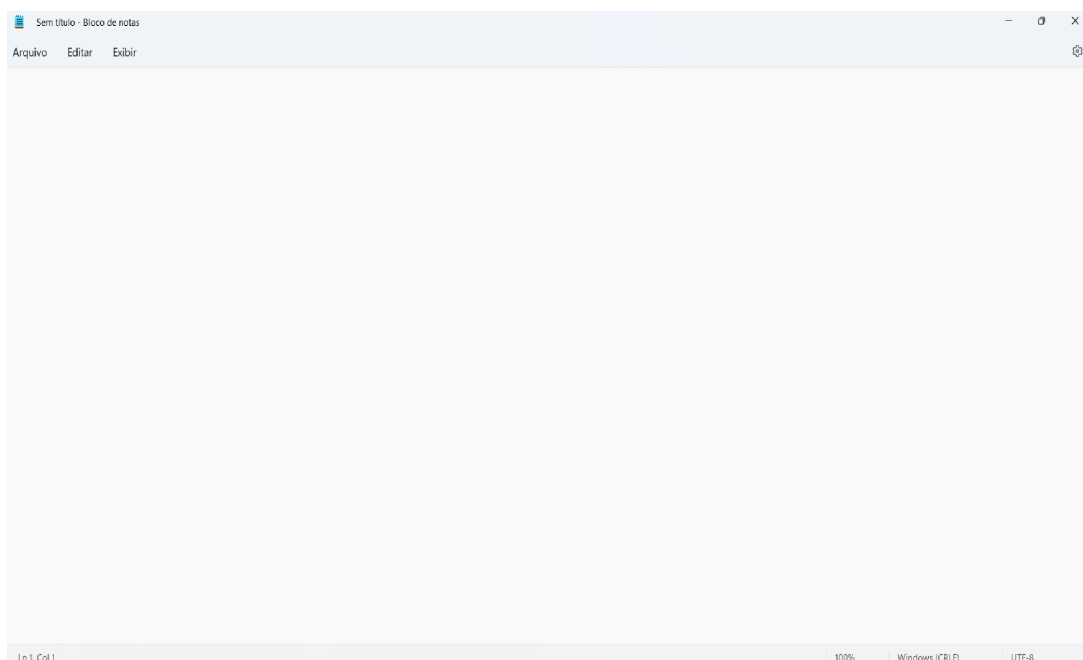
TEMA 2 – AMBIENTE DE DESENVOLVIMENTO

Para criarmos as nossas páginas HTML, necessitamos utilizar programas que permitem fazer o desenvolvimento das páginas. Há diversos programas disponíveis no mercado, desde os mais simples até programas mais intuitivos. Veremos, a seguir, as ferramentas de desenvolvimento HTML para o ambiente Windows.

2.1 Bloco de Notas (Notepad)

O Bloco de Notas oferece funcionalidades mínimas de edição, mas suficientes para codificação HTML. Para abrir o bloco de notas, basta pressionar a tecla Windows+R e digitar *notepad*.

Figura 1 – Descrição da Imagem: Tela do Bloco de notas aberto.



Após a digitação do código, basta digitar o nome do arquivo.html entre aspas: “nome_do_arquivo.html”.



2.2 Notepad++

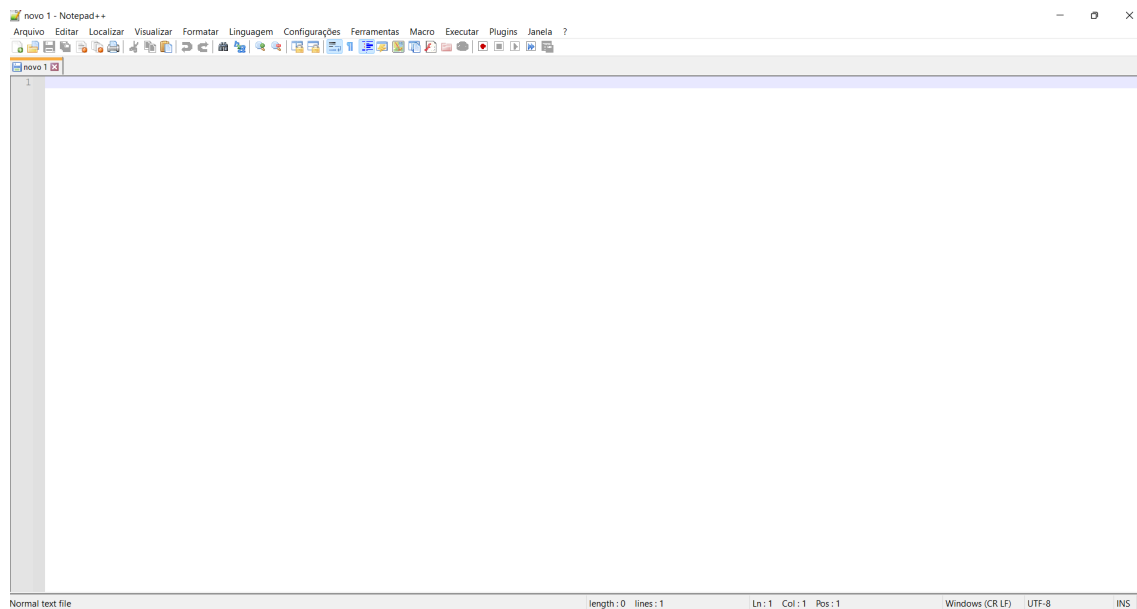
Trata-se de um editor HTML gratuito. Possui mais funcionalidades que o Bloco de Notas.

Saiba mais

NOTEPAD. Disponível em: <<https://www.notepad-plus-plus.org/>>. Acesso em: 7 dez. 2022.

O processo de instalação é simples, basta avançar.

Figura 2 – Descrição da imagem: Tela do NotePad++



Antes de iniciar seu trabalho, faça as seguintes configurações:

1. Menu Linguagem >>H>>> HTML
2. Formatar >> Codificação em UTF-8

2.3 Sublime Text

Criado por Jon Skinner, um engenheiro de *software* que trabalhou para o Google. O objetivo era criar um editor minimalista, com interface intuitiva, funcional e rápida.

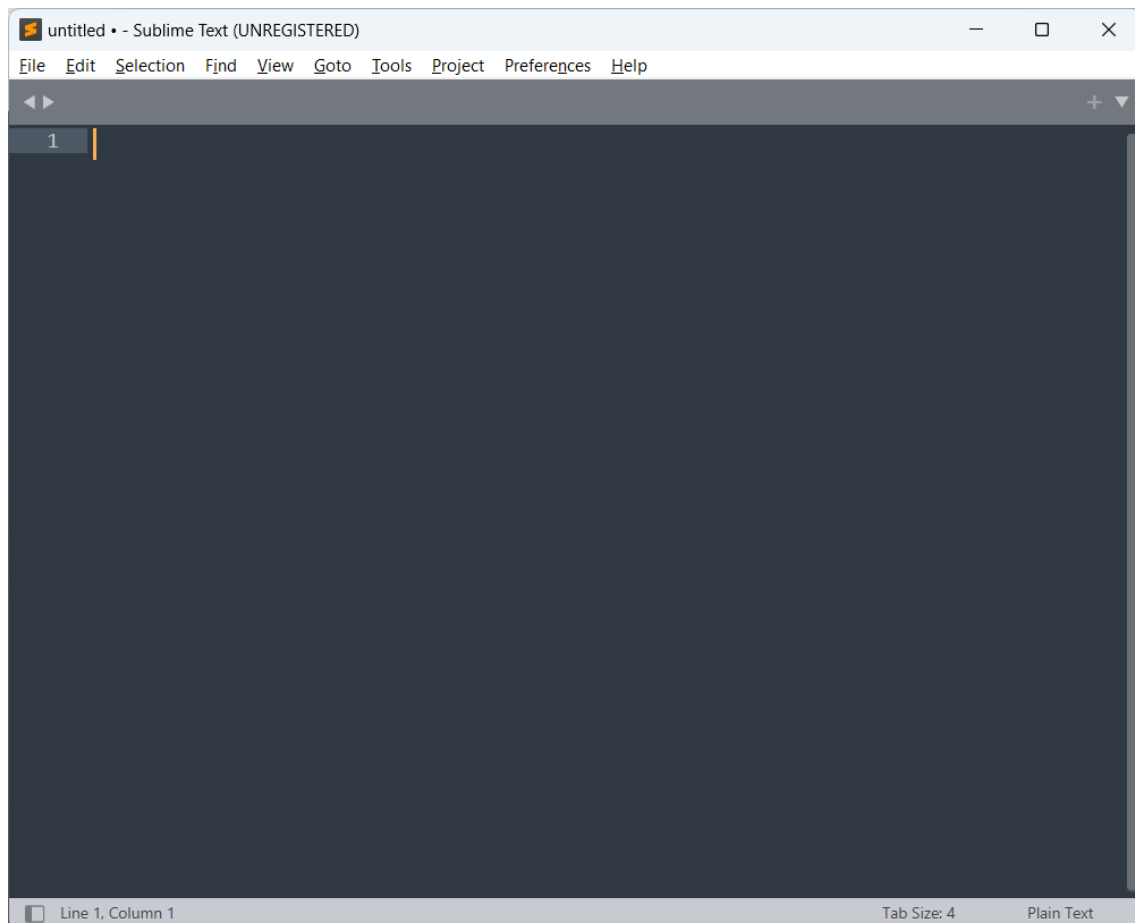


Saiba mais

SUBLIME TEXT. Disponível em: <<https://www.sublimetext.com/>>. Acesso em: 7 dez. 2022.

Como todo processo de instalação, basta avançar e instalar.

Figura 3 – Tela do Sublime Text



Fonte: Sublime Text, S.d.

Para utilizar o Sublime Text, crie uma pasta no Explorer. No Sublime Text:

1. File
2. Open Folder > Selecionar a pasta que você criou
3. File > New File
4. Salvar o arquivo.html (dar um nome para o arquivo).html
5. Na área de digitação do código digitar: HTML e pressionar a tecla ENTER.



2.4 Visual Studio Code

O Visual Studio Code é um ambiente de desenvolvimento poderoso, projetado para facilitar a escrita de aplicativos *web*, móveis e em nuvem usando linguagens que estão disponíveis para diferentes plataformas de desenvolvimento e para suportar o ciclo de vida de desenvolvimento de aplicativos com um depurador integrado e com suporte ao popular controle de versão Git. Com o Visual Studio Code, você pode trabalhar com arquivos de código individuais ou com sistemas de arquivos estruturados baseados em pastas.

O Visual Studio Code foi a primeira ferramenta de desenvolvimento multiplataforma da família Microsoft Visual Studio que roda em Windows, Linux e macOS. É gratuito, de código aberto e é definitivamente uma ferramenta centrada em código, o que facilita a edição de arquivos de código e sistemas de projetos baseados em pastas, com suporte para várias linguagens e recursos avançados de edição, como IntelliSense.

Saiba mais

MICROSOFT. VS Code. Disponível em: <<https://github.com/Microsoft/vscode>>. Acesso em; 7 dez. 2022.

Combina a simplicidade de um poderoso editor de código com as ferramentas que um desenvolvedor precisa para dar suporte ao desenvolvimento do ciclo de vida do aplicativo, incluindo depuradores e integração de controle de versão com base no Git. É, portanto, uma ferramenta de desenvolvimento completa, ao invés de ser um simples editor de código.

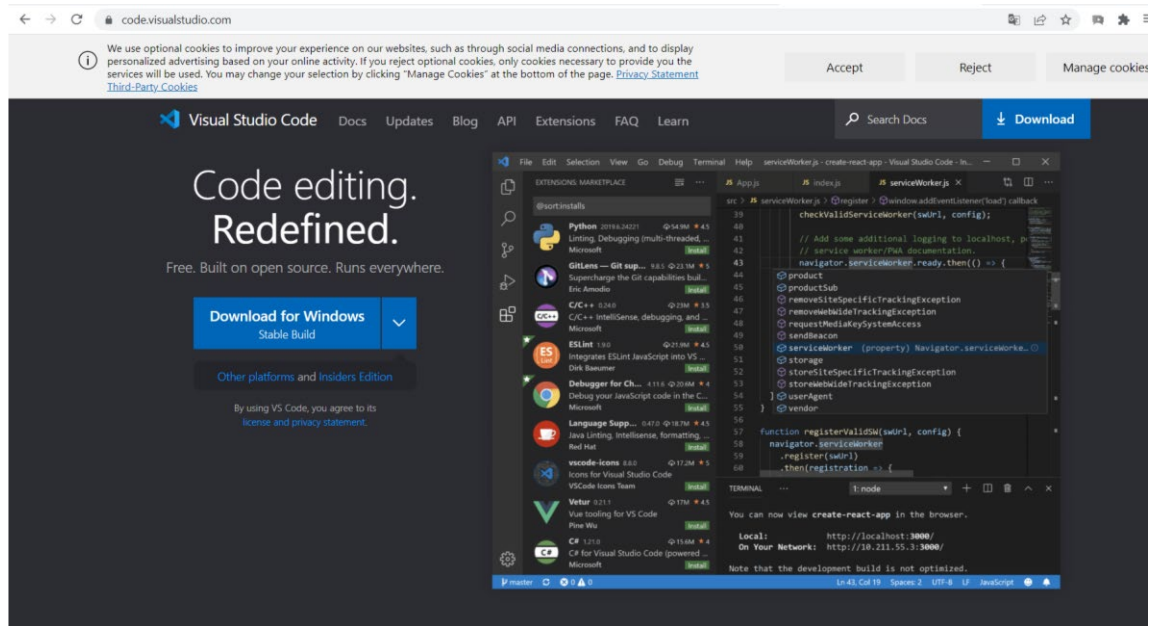
Saiba mais

Para baixar o Visual Studio Code, acesse o *link* a seguir:

VISUAL STUDIO CODE. Disponível em: <<https://code.visualstudio.com/>>. Acesso em: 7 dez. 2022.



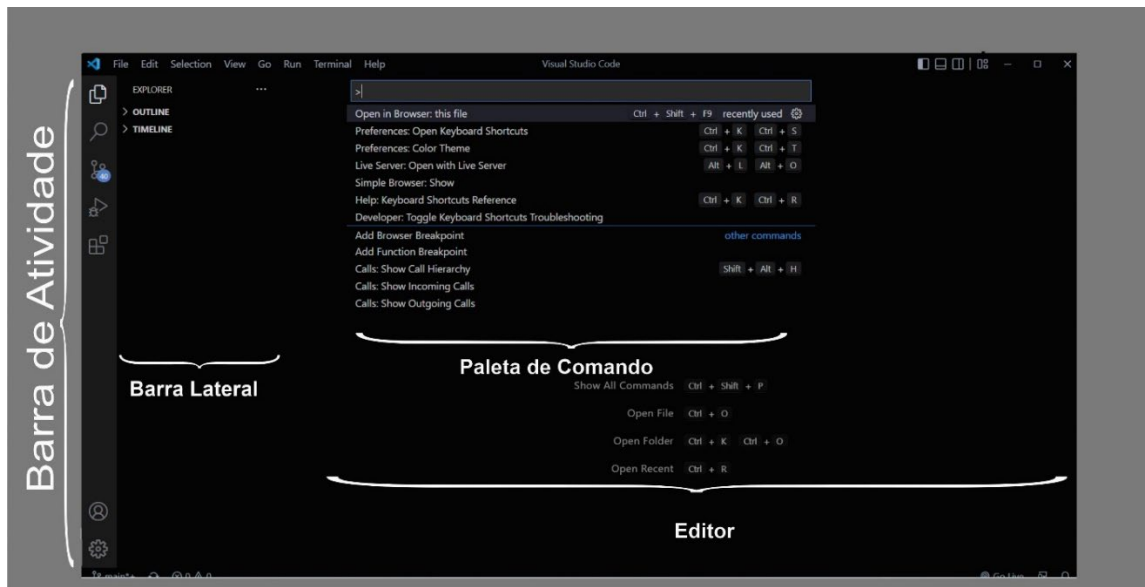
Figura 4 – Tela inicial do *site* para baixar o Visual Studio Code



Fonte: Visual Studio Code, S.d.

A seguir, a tela do Visual Studio Code:

Figura 5 – Tela do Visual Studio Code. Lado esquerdo superior da tela: Barra de Atividades. Da esquerda para a direita: Barra Lateral. Lado direito da tela: Editor de códigos. Para abrir a Paleta de Comandos: teclas CTRL+SHIFT+P



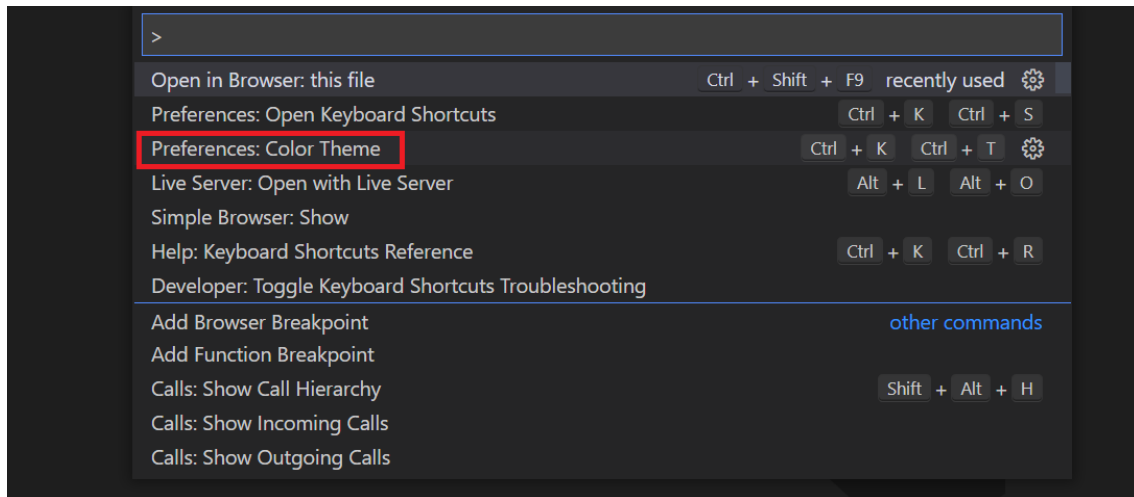
Talvez você deseje alterar a aparência da tela (cores) para melhor visualização. O comando que permite mudar o tema é Color Theme:

Pressione as teclas: CTRL+SHIFT+P >> Preferences: Color Theme.

Basta você testar as opções e escolher a que mais lhe agrada.

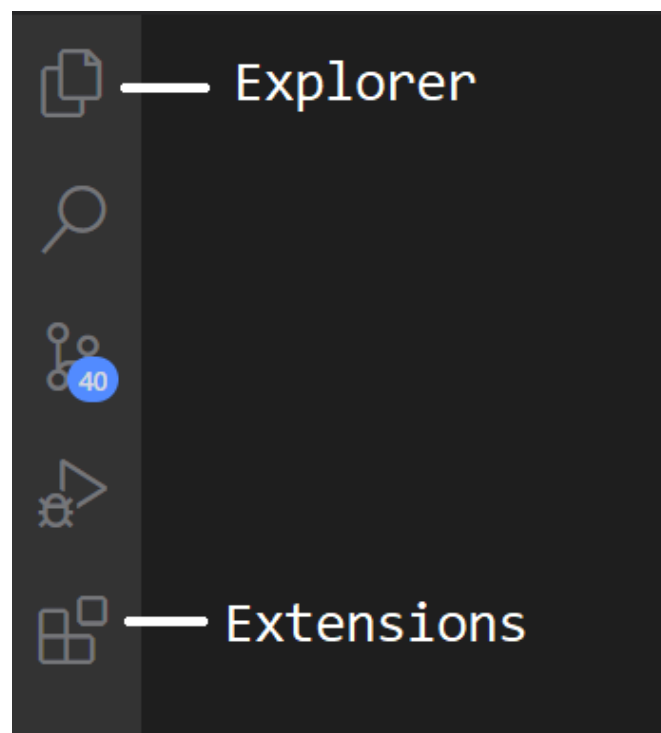


Figura 6 – Imagem das opções de *Preferences*



A Barra de Atividades contém dois botões importantes, os quais iremos utilizar com mais frequência: o botão Explorer e o botão Extensions.

Figura 7 – Barra de Atividades. 1º ícone: Explorer. 5º ícone: Extensions



A barra de extensões (*extensions*) pode ser habilitada clicando no quinto botão na parte superior da barra de atividades e permite pesquisar e instalar extensões para o Visual Studio Code, que incluem idiomas adicionais, depuradores, trechos de código e muito mais.

O botão Explorer mostra a pasta selecionada e os arquivos contidos.



2.5 Como começar?

1. Vamos instalar primeiramente algumas extensões.

Basta clicar no botão *Extensions*, digitar o nome das Extensões abaixo e clicar no botão *Install*.

- Atom One Dark Theme

Uma opção a mais para você conhecer para tema.

- Winter is Coming Theme

Uma opção a mais para você conhecer para tema.

- Material Icon Theme

Disponibiliza ícones para o Explorer, inclusive para pastas.

- vscode-icons

Oferece temas e recursos.

- Open in browser

Permite visualiza o arquivo html no *browser* e ao lado.

- Live Server

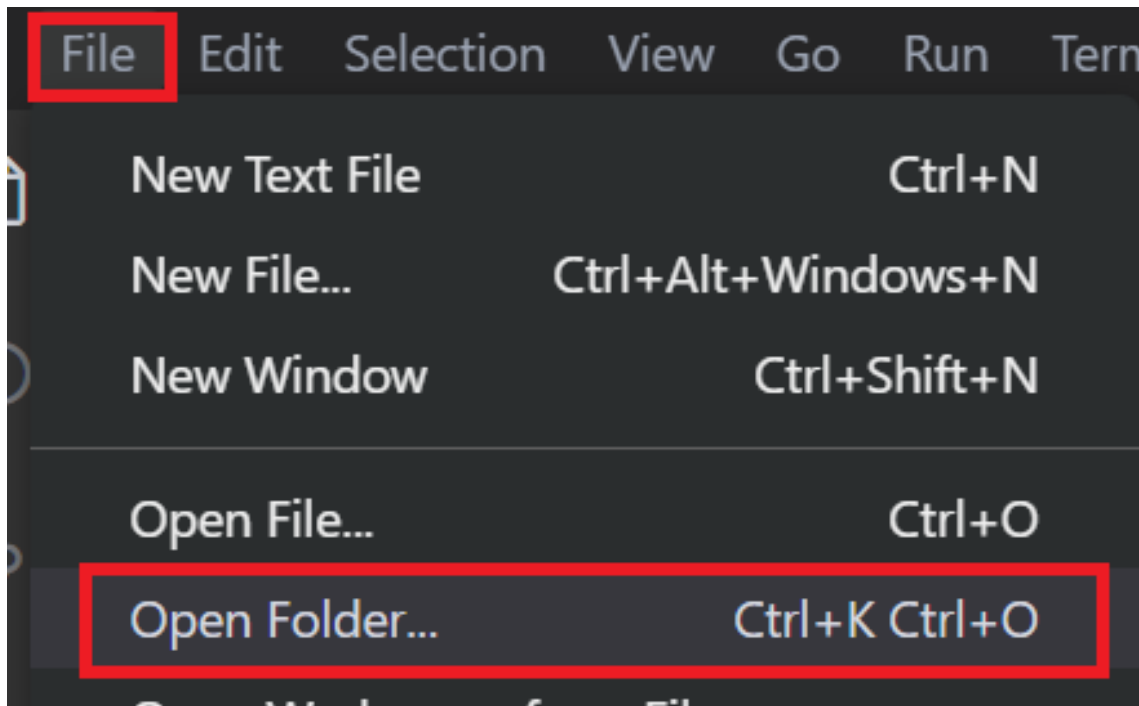
Atualiza automaticamente as alterações realizadas.

2. Criar uma pasta para salvar os exercícios no Windows Explorer.

3. Abrir essa pasta no Visual Studio Code.



Figura 8 – Menu File escolher a opção *Open Folder*. Selecionar a pasta que você criou



4. Criar o 1º Arquivo em HTML:

Figura 9 – Após ter selecionada a pasta. 1. Ícone New File (Novo arquivo). 2. Digitar o nome do arquivo.html

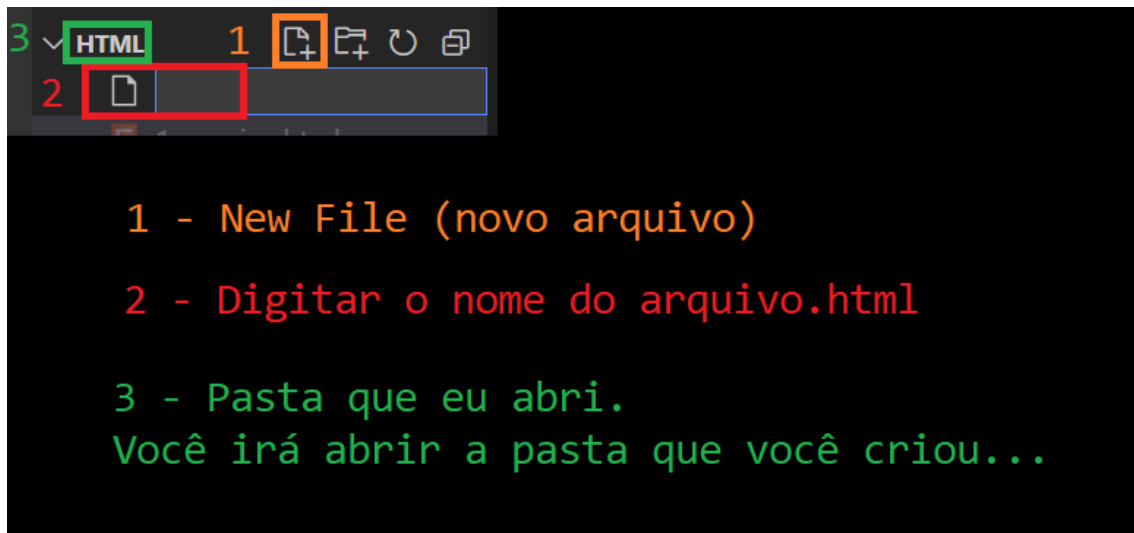
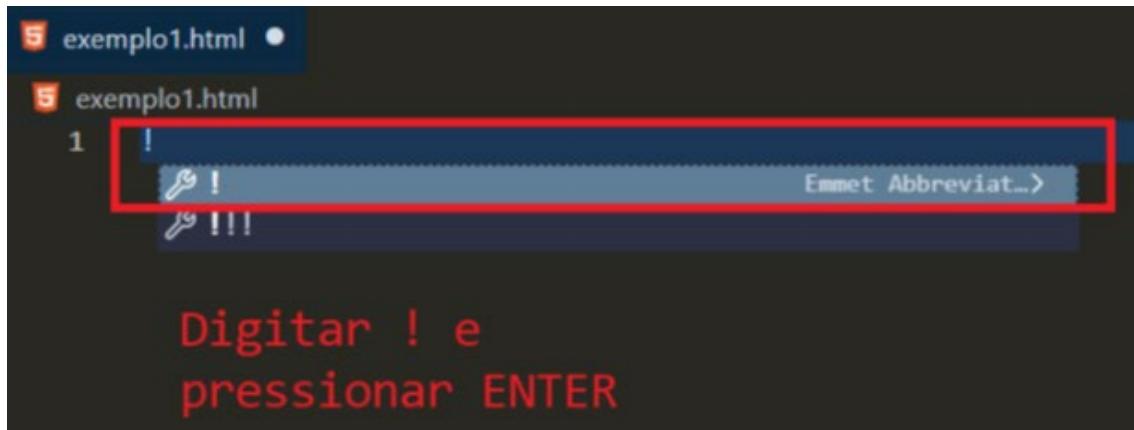




Figura 10 – Na linha 1, digitar ! (ponto de exclamação) e pressionar ENTER



Ficará assim:

```
<!DOCTYPE html>
<html lang="en">
<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>
```

Basta pressionar CTRL+S para salvar o arquivo. Fechar o arquivo. Fechar o Visual Studio Code.

TEMA 3 – HTML

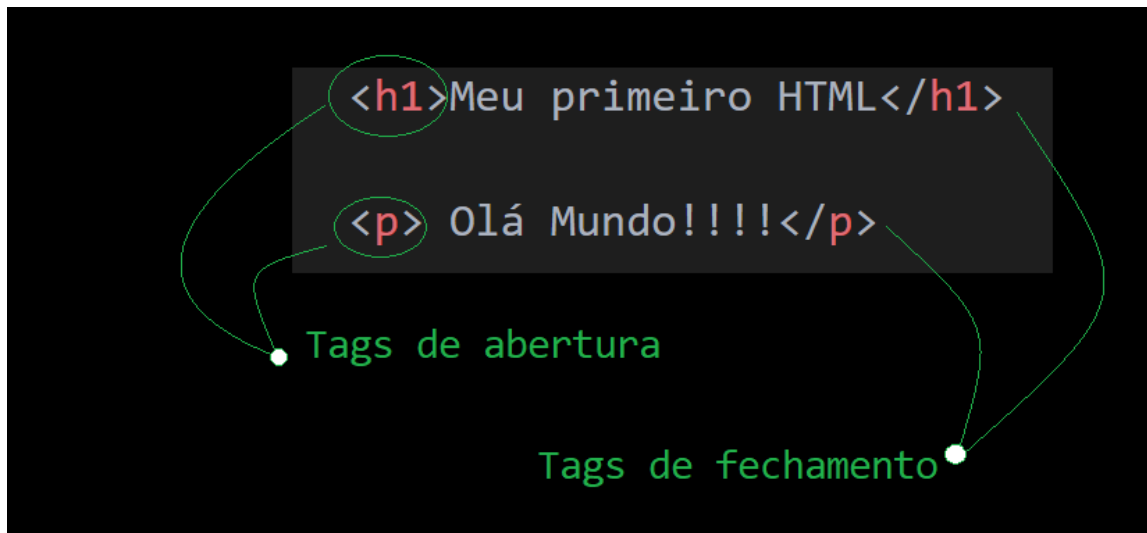
HTML é uma linguagem de marcação que o desenvolvedor usa para se comunicar com o *browser* (navegador), que por sua vez interpreta a marcação em algo que o usuário entenda, como uma página *web*. Como qualquer linguagem, HTML também tem suas regras de sintaxe e escrita. O produto de uma marcação HTML é um arquivo de texto que deve ser gravado com a extensão *.html*, por exemplo *nome_arquivo.html*.



3.1 Tag HTML

Para escrever em HTML, precisamos utilizar *tags* (etiquetas). Iniciamos com uma *tag* de abertura e no fim uma *tag* de fechamento. Na Figura 11, mostramos a marcação de um cabeçalho e a seguir um parágrafo:

Figura 11 – Tags HTML



Descrição da imagem

Tags HTML

```
<h1>Meu primeiro HTML</h1>
```

```
<p> Olá Mundo!!!!</p>
```

<h1> e <p> tags de abertura

</h1> e </p> tags de fechamento

Uma *tag* HTML deve ser escrita entre os sinais < e >. A *tag* de fechamento é igual à *tag* de abertura com um sinal de barra / na frente. Existem várias *tags* HTML, que serão estudadas nesta abordagem, cada uma delas destinada a marcar um determinado tipo de conteúdo. No exemplo mostrado na imagem acima, usamos as *tags* <h1></h1> e <p></p> que se destinam a marcar cabeçalhos de nível 1 e parágrafos, respectivamente. Alguns autores utilizam o termo elemento HTML para *tags*.

3.2 Estrutura mínima de uma página HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
```



```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1.0">
<title>Document</title>
</head>
<body>

</body>
</html>
```

Quadro 1 – Explicação

<code><!DOCTYPE html></code>
Em HTML, obrigatoriamente temos que começar com a chamada declaração do tipo de documento (<i>doctype</i>). Essa linha informa ao <i>browser</i> (que é quem vai ler e interpretar o que está escrito na página) que a marcação que se segue é do tipo HTML5. Não se trata de marcação e sim de declaração.
<code><html lang="en"></code>
Especifica o idioma da página, “en” refere-se ao inglês. Podemos alterar para “pt-br”.
<code></html></code>
Fecha o documento página HTML.
<code><head> e </head></code>
O elemento <code><head></code> contém informações gerais (metadados) sobre o documento, como seu título, <i>links</i> para <i>scripts</i> e folhas de estilos.
<code><meta charset="UTF-8"></code>
Charset: define a codificação de caracteres usada na página. UTF-8 (Unicode Transformation Format 8): é o formato recomendado para ser utilizado. Pode representar qualquer caractere Unicode padrão.
<code><meta name="viewport" content="width=device-width, initial-scale=1.0"></code>
Meta <i>tag</i> que se refere a responsividade que iremos tratar mais adiante.
<code><title>Document</title></code>



Tags de fechamento e abertura do elemento *title* (título). Esse é o elemento mais importante de um documento. Ele marca o título do documento. O título deve ser escolhido com atenção, pois ele é identificado pelos mecanismos de busca e indexação, como o Google (SILVA, 2016).

<body> e </body>

Tags de fechamento e abertura do elemento body. Dentro do <body> são admitidos vários outros elementos HTML. Tudo o que estiver nessa seção será renderizado pelo navegador e mostrado para o usuário. Ou seja, é o conteúdo das páginas que visualizamos.

3.3 HTML – Títulos, parágrafos, quebra de linha e listas

HTML é uma linguagem de marcação, similar a um editor de texto. É absolutamente voltada a semântica e ao uso das *tags*, nesta seção iremos abordar as marcações textuais básicas em HTML.

3.3.1 Títulos (*Heading*)

As tags de títulos são definidas de <h1> a <h6>, e são empregadas para definir os cabeçalhos em HTML. Eles são do tipo nível de bloco.

<h1> define o título mais relevante.

<h6> define o título menos importante.

Utilize apenas um <h1> por página.

No tópico 2.4 recomendamos criar uma pasta, abrir essa pasta no Visual Studio Code, e salvar todos os arquivos dentro dessa pasta. No item 4 do tópico 2.4, você viu como criar um novo arquivo. Vamos criar um arquivo novamente?

1. Criar um arquivo: títulos.html



Figura 12 – Após ter selecionada a pasta (*Open Folder*) – 1. Ícone New File (Novo arquivo); 2. Digitar o nome do arquivo: títulos.html

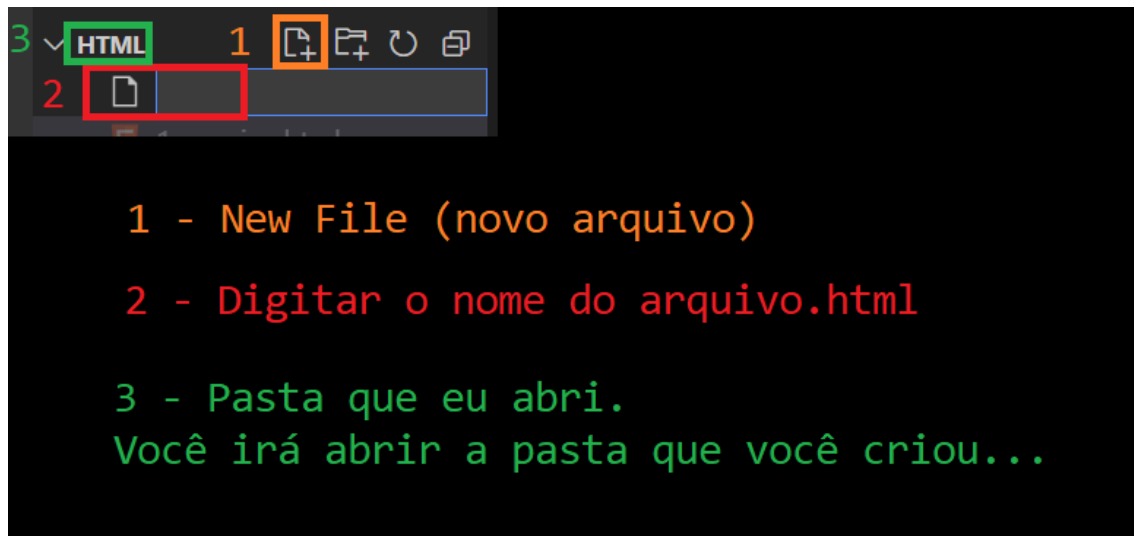
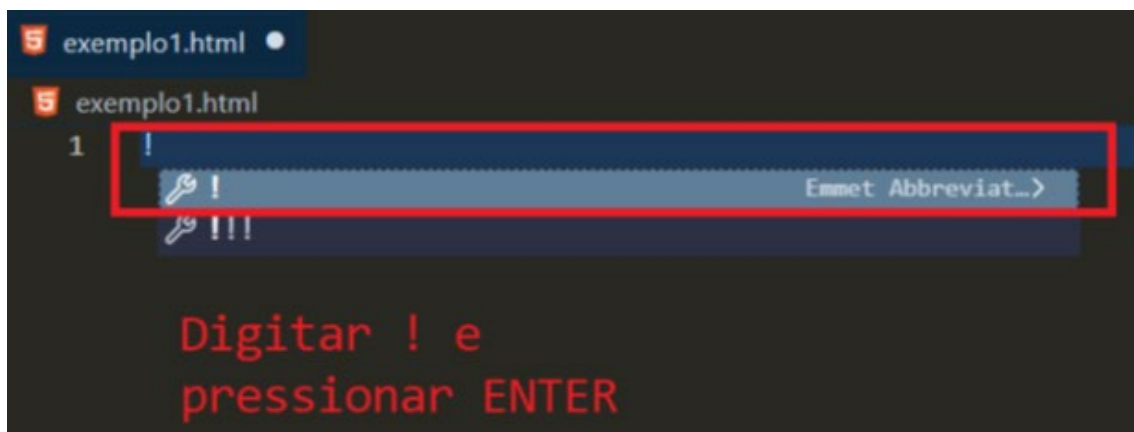


Figura 13 – Na linha 1, digitar ! (ponto de exclamação) e pressionar ENTER.



Digite o código abaixo entre *tags* `<body>` e `</body>`:

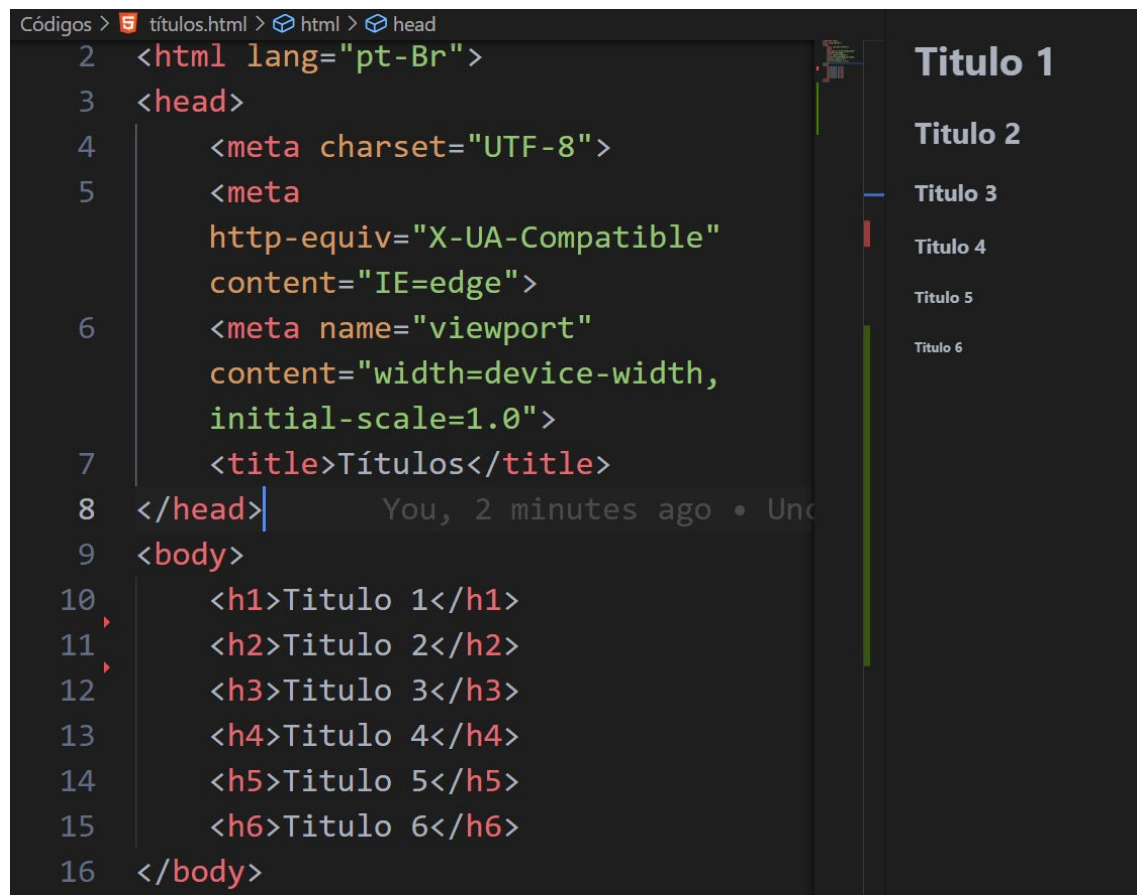
```
<!DOCTYPE html>
<html lang="pt-Br">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0"
  <title>Títulos</title>
</head>
<body>
  <h1>Titulo 1</h1>
  <h2>Titulo 2</h2>
  <h3>Titulo 3</h3>
  <h4>Titulo 4</h4>
```



```
<h5>Titulo 5</h5>
<h6>Titulo 6</h6>
</body>
</html>
```

Para visualizar o código, você pode digitar: **CTRL+SHIFT+P > HTML: Open Preview to the Side.**

Figura 14 – Visualização no VS Code



Descrição da imagem:

A imagem acima mostra a visualização no VS Code.

Lado esquerdo da tela: o código do lado direito como aparece na tela do navegador.

Na visualização, aparece somente o texto digitado entre as tags <h>.

Na tela aparecerá o conteúdo digitado entre as tags <body>.

Na barra de título do navegador, irá aparecer a palavra Títulos digitada entre a tag <title> e</title>.

Outra possibilidade é você digitar: **CTRL+SHIFT+P > HTML: Open in Browser.**



Figura 15 – Visualização do conteúdo na tela do browser. O mesmo conteúdo digitado entre as *tags* de título `<h>`. Digite o código e visualize das duas maneiras.

Imagem da Visualização no Browser

Título 1

Título 2

Título 3

Título 4

Título 5

Título 6

3.3.2 Parágrafos, quebras de linhas e linha horizontal

- A *tag* `<p>` define um parágrafo. Os navegadores adicionam uma linha após um parágrafo `</p>`.
- A *tag* `
` força uma quebra de linha. Não possui *tag* de fechamento.
- A *tag* `<hr>` insere uma linha horizontal. Não possui *tag* de fechamento. Eles elementos são do tipo nível de bloco.

Saiba mais

Vamos colocar em prática? Acesse o *link* a seguir, assista ao vídeo e implemente:

Estrutura básica de uma página HTML. CP. **Uninter**, S.d. Disponível em: https://n-cpuninter.github.io/FUNDAMENTOS-DE-DESENVOLVIMENTO-WEB/html/video_paragrafo.html. Acesso em: 7 dez. 2022.

3.3.3 Listas

Podem-se gerar dois tipos de listas:



Figura 16 – Na imagem temos duas listas: uma lista não ordenada que possui antes de cada item da lista um símbolo “bolinha”, e uma lista ordenada que possui antes de cada item da lista uma numeração sequencial: 1,2,3.

Listas não ordenadas

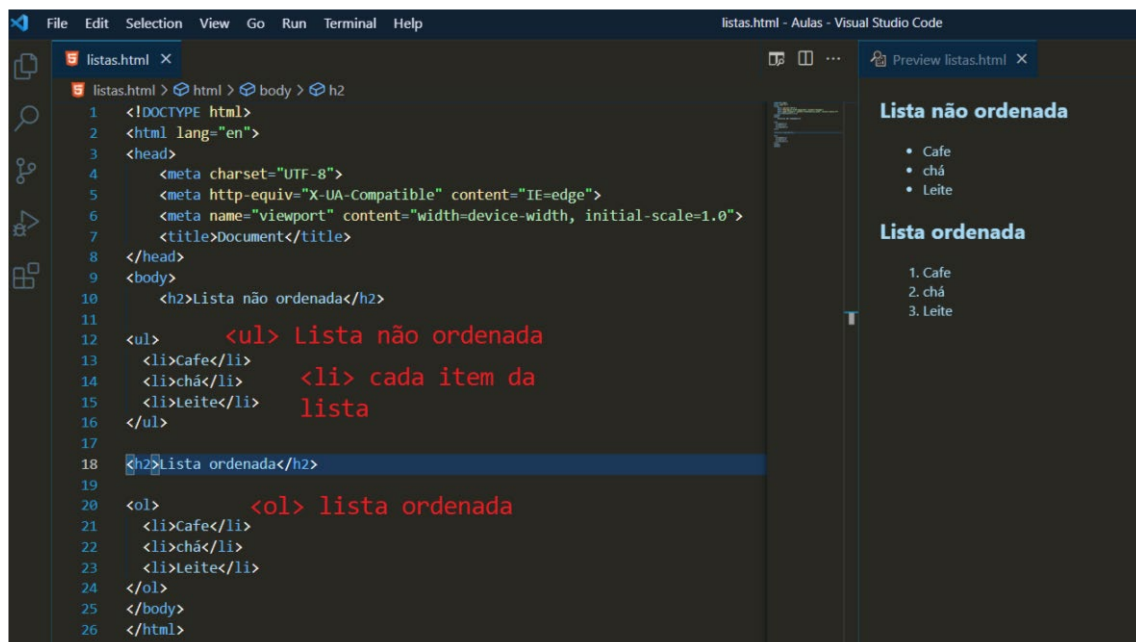
- HTML
- CSS
- JavaScript

Listas ordenadas

1. HTML
2. CSS
3. JavaScript

Digite o código da imagem:

Figura 17 – Lado esquerdo: código para você praticar. Lado direito: visualização no VS Code. Digitar o conteúdo que está entre as tags <body> e </body>.



Saiba mais

N-CP UNINTER. Fundamentos-de-desenvolvimento-web. **N-CP Uninter**, S.d. Disponível em: <<https://github.com/N-CPUninter/FUNDAMENTOS-DE->



DESENVOLVIMENTO-WEB/blob/main/txt-codigos/listas.txt>. Acesso em: 7 dez. 2022.

TEMA 4 – LINKS, IMAGENS E TABELAS

4.1 Links

Para inserir ligações (*hyperlinks*, *links*), utilizamos a tag `<a>`, elemento do tipo *inline*. O atributo `href` é necessário e indica o destino do *link*.

`Texto`

`<a>` - *anchor* elemento

`href` - *hyperlink reference* (referência de *hyperlink*)

Optativo:

`target` alvo (onde irá abrir o *link*). Valores:

`_self` - Predefinição. Abre o documento na mesma janela/guia em que foi clicado

`_blank` - Abre o documento em uma nova janela ou guia

`_parent` - Abre o documento no quadro pai

`_top` - Abre o documento em todo o corpo da janela

`Visite o Google`

Exemplo de *hyperlink* para você praticar:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Links</title>
</head>
<body>
  <h2>Links</h2>
  <a href="https://google.com.br">Google</a>
  <br>
```



```
<a href="listas.html">Listas</a>
<br><br>
<a href="imagens.html">Imagens</a>
</body>
</html>
```

Quadro 2 – Explicação

```
<a href="https://google.com.br">Google</a>
```

No exemplo acima, no atributo `<href>` estamos fazendo um *link* para o Google. Observe que colocamos o endereço completo do site.

```
<a href="listas.html">Listas</a>
```

```
<a href="imagens.html">Imagens</a>
```

No exemplo acima, no atributo `<href>` estamos fazendo um *link* para arquivos html que estão armazenados no mesmo local, onde está salvo o arquivo que contém o link.

4.2 Imagens

Para inserir imagens, utilizamos o elemento ``. Esse é um elemento *inline* e não necessita da *tag* de fechamento. O uso desse elemento exige no mínimo a inserção de dois atributos.

```

```

`` requer dois atributos: `src` (caminho da imagem) e `alt` (texto alternativo, se a imagem por algum motivo não puder ser exibida).

Outros atributos que podem ser utilizados são `width` (largura) e `height` (altura).

Exemplo:

```
<!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>Imagens</title>
</head>
```



```
<body>
  <h2>Imagens</h2>

  
  <br><br><br>
  
</body>
</html>
```

Saiba mais

N-CP Uninter. Fundamentos de desenvolvimento. **N-CP Uninter**, S.d. Disponível em: <<https://n-cpuninter.github.io/FUNDAMENTOS-DE-DESENVOLVIMENTO-WEB/html/imagem.html>>. Acesso em: 7 dez. 2022.

4.3 Tabelas

Utilizamos a tag `<table>` para tabelas, juntamente com os elementos `<tr>`, `<th>` e `<td>`.

```
<th>titulo</th>
```

```
<tr>cria linha</tr>
```

```
<td> cria coluna</td>
```

Exemplo de tabela:

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Tabelas</title>
</head>
<body>
  <h2>Tabelas</h2>
  <table width="80%" border="1px">
    <tr>
      <th>Aluno</th>
      <th>RU</th>
    </tr>
    <tr>
      <td>Ana Maria</td>
      <td>2131</td>
```



```
</tr>
<tr>
    <td>João Pedro</td>
    <td>4346</td>
</tr>

</table>
</body>
</html>
```

Quadro 3 – Explicação

table width="80%" border="1px">

width= largura – podemos trabalhar com porcentagem

border = 1px espessura da borda

<tr> e </tr>

<tr> cria uma linha

</tr> fecha a linha

<th> e </th>

<th> cria uma coluna para título

</th> fecha a coluna para título

<td> e </td>

<td> cria uma coluna

</td> fecha a coluna

Visualização da tabela no navegador

Tabelas


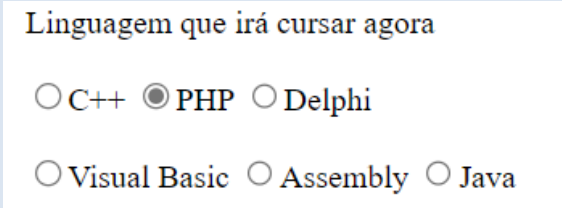
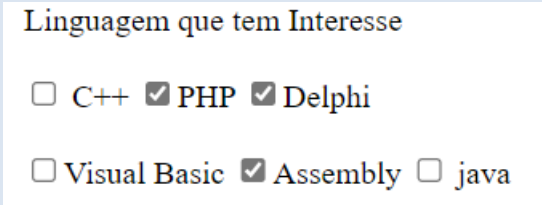
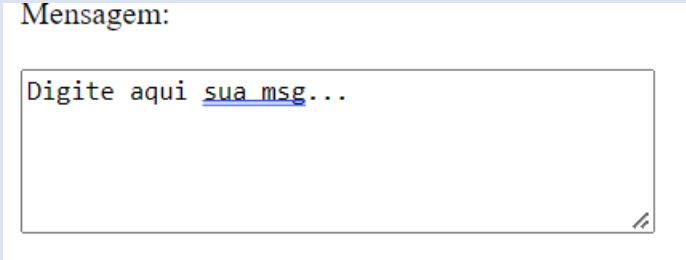
Aluno	RU
Ana Maria	2131
João Pedro	4346

TEMA 5 – FORMULÁRIOS

Utiliza-se a tag **<form>** para criar um formulário. O objetivo de utilizar formulários é receber entradas (informações) dos usuários. Esses dados podem ser enviados para um banco de dados, ou até mesmo recebidos no *e-mail*. Os campos devem estar entre as tags **<form>** e **</form>**:



```
<form>
  .
  elementos
  .
</form>
```

Elemento	Descrição
<code><input type="text"></code>	Exibe um campo de entrada de texto de linha única 
<code><input type="radio"></code>	Exibe um botão de opção (para selecionar uma das muitas opções) 
<code><input type="checkbox"></code>	Exibe uma caixa de seleção (para selecionar zero ou mais de muitas opções) 
<code><input type="button"></code>	Exibe um botão clicável
<code><textarea></code> <code></textarea></code>	Insere uma área de texto para o usuário digitar textos. 
<code><select></code> <code></select></code>	Cria uma lista suspensa. Os <code><option></code> definem uma opção que pode ser selecionada. Por padrão, o primeiro item na lista suspensa é selecionado. Para definir uma opção pré-selecionada, adicione o <i>selected</i> :



	<div>Cursos que já fez na Escola</div> <div>PHP ▼</div>
<code><label> </label></code>	<p>Adiciona um rótulo a vários elementos:</p> <ul style="list-style-type: none"><code><input type="checkbox"></code><code><input type="color"></code><code><input type="date"></code><code><input type="datetime-local"></code><code><input type="email"></code><code><input type="file"></code><code><input type="month"></code><code><input type="number"></code><code><input type="password"></code><code><input type="radio"></code><code><input type="range"></code><code><input type="search"></code><code><input type="tel"></code><code><input type="text"></code><code><input type="time"></code><code><input type="url"></code><code><input type="week"></code><code><meter></code><code><progress></code><code><select></code><code><textarea></code>

Exemplo de formulário

```
<!DOCTYPE html>
<html lang="pt-Br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Exemplo de formulário</title>
<body>
  <h1>Exemplo de Formulário Simples</h1>
  <form action="" method="">
    <table>
```



```
<tr>
    <th><label for="nome">Nome:</label></th>
    <td><input type="text" name="nome"
id="nome"></td>
</tr>
<tr>
    <th><label for="email">E-mail:</label></th>
    <td><input type="email" name="email"
id="email"></td>
</tr>
<tr>
    <th><label for="msg">Digite sua Mensagem:</la-
bel></th>
    <td><textarea name="msg" rows="5" cols="40">Dig-
ite sua mensagem</textarea></td>
</tr>
<tr>
    <td></td>
    <td><button>Enviar</button></td>
</tr>
</table>
</form>
</body>
</html>
```

Figura 18 – Visualização do código acima no *browser*

Exemplo de Formulário Simples

Nome:	<input type="text"/>
E-mail:	<input type="text"/>
Digite sua Mensagem:	<div><div>Digite sua mensagem</div><div></div></div>
	<input type="button" value="Enviar"/>

FINALIZANDO

Chegamos ao final desta abordagem. Agora você já é habilitado a criar páginas simples em HTML, pois vimos os elementos principais do HTML. Pode criar uma pasta e com o uso de *links* entre páginas você já pode relacionar estas páginas. Faça um exercício: crie uma página pessoal, para você apresentar profissionalmente. Com os assuntos que veremos nos próximos conteúdos, poderá fazer a formatação desta página.



REFERÊNCIAS

ALVES, W. P. **HTML e CSS**: aprenda como construir páginas web. São Paulo: Expressa, 2021.

BUSH, V. As we may think. **W3**, S.d. Disponível em: <<https://www.w3.org/History/1945/vbush/vbush.shtml>>. Acesso em: 7 dez. 2022.

CHAGAS, E. Definição da audiência: primeiro passo para ter mais resultados com seu conteúdo. **Contentools**, 2014. Disponível em: <<https://blog.contentools.com.br/marketing-de-conteudo/definicao-da-audiencia-acerte-no-primeiro-passo-e-tenha-mais-resultados-com-seu-conteudo/>>. Acesso em; 7 dez. 2022.

GALHEGO, M. **HTTP** – o mínimo que todo desenvolvedor *web* precisa saber: desenvolvimento *web*. Edição do Kindle

RODWIN, N. **Tim Berners-Lee e a World Wide Web**. Edição do Kindle.

SILVA, M. S. **Fundamentos de HTML5 e CSS3**. São Paulo: Novatec, 2021.

SUBLIME TEXT. Disponível em: <<https://www.sublimetext.com/>>. Acesso em: 7 dez. 2022.

TERUEL, E. C. **HTML5**: guia prático. 2. ed. rev. atual. e ampl. São Paulo: Érica, 2014.