

Objetivos:

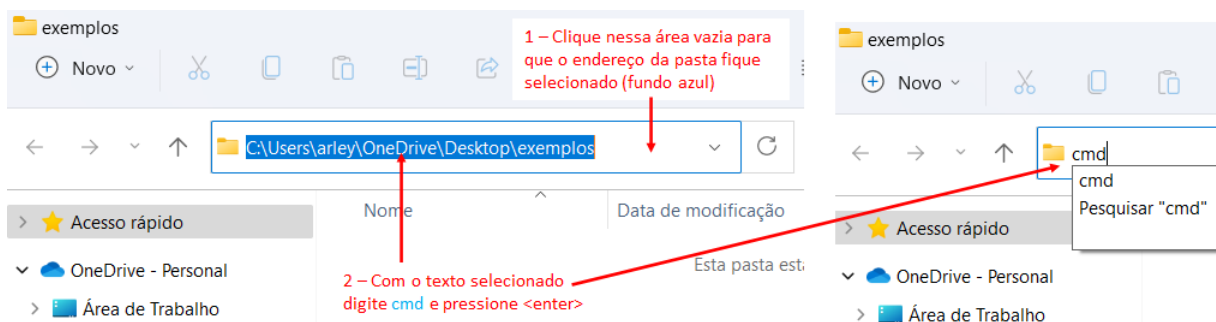
- I. Visual Studio Code;
- II. Conceitos básicos da linguagem de marcação HTML;
- III. Caracteres especiais no documento HTML;
- IV. Comentários no HTML;
- V. Principais elementos do corpo de uma página HTML.

I. Visual Studio Code

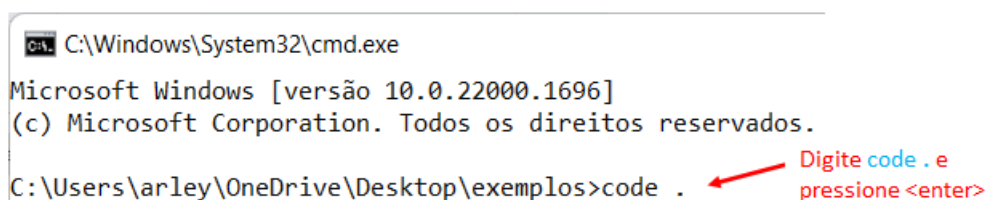
Para fazer as páginas HTML utilizaremos o VS Code. Porém, os arquivos HTML podem ser salvos em qualquer editor de texto do seu computador que aceita salvar nas extensões [html](#) ou [htm](#).

Passos para criar uma página HTML no VS Code:

- 1) Crie uma pasta de nome [exemplos](#) no desktop (área de trabalho do seu computador) ou em qualquer outro local do seu computador. Aqui foi sugerido o nome [exemplos](#), mas poderia ser qualquer nome de pasta sem espaços e caracteres especiais (acentos e cedilha);
- 2) O prompt de comando do Windows (cmd) é um ambiente que nos permite executar os programas sem o recurso do clique do ambiente janelado do Windows. A seguir estão os passos para abrir o CMD (prompt de comando do Windows) na pasta [exemplos](#):

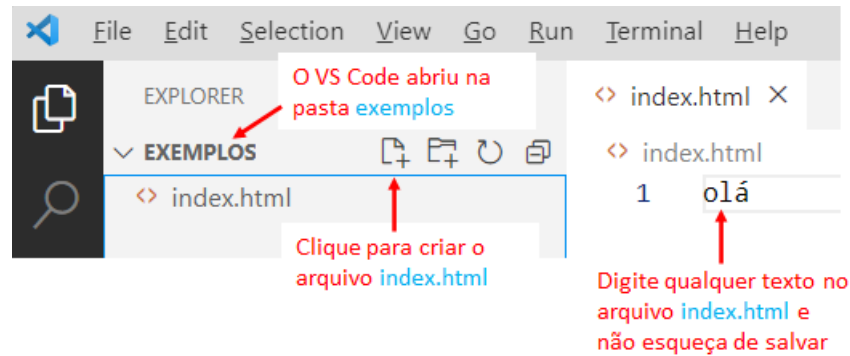


- 3) Veja que o prompt de comando abriu na pasta [exemplos](#). Digite code espaço ponto ([code .](#)) e pressione <enter> para abrir o VS Code na pasta [exemplos](#):



O VS Code será aberto na pasta [exemplos](#). Existem outras formas de abrir o VS Code numa pasta, porém esta é a sequência adotada pelos desenvolvedores da comunidade Web.

- 4) Crie o arquivo [index.html](#) na pasta [exemplos](#) e coloque um texto qualquer no arquivo. Após salvar o arquivo, abra ele no navegador.



Observação: o nome index é especial na programação Web, ao endereçar apenas a pasta o arquivo index será o arquivo escolhido para ser executado.

II. Conceitos básicos da linguagem de marcação HTML

HTML (HyperText Markup Language) **não** é uma linguagem de programação. HTML é uma linguagem de marcação utilizada para dizer ao navegador como estruturar a página web. HTML usa marcações para representar textos, imagens e outros conteúdos para exibição no navegador.

Uma marcação possui a seguinte estrutura:

```
<tagname atributoA="valorDoAtributoA" atributoB=" valorDoAtributoB">
    Conteúdo
</tagname>
```

- tagname: é o nome da marcação;
- atributo: é algo utilizado pela marcação. O valor do atributo precisa estar entre aspas simples ou duplas;
- conteúdo: pode ser um texto ou outras marcações;
- fechamento: toda marcação que possui conteúdo precisa ter a marcação de fechamento. A marcação de fechamento possui a barra de divisão /.

Algumas marcações não possuem conteúdo, desta forma, elas são formadas apenas pela marcação de abertura:

```
<tagname atributoA="valorDoAtributoA">
```

Algumas considerações:

- Uma página HTML é chamada de documento no navegador. Isso ocorre pelo fato de o navegador carregar as marcações no DOM (Document Object Model - Modelo de Objeto de Documentos). O DOM é uma API (Application Programming Interface) definida pelo W3C (World Wide Web Consortium - Consórcio da Rede Mundial) para representar e interagir com qualquer documento HTML ou XML. O W3C é uma sociedade internacional que mantém as regras e frameworks relacionadas à Web;
- O DOM é um modelo de documento carregado pelo navegador. Este documento é representado através de uma árvore de nós (marcações), onde cada um destes nós representa uma parte do documento (por exemplo, um texto, uma imagem ou comentário) (<https://developer.mozilla.org/pt-BR/docs/Glossary/DOM>);

- Uma página é formada por marcações;
- Uma marcação (tag em inglês) é muitas vezes chamada de elemento HTML;
- Um documento HTML possui a extensão .html ou .htm;

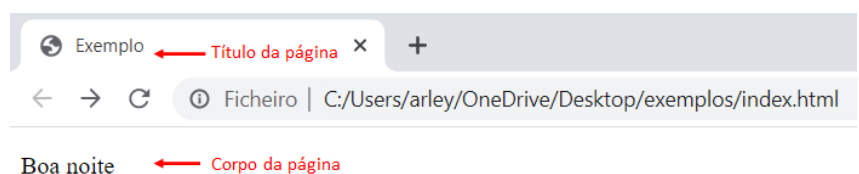
O documento HTML tem como marcação raiz o `<html>`, ou seja, **todo** o documento HTML deverá ser filho do elemento raiz. Em outras palavras, o conteúdo do documento será o conteúdo entre as marcações `<html>` `</html>`.

O código HTML a seguir possui os elementos mínimos para construirmos a estrutura de uma página.

Código do arquivo index.html

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8">
    <title>Exemplo</title>
  </head>
  <body>
    <p>Boa noite</p>
  </body>
</html>
```

Resultado no navegador



Estrutura de um documento HTML:

- `<!DOCTYPE html>` deve ser colocado no topo do documento. O navegador usa o atributo `html` do `DOCTYPE` para saber a versão que o documento HTML foi escrito. O atributo `html` em `<!DOCTYPE html>` indica que o documento está na versão HTML 5 (última versão disponível).

Observação: `DOCTYPE` não é considerado marcação.

- `<html> </html>`: as marcações `<html>` envolve o conteúdo da página inteira e é conhecido como o "elemento raiz" da página HTML. Dentro das marcações `html` devem estar apenas as marcações `head` e `body`, ou seja, as marcações `head` e `body` são consideradas filhas da marcação `html`;
 - ✓ O atributo `lang` na marcação `html` é usado para indicar o idioma do texto da página. Este recurso é interessante para os sites de buscas.
- `<head> </head>`: o `head` de um documento HTML é a parte que não é exibida no navegador quando a página é renderizada (exibida). O elemento `head` recebe as marcações utilizadas para instruir o navegador na construção da página, tais como:
 - ✓ As marcações `meta` são usadas para definir metadados da página. Os metadados são definidos pelos atributos da marcação `meta`:
 - `<meta charset="utf-8">` indica ao navegador a codificação dos caracteres do documento. Quando o navegador renderiza a página usando uma codificação diferente

da usada na criação da página, os caracteres acentuados e cedilha podem ser exibidos errados, tal como, na palavra Sebastião (SebastiÃfÃfo);

- `<meta name="viewport" content="width=device-width, initial-scale=1.0">` especifica a área visível da página na tela. Varia de acordo com o dispositivo (navegador, celular etc.).

`width=device-width` define que a página deve ocupar a largura da tela.

`initial-scale=1.0` define o zoom ao exibir a página.

- ✓ `<title>Exemplo</title>` o conteúdo da marcação será o título da página. Aparecerá na guia do navegador na qual a página está carregada e é usado para descrevê-la quando o usuário salvar nos favoritos.

- `<body> </body>`: o elemento `body` contém aquilo que será exibido na tela do navegador. O conteúdo serão outras marcações, tais como, marcações de parágrafo `<p>`, divisão `<div>`, hiperlink `<a>` etc.

III. Caracteres especiais no documento HTML

Os caracteres `<`, `>`, `"`, `'` e `&` são “caracteres especiais” no HTML, pois eles já são usados na sintaxe da linguagem de marcação do HTML. Para termos eles nas nossas páginas teremos de usar as “entidades HTML”, que são códigos especiais que representam caracteres. Cada entidade começa com o “e” comercial (&), e termina com o ponto e vírgula (;). A seguir tem-se alguns exemplos:

<code><div>2 &lt; 3</div></code>	2 < 3
<code><div>2 &le; 3</div></code>	2 ≤ 3
<code><div>3 &gt; 2</div></code>	3 > 2
<code><div>3 &ge; 2</div></code>	3 ≥ 2
<code><div>3 &ne; 2</div></code>	3 ≠ 2
<code><div>Aspas &quot;duplas&quot;</div></code>	Aspas "duplas"
<code><div>Aspas &apos;simples&apos;</div></code>	Aspas 'simples'
<code><div>e &amp;</div></code>	e &

Para outros exemplos de caracteres especiais acesse <https://www.html.am/reference/html-special-characters.cfm>.

IV. Comentários no HTML

Os marcadores `<!--` e `-->` são usados para delimitar o bloco a ser ignorado pelo navegador, isto é, ficará invisível para o usuário na tela do navegador. No exemplo a seguir as marcações `<p>Boa tarde</p>` e `<p>Boa noite</p>` não foram renderizadas na tela do navegador.

As marcações de comentários são usadas para ignorar um bloco de código ou para o programador deixar alguma descrição para outros programadores, assim como aconteceu em `<!-- exemplo de uso de comentários -->`.

Código do arquivo index.html

Resultado no navegador

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <!-- exemplo de uso de comentários -->
    <meta charset="utf-8" />
    <title>Exemplo</title>
  </head>
  <body>
    <p>Bom dia</p>
    <!--
    <p>Boa tarde</p>
    <p>Boa noite</p>
    -->
    <p>Boa semana</p>
  </body>
</html>
```

Bom dia

Boa semana

V. Principais elementos do corpo de uma página HTML

A linguagem HTML possui vários marcadores e cada um possui a sua função na composição do documento HTML. Aqui serão apresentados apenas os mais utilizados na construção das páginas. Para mais detalhes veja a lista de marcações em <https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element>.

Títulos

Os elementos HTML `<h1>` a `<h6>` representam seis níveis de título de seção. `<h1>` é o nível de seção mais alto e `<h6>` é o mais baixo.

Código do arquivo index.html

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8" />
    <title>Exemplo</title>
  </head>
  <body>
    <h1>Título H1</h1>
    <h2>Título H2</h2>
    <h3>Título H3</h3>
    <h4>Título H4</h4>
    <h5>Título H5</h5>
    <h6>Título H6</h6>
  </body>
</html>
```

Resultado no navegador

Título H1

Título H2

Título H3

Título H4

Título H5

Título H6

Parágrafo

O elemento `<p>` representa um parágrafo. Em mídias visuais, parágrafos são representados como blocos indentados de texto com a primeira letra avançada e separados por linhas em branco. Já em HTML, parágrafos são usados para agrupar conteúdos relacionados de qualquer tipo, como imagens e campos de um formulário.

Código do arquivo index.html

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8" />
    <title>Exemplo</title>
  </head>
  <body>
    <p>Domingo</p>
    <p>Segunda-feira</p>
    <p>Terça-feira</p>
  </body>
</html>
```

Resultado no navegador

Domingo
Segunda-feira
Terça-feira

Divisão

O elemento de divisão `<div>` é um container genérico para conteúdo, que de certa forma não representa nada. O `<div>` fará sentido quando aplicarmos formatações de estilos.

Código do arquivo index.html

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8" />
    <title>Exemplo</title>
  </head>
  <body>
    <div>Domingo</div>
    <div>Segunda-feira</div>
    <div>Terça-feira</div>
  </body>
</html>
```

Resultado no navegador. Observe que não existe espaço entre as linhas assim como existe quando utilizamos o elemento `<p>`.

Domingo
Segunda-feira
Terça-feira

SPAN

O elemento `` é um container genérico em linha para conteúdo fraseado, que não representa nada por natureza (assim como o `<div>`). O `` fará sentido quando aplicarmos formatações de estilos.

Código do arquivo index.html

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8" />
    <title>Exemplo</title>
  </head>
  <body>
    <span>Domingo</span>
    <span>Segunda-feira</span>
    <span>Terça-feira</span>
  </body>
</html>
```

Resultado no navegador. Observe que os três elementos ficaram na mesma linha, isso ocorre pelo fato de o `` ser inline.

Domingo Segunda-feira Terça-feira

Existem os elementos que formam blocos e os elementos inline (na linha). Os elementos de bloco ocupam toda a largura disponível e os elementos inline não alteram o fluxo do texto. `<p>` e `<div>` são elementos de bloco e `` é elemento inline.

Hyperlink

O elemento hyperlink `<a>` é o que torna a web uma rede, pois é através deles que navegamos de uma página para outra.

Um hyperlink é criado envolvendo um texto ou outros elementos HTML com a marcação `<a>`. No exemplo a seguir os textos [G1](#) e [UOL](#) foram envolvidos pela marcação `<a>`, desta forma, a navegação de uma página para outra ocorrerá ao clicar no texto [G1](#) ou [UOL](#). No outro exemplo a navegação ocorrerá ao clicar nos textos [Buscador](#) ou [Google](#), pois a marcação `<a>` envolve as marcações `<p>`.

Código do arquivo index.html

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8" />
    <title>Exemplo</title>
  </head>
  <body>
    <!-- a página do G1 será aberta na aba
         atual do navegador -->
    <a href="http://g1.globo.com">G1</a>
    <!-- o atributo target com valor _blank indica a
         página será aberta em outra aba do navegador -->
    <a href="https://www.uol.com.br" target='_blank'
        title='Clique para ir'>UOL</a>
    <!-- as marcações <p> são o conteúdo da marcação <a> -->
```

Resultado no navegador

[G1 UOL](#)

[Buscador](#)

[Google](#)

Não é um hyperlink [Página atual](#)

```
<a href='https://www.google.com/'>
  <p>Buscador</p>
  <p>Google</p>
</a>
<!-- não é hyperlink pela ausência do atributo href -->
<a>Não é um hyperlink</a>
<!-- o valor # no atributo href indica a página atual -->
<a href='#' target='_blank'>Página atual</a>
</body>
</html>
```

Observações:

- O atributo **href** cria a hiperligação entre as páginas. O **href** recebe a URL (Uniform Resource Locator - Localizador Uniforme de Recursos) de destino;
- A marcação a seguir não é um hyperlink pela falta do atributo **href**:

```
<a>Não é um hyperlink</a>
```
- O valor **#** no atributo **href** é usado para fazer um link para a própria página, isto é, quando o usuário clicar no texto **Página atual** o navegador abrirá a página atual novamente;
- O atributo **target** com valor **_blank** indica que o link será aberto em outra aba do navegador, sendo que o padrão é **_self**, que significa abrir na aba atual;
- O atributo **title** é usado para exibir um texto quando o usuário posicionar o cursor sobre o elemento.

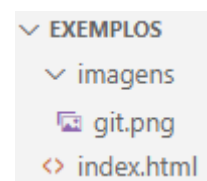
Imagem

O elemento **** é usado para incorporar uma imagem na página. No processo de carregamento da página, primeiramente o navegador carrega o documento HTML e na sequência ele faz a requisição dos recursos incorporados, assim como as imagens. No exemplo a seguir a imagem **logo.png** é um arquivo a parte, isto é, ele não fica dentro do arquivo HTML.

```
<img src='imagens/git.png' alt='GitHub' width="50" height="50" title="Mensagem">
```

O elemento **** possui apenas a marcação de abertura e requer os atributos:

- **src**: **source** - fonte - recebe a URL relativa ou absoluta para o arquivo da imagem.
 Uma URL absoluta é aquela que possui o caminho completo, desde a raiz, por exemplo, **C:\Users\Desktop\exemplos\imagens\git.png**.
 Um caminho relativo parte do local onde está a marcação até a imagem, como estamos no arquivo **index.html**, então para chegarmos no arquivo **git.png** temos de ter **imagens/git.png**.
- **alt**: texto a ser exibido se o arquivo da imagem não puder ser localizado. O atributo **alt** é usado pelos mecanismos de buscas e pelos leitores de telas para deficientes visuais;



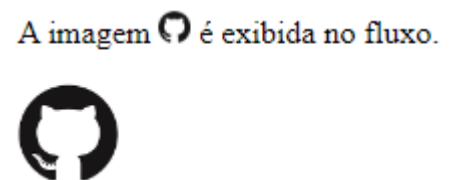
- **width** e **height**: não são obrigatórios, mas ao especificar eles, a página é renderizada já considerando a largura e altura da imagem a ser carregada, isso faz diferença em redes lentas. A unidade de medida é pixels no HTML 5, ou seja, no exemplo anterior a imagem será renderizada numa área de 50x50px;
- **title**: assim como em qualquer outro elemento HTML, o atributo title é usado para mostrar um texto quando o usuário posiciona o cursor sobre o elemento.

O elemento `` não forma bloco, o seja, ele segue o fluxo do texto. Por este motivo, no exemplo a seguir, a imagem foi inserida no texto do elemento `<p>` sem quebrar o texto. Para renderizar a imagem em um bloco (nada à esquerda e direita) precisamos colocar o elemento `` dentro de um elemento que forma bloco, assim como colocamos em `<div> </div>`.

Código do arquivo index.html

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8" />
    <title>Exemplo</title>
  </head>
  <body>
    <p>A imagem <img src='imagens/git.png' width="16"
height="16"> é exibida no fluxo.</p>
    <div>
      <img src='imagens/git.png' alt='GitHub'
        width="50" height="50" title="Mensagem">
    </div>
  </body>
</html>
```

Resultado no navegador.



Listas ordenadas e não ordenadas

Os elementos `` (ordered list) e `` (unordered list) são usados para criar listas, porém esses elementos sozinhos não são capazes de criar as listas, pois uma lista é formada por itens, que são criados pelos elementos `` (list item). A seguir temos alguns exemplos de listas.

Código do arquivo index.html

Resultado no navegador.

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8" />
    <title>Exemplo</title>
  </head>
  <body>
    <p>Lista sem ordem</p>
    <ul>
      <li>Abacaxi</li>
      <li>Limão</li>
      <li>Pêra</li>
    </ul>
    <p>Lista ordenada</p>
    <ol>
      <li>Alface</li>
      <li>Repolho</li>
      <li>Tomate</li>
    </ol>
    <p>Lista aninhada</p>
    <ol>
      <li>Grãos</li>
      <li>
        Frutas:
        <ul>
          <li>Maça</li>
          <li>Mamão</li>
        </ul>
      </li>
      <li>Verduras</li>
    </ol>
  </body>
</html>
```

Lista sem ordem

- Abacaxi
- Limão
- Pêra

Lista ordenada

1. Alface
2. Repolho
3. Tomate

Lista aninhada

1. Grãos
2. Frutas:
 - Maça
 - Mamão
3. Verduras

Atributos do elemento ``:

- **type**: recebe o tipo de numeração, que pode ser:
 - **"a"** indica letras minúsculas;
 - **"A"** indica letras maiúsculas;

<pre><ol type="a"> Alface Repolho Tomate </pre>	<pre>a. Alface b. Repolho c. Tomate</pre>
<pre><ol type="A"> Alface Repolho Tomate </pre>	<pre>A. Alface B. Repolho C. Tomate</pre>

- **"i"** indica algarismos romanos minúsculos;


```
<ol type="i">
  <li>Alface</li>
  <li>Repolho</li>
  <li>Tomate</li>
</ol>
```

 i. Alface
ii. Repolho
iii. Tomate
- **"I"** indica algarismos romanos maiúsculos;


```
<ol type="I">
  <li>Alface</li>
  <li>Repolho</li>
  <li>Tomate</li>
</ol>
```

 I. Alface
II. Repolho
III. Tomate
- **"1"** indica números. É a numeração padrão caso não seja fornecido o atributo type.


```
<ol type="1">
  <li>Alface</li>
  <li>Repolho</li>
  <li>Tomate</li>
</ol>
```

 1. Alface
2. Repolho
3. Tomate
- **start**: especifica o valor inicial para a numeração dos itens;


```
<ol type="A"
start="3">
  <li>Alface</li>
  <li>Repolho</li>
  <li>Tomate</li>
</ol>
<ol type="i"
start="5">
  <li>Alface</li>
  <li>Repolho</li>
  <li>Tomate</li>
</ol>
```

 C. Alface
D. Repolho
E. Tomate

v. Alface
vi. Repolho
vii. Tomate
- **reversed**: especifica a ordem invertida para a numeração dos itens. Como o atributo recebe o valor booleano **true** ou **false**, então não precisamos fornecer o valor do atributo quando for **true**.


```
<ol type="A" start="5" reversed>
  <li>Alface</li>
  <li>Repolho</li>
  <li>Tomate</li>
</ol>
<ol type="i" start="9" reversed="true">
  <li>Alface</li>
  <li>Repolho</li>
  <li>Tomate</li>
</ol>
```

 E. Alface
D. Repolho
C. Tomate

ix. Alface
viii. Repolho
vii. Tomate

Atributo do elemento ****:

- **type**: recebe o tipo de marcador, que pode ser:
 - **"disk"** indica o marcador no formato de disco preenchido. É o marcador padrão caso não seja fornecido o atributo type.


```
<ul type="disk">
  <li>Alface</li>
  <li>Repolho</li>
  <li>Tomate</li>
</ul>
```

 • Alface
• Repolho
• Tomate

- "circle" indica o marcador no formato de círculo;


```
<ul type="circle">
  <li>Alface</li>
  <li>Repolho</li>
  <li>Tomate</li>
</ul>
```

- Alface
 - Repolho
 - Tomate
- "square" indica o marcador no formato de quadrado.


```
<ul type="square">
  <li>Alface</li>
  <li>Repolho</li>
  <li>Tomate</li>
</ul>
```

- Alface
 - Repolho
 - Tomate

Tabela

Uma tabela é formada por linhas e colunas, que constituem as células da tabela. Desta forma, o elemento `<table>` precisa de outros elementos para criar uma tabela.

A tabela é formada por linhas `tr` (table row) e cada linha é formada por células que podem ser `th` (table header) ou `td` (table data), que definem, respectivamente, as células do título e corpo da tabela.

O elemento `th` possui uma formatação em negrito e centralizado horizontalmente. A tabela é uma estrutura rígida, onde todas as linhas precisam ter o mesmo número de colunas e todas as colunas precisam ter o mesmo número de linhas. Para resolver esse problema usamos os atributos `colspan` e `rowspan` para indicar que uma célula ocupa mais de uma coluna e linha, respectivamente. No exemplo a seguir o atributo `colspan` foi usado para fazer a célula `<td colspan="3">Total: 4</td>` ocupar o espaço horizontal de 3 células e `<td rowspan="2">Técnico</td>` foi usado para ocupar o espaço vertical de 2 células.

Os elementos `thead`, `tbody` e `tfoot` são opcionais. Eles são usados para definir blocos de linhas da tabela, essa estrutura é interessante no momento de aplicarmos formatação, pois os elementos do corpo (`tbody`) podem receber uma formatação diferente dos elementos do cabeçalho (`thead`) e rodapé (`tfoot`).

Como o elemento `table` e os seus sub elementos não possuem a propriedade de borda, então tivemos de usar um estilo CSS para adicionar essa formatação. Porém, a explicação de CSS estará em aulas futuras.

Código do arquivo index.html

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8" />
    <title>Exemplo</title>
    <style>
      table { border: 1px solid; }
      td, th { border: 1px solid; }
    </style>
  </head>
  <body>
```

Resultado no navegador.

Nome	Idade	Escolaridade
Ana	20	Médio
José	28	Superior
Maria	21	Técnico
Pedro	30	
Total: 4		

colspan="3" rowspan="2"

```
<table>
  <thead>
    <tr>
      <th>Nome</th>
      <th>Idade</th>
      <th>Escolaridade</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Ana</td>
      <td>20</td>
      <td>Médio</td>
    </tr>
    <tr>
      <td>José</td>
      <td>28</td>
      <td>Superior</td>
    </tr>
    <tr>
      <td>Maria</td>
      <td>21</td>
      <td rowspan="2">Técnico</td>
    </tr>
    <tr>
      <td>Pedro</td>
      <td>30</td>
      <!-- essa linha não pode ter a
           3a coluna por causa do rowspan da
           linha anterior -->
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td colspan="3">Total: 4</td>
      <!-- essa linha só pode ter a
           1 coluna por causa do colspan ocupar
           3 colunas -->
    </tr>
  </tfoot>
</table>
</body>
</html>
```

Exercícios

Veja o vídeo se tiver dúvidas nos exercícios: <https://youtu.be/J4-kShpZoMY>

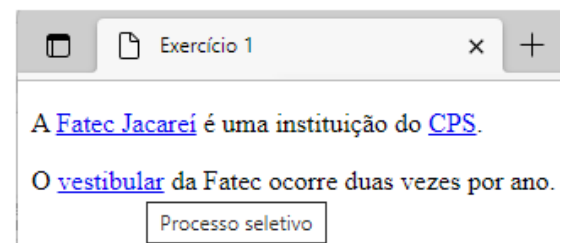
Exercício 1: Um hiperlink é uma ligação de uma página para outra. O destino do hiperlink é dado pelo valor da propriedade href do elemento <a>.

Para mais detalhes acesse <https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/a>.

Requisitos do exercício:

- Completar o código para reproduzir o exemplo. Os links estão em <http://www.fatecjacarei.com.br>, <https://www.cps.sp.gov.br> e <https://www.vestibularfatec.com.br/home>;
- Cada linha deverá estar em um parágrafo;
- Ao posicionar o cursor sobre a palavra **vestibular** deverá ser exibido o texto “Processo seletivo”;
- Ao clicar nos links **Fatec Jacareí** e **CPS** as páginas de destinos deverão ser carregadas na aba atual do navegador;
- Ao clicar no link **vestibular** a página de destino deverá ser carregada em outra aba do navegador.

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8" />
    <title>Exercício 1</title>
  </head>
  <body>
    </body>
</html>
```



Exercício 2: Completar o código HTML a seguir para colocar os logos da Fatec, CPS e brasão do Estado de SP.

Para mais detalhes acesse <https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/img>.

Requisitos do exercício:

- Cada imagem deverá estar em uma linha;
- A imagem da Fatec deverá ocupar 76x36px;
- A imagem do CPS deverá ocupar 85x56px;
- A imagem do brasão do Estado deverá ocupar 81x54px.

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8" />
    <title>Exercício 2</title>
  </head>
  <body>
  </body>
</html>
```



Exercício 3: Alterar o código HTML do Exercício 2 para incluir um hyperlink em cada imagem.

Requisitos do exercício:

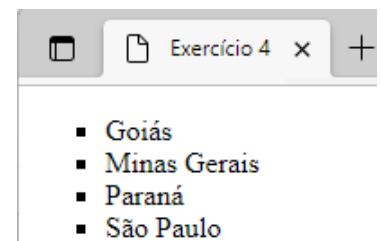
- Ao clicar na imagem da Fatec o navegador deverá carregar <http://www.fatecjacarei.com.br>;
- Ao clicar na imagem do CPS o navegador deverá carregar <https://www.cps.sp.gov.br>;
- Ao clicar na imagem do brasão o navegador deverá carregar <https://www.saopaulo.sp.gov.br>.

Exercício 4: Fazer uma página HTML com os nomes dos seguintes estados do país.

Para mais detalhes acesse <https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/ul>.

Requisito do exercício:

- Os nomes deverão estar numa lista com o marcador quadrado.

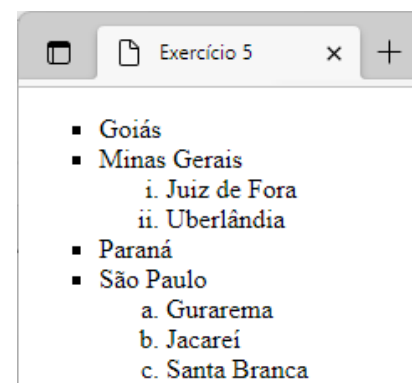


Exercício 5: Alterar o código HTML do Exercício 4 para incluir as cidades nos estados de Minas Gerais e São Paulo.

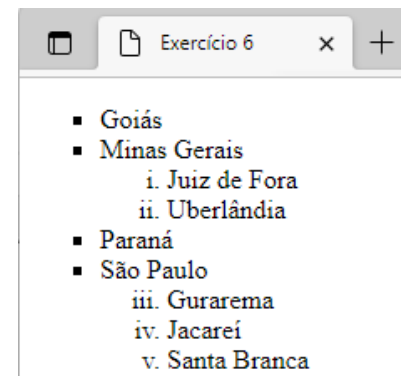
Para mais detalhes acesse <https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/ol>.

Requisitos do exercício:

- As cidades deverão ser exibidas como sub listas;
- A lista de cidades de Minas Gerais deverá ser numerada usando algarismos romanos minúsculo;
- A lista de cidades de São Paulo deverá ser numerada usando letras minúsculas do alfabeto.



Exercício 6: Alterar o código HTML do Exercício 5 para que a numeração das cidades de São Paulo continue a numeração das cidades de Minas Gerais.



Exercício 7: Criar uma tabela HTML com os nomes das disciplinas do 1º semestre de DSM.

Para mais detalhes acesse <https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/table>.

Requisitos do exercício:

- Agrupe as linhas usando os elementos `thead` e `tbody`;
- Utilize os estilos fornecidos na marcação `<style>` para criar as bordas da tabela e células.

```
<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8" />
    <title>Exercício 7</title>
    <style>
      table { border: 1px solid; }
      td, th { border: 1px solid; }
    </style>
  </head>
  <body>
  </body>
</html>
```

Período	Disciplina	Carga horária
1	Algoritmos e Lógica de Programação	80
1	Desenvolvimento Web I	80
1	Design Digital	80
1	Engenharia de Software I	80
1	Modelagem de Banco de Dados	80
1	Sistemas Operacionais e Redes de Comp.	80

Exercício 8: Alterar o código HTML do Exercício 7 para que as células da 1ª coluna sejam agrupadas.

Período	Disciplina	Carga horária
1	Algoritmos e Lógica de Programação	80
	Desenvolvimento Web I	80
	Design Digital	80
	Engenharia de Software I	80
	Modelagem de Banco de Dados	80
	Sistemas Operacionais e Redes de Comp.	80

Exercício 9: Alterar o código HTML do Exercício 8 para ter uma linha com o total de horas das disciplinas.

Requisitos:

- Coloque a linha do total em um elemento `tfoot`;
- A célula com o texto Total é do tipo `table header`;
- Agrupe a 2ª e 3ª colunas numa única célula.

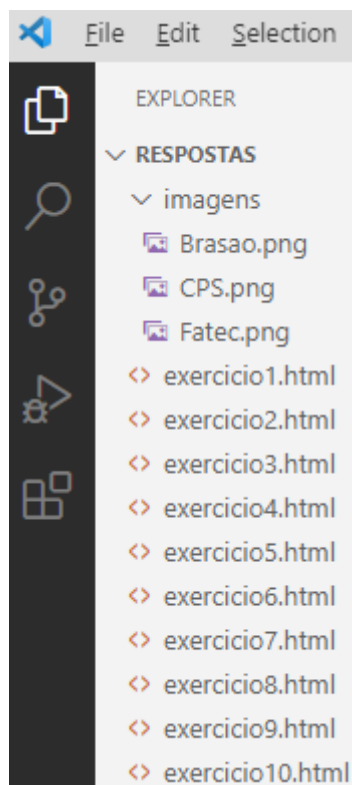
Período	Disciplina	Carga horária
1	Algoritmos e Lógica de Programação	80
	Desenvolvimento Web I	80
	Design Digital	80
	Engenharia de Software I	80
	Modelagem de Banco de Dados	80
	Sistemas Operacionais e Redes de Comp.	80
Total		480 h/a

Exercício 10: Criar uma página HTML com os links para os exercícios 1 a 9.

Requisitos do exercício:

- Crie a estrutura de pastas e arquivos sugerida a seguir. Observe que as imagens do Exercício 2 estão na pasta imagens;
- Coloque os links em uma lista ordenada por números.

Estrutura de pastas e arquivos no VS Code:



Resultado no navegador:

