

# Introdução

Para obter mais informações sobre o conteúdo, recomendo aqui as fontes de pesquisa utilizadas.

- **GitHub**

Acesse o conteúdo do repositório público, lá o Guanabara disponibiliza todo material usado em aula.



- **Módulo 1º - Curso em Vídeo**

Assista a playlist completa do curso de HTML5 e CSS3.



- **W3scools**

W3Schools é um site educacional voltado ao aprendizado de tecnologias web. Seu conteúdo inclui tutoriais e referências relacionadas a HTML, CSS e Java Script, que irão ser estudados ao longo do curso.



- **W3C**

É a organização responsável pelas normas técnicas da World Wide Web. As tags obsoletas e outras informações são encontradas no item 15.



# Ferramentas

- **VSCode**

O Visual Studio Code é um editor de código-fonte desenvolvido pela Microsoft para Windows, Linux e macOS.



- **Gimp**

GIMP (GNU Image Manipulation Program) é um programa de código aberto voltado principalmente para criação e edição de imagens, e em menor escala também para desenho vetorial.



- **HandBrake**

HandBrake é um programa multiplataforma e multitarefa em código aberto (licenciado em GPL) de conversão de arquivos de vídeo e DVD e Blu-ray para MPEG-4, disponível para Mac OS X, Linux e Windows.



# HTML BÁSICO

## MENU

Elementos HTML	4
Tag/Elemento	4
Atributos	4
Parágrafos	5
HTML Entities	6
Imagem	9
Favicon	10
Hierarquia de títulos	11
Tags obsoletas	12
Formatação de textos	13
Citações e códigos	15
Listas OL e UL	17
Links	19
Imagem dinâmica	21
Áudio	22
Vídeo	22
Estilos	23

# Elementos HTML

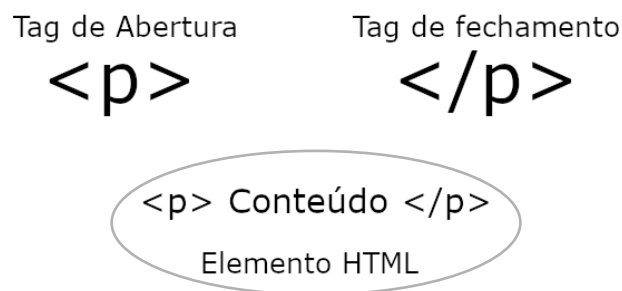
Um elemento é tudo, desde a tag de abertura até a tag de fechamento.

<p>Um parágrafo</p>		
Tag de abertura	Conteúdo do elemento	Tag de fechamento
<p>	Um parágrafo	</p>

Alguns elementos HTML não têm conteúdo (como o elemento <br>). Esses elementos são chamados de elementos vazios ou "*empty elements*" em inglês.

## Tag/Elemento

As Tags servem para marcar onde começam e terminam os Elementos. Consequentemente, os elementos englobam da tag de abertura até a tag de fechamento (veja o exemplo):



## Atributos

São sempre especificados na tag de abertura para fornecerem informações adicionais sobre os elementos, geralmente vêm em pares de nome/valor como: nome= "valor".

<a href="http://www.login.com">aqui</a>			
Tag de abertura	Nome do atributo	Nome do valor	Tag de fechamento
<a>	href	http://www.login.com	</a>

A tag âncora <a> define um hiperlink, que é usado para vincular uma página a outra. O atributo href fornece o link de destino para que a tag funcione.

# Parágrafos

Foram dados os primeiros passos! Começamos pela estrutura básica de um documento em HTML.

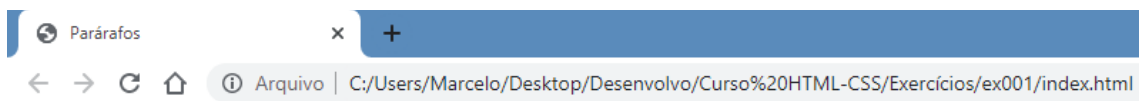


<pre>&lt;!DOCTYPE html&gt;</pre>	1 – Indica que o documento está em HTML5
<pre>&lt;html&gt;</pre>	2 e 9 – Delimita o documento HTML
<pre>&lt;head&gt;</pre>	3 e 5 – Contém as informações do documento
<pre>&lt;/head&gt;</pre>	
<pre>&lt;body&gt;</pre>	6 e 8 – Contém o conteúdo do documento
<pre>&lt;/body&gt;</pre>	
<pre>&lt;/html&gt;</pre>	

Preenchendo um pouco mais.

```
<!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>Parágrafos</title>
</head>
<body>
  <h1>Olá Mundo</h1>
  <p>Um parágrafo</p>
</body>
</html>
```

Obtivemos este resultado:



## Olá Mundo

Um parágrafo

## DICA I

Com as teclas Ctrl + Enter você pode pular linhas sem alterar o elemento.

## HTML Entities

Diferente das tags, os caracteres são exibidos em HTML no navegador. Um caractere inicia-se com e-comercial (&) e termina com ponto e vírgula (;).

Desta maneira:

```
9  <body>
10  <h1>Adicionando símbolos</h1>
11  <p>Veja alguns exemplos: &reg; &copy; &trade;</p> <br>
12  <p>Veja alguns exemplos de emoji: &#x1F440; &#x1F4A3; &#x1F4A1;</p>
13  </body>
```

Obtivemos este resultado visual:



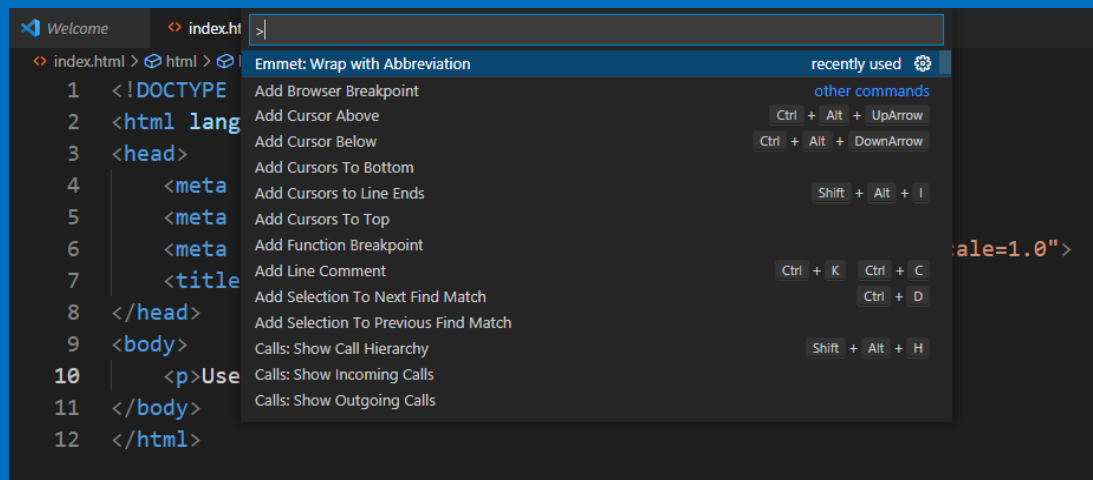
Percebeu? Geramos esse espaçamento entre os parágrafos através da tag `<br>`, que significa quebra de linha ou "*line break*" em inglês.

Veja também esses outros:

<b>Símbolo</b>	<b>Descrição</b>	<b>Entity</b>
	Non-breaking space – Espaço em branco	&nbsp;
®	Registered trademark – Marca registrada	&reg;
©	Copyright	&copy;
™	Trade mark - Marca comercial	&trade;
😄	Emoji	&#x[code];
€	Euro	&euro;
£	Libra	&pound;
¥	Yen	&yen;
¢	Centavo americano	&cent;
∅	Vazio	&empty;
Σ	Soma	&sum;
Δ	Delta	&Delta;
<	Less then – Menor que	&lt;
>	Greater then – Maior que	&gt;
←	Leftwards arrow – Seta para esquerda	&larr;
→	Rightwards arrow – Seta para direita	&rarr;
↓	Downwards arrow – Seta para baixo	&darr;
↑	Upwards arrow – Seta para cima	&uarr;
"	Quote – Citar	&quot;
'	Apostrophe – Apóstrofe	&apos;
<a href="#">Veja mais</a>		

## DICA 2

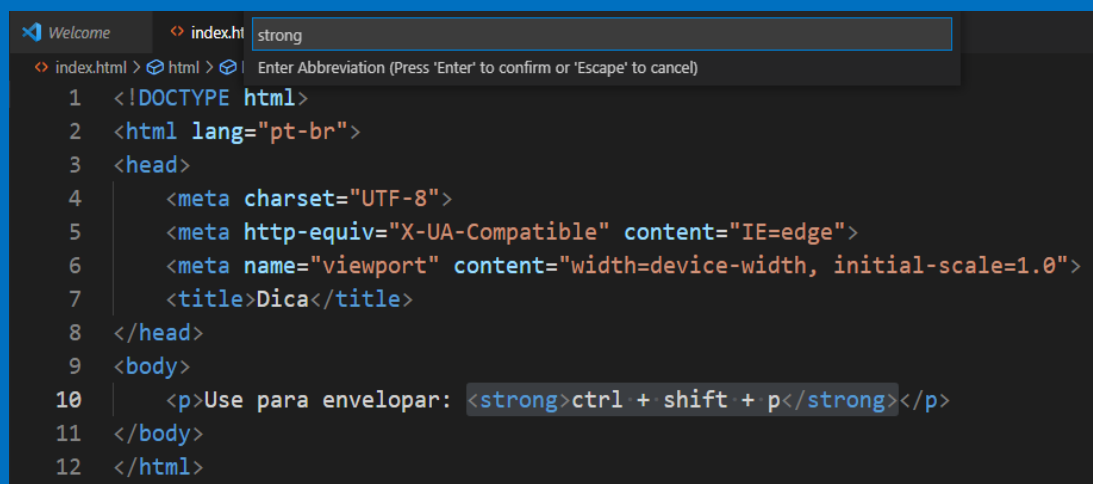
Pressione as teclas Ctrl + shift + p e selecione a opção "*Emmet: Wrap with Abbreviation*" para envelopar uma seleção – é uma maneira de economizar seu tempo ao inserir tags (veja abaixo).



The screenshot shows the Visual Studio Code interface with a file named `index.html` open. The file content is as follows:

```
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>Dica</title>
8 </head>
9 <body>
10  <p>Use para envelopar: <strong>ctrl + shift + p</strong></p>
11 </body>
12 </html>
```

The Emmet command palette is open, showing the "Emmet: Wrap with Abbreviation" command selected. The palette also lists other commands such as "Add Browser Breakpoint", "Add Cursor Above", "Add Cursor Below", "Add Cursors To Bottom", "Add Cursors To Line Ends", "Add Cursors To Top", "Add Function Breakpoint", "Add Line Comment", "Add Selection To Next Find Match", "Add Selection To Previous Find Match", "Calls: Show Call Hierarchy", "Calls: Show Incoming Calls", and "Calls: Show Outgoing Calls".



The screenshot shows the Visual Studio Code interface with the same file `index.html`. The file content is now as follows:

```
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>Dica</title>
8 </head>
9 <body>
10  <p>Use para envelopar: <strong>ctrl + shift + p</strong></p>
11 </body>
12 </html>
```

The file content is identical to the previous screenshot, but the command palette is no longer open. The text "Use para envelopar: <strong>ctrl + shift + p</strong></p>" is now wrapped in a `<strong>` tag.





### DICA 3

Com as teclas Ctrl + L você pode selecionar uma linha completa.

## Favicon

Fizemos uso de *favicon* – ícone associado com uma URL.



Passando essa ideia para o VSCode, vamos adicionar o `link:favicon`.

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   |
5   | <link:favicon
6 </head>
7 <body>
8   |
9   | <link:favicon
10 </html>
```

Dessa maneira, coloque-o acima de `<title>`.

```
3 <head>
4   | <link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
5   | <title>Teste de ícone</title>
6 </head>
```

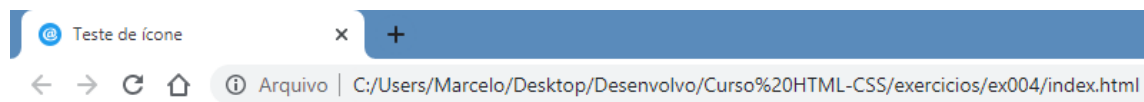
`<link`

rel      Relação de link, tipo de link ou o relacionamento entre os recursos de origem e destino

href     Referência de hipertexto – Hypertext REFerence

type     Tipo   >

Por fim, com a adição dos outros recursos, obtivemos este resultado:



**Olha o favicon aqui em cima!**

## DICA 4

Pressione Ctrl + Espaço quando for inserir o favicon. O editor irá disponibilizar todos os arquivos disponíveis da pasta do seu projeto. Isso pode evitar falhas causadas por erros de digitação.

```
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   <link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
8   <title>Dica</title>
9 </head>
```

★favicon.ico

## Hierarquia de títulos

Foi apresentado o uso de *heading*.

```
10 <body>
11   <h1>Entendendo Portugal</h1>
12   <h2>Introdução</h2>
13     <h3>Justificação da Temática</h3>
14     <p>Conteúdo</p>
15     <h3>Caráter excepcional</h3>
16     <p>Conteúdo</p>
17     <h3>Perguntas de partida</h3>
18     <p>Conteúdo</p>
19     <h3>Metodologia de Investigação</h3>
20     <h4>Etapas de Investigação</h4>
21     <p>Conteúdo</p>
22     <h4>Instrumentos de análise</h4>
23     <p>Conteúdo</p>
24 </body>
```

Assim, demarcamos o tema principal (h1) e os subtítulos subsequentes.

- <h1> Highest level, ou o mais importante heading >
- <h2> Second level heading >
- ⋮
- <h6> Sixth level heading, ou o menos importante heading >

Os títulos irão organizar o conteúdo da página, deixando-a mais atraente aos olhos do leitor.

# Tags obsoletas

Vimos um exemplo claro sobre o que não fazer com o seu site.

```
<> html4.html > ...
1  <html>
2  <head>
3  |   <title>Meu site obsoleto</title>
4  </head>
5  <body bgcolor="green">
6  |   <center>
7  |   |   <h1>Veja como não fazer um site</h1>
8  |   </center>
9  |   <marquee behavior="" direction="right">Como você pode ver o seu site vai ficar tão ruim
    |   quanto esse, usando tags obsoletas</marquee>
10 </body>
11 </html>
```

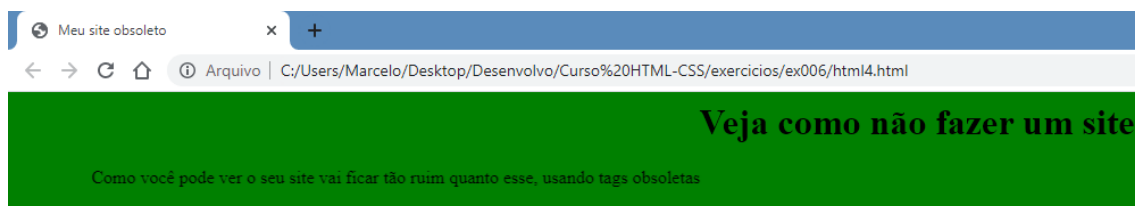
<body

    bgcolor        Cor de fundo – Backgroundcolor   >

    <center        Centro   >

    <marquee      Trecho de texto exibido horizontalmente ou vertical-  
                  mente na página do site   >

E obtivemos este resultado:



# Formatação de textos

Nesse capítulo, vimos a diferença entre as tags semânticas (de significado) e não semânticas (de forma).

Basicamente, **tags semânticas** deixam o site com suas informações bem explicadas e compreensíveis para o computador, ajudando até mesmo em sua busca no Google e facilita do o entendimento de leitores de acessibilidade.

E **tags não semânticas** não significam nada em termos de semântica, o que dificulta os mecanismos de buscas ou leitores de tela, que necessitam de ajuste na forma de apresentação de acordo com a estrutura da página de forma automatizada para que seja possível apresentar as informações aos usuários.

[Saiba mais](#)

Observe um documento com as principais formatações:

```
10 <body>
11   <h1>Principais formatações</h1> <!--Tag de significado e Tag de forma-->
12   <h2>Negrito/Destaque</h2>
13     <p>Aqui--> <b>termo em negrito</b> <!--Não Semântica--> </p>
14     <p>Aqui--> <strong>termo em destaque</strong> <!--Semântica--> </p>
15   <h2>Itálico/Ênfase</h2>
16     <p>Aqui--> <i>termo em itálico</i> <!--Não Semântica--> </p>
17     <p>Aqui--> <em>termo em ênfase</em> <!--Semântica--> </p>
18   <h2>Marcação</h2>
19     <p>Aqui--> <mark>um texto marcado</mark> <!--Semântica--> </p>
20   <h2>Texto grande e pequeno</h2>
21     <p>Aqui--> texto grande <!--Big está em desuso--> </p>
22     <p>Aqui--> <small>texto pequeno</small> </p>
23   <h2>Texto deletado, inserido e sublinhado</h2>
24     <p>Aqui--> <del>tachado</del> <!--Semântica--> </p>
25     <p>Aqui--> <ins>inserido</ins> <!--Semântica--> </p>
26     <p>Aqui--> <u>sublinhado</u> <!--Não Semântica--> </p>
27   <h2>Texto sobrescrito e subscrito <!--Semântica--> </h2>
28     <p>Aqui--> x<sup>20</sup>+3 <!--Semântica--> </p>
29     <p>Aqui--> H<sub>2</sub>O <!--Semântica--> </p>
30 </body>
```

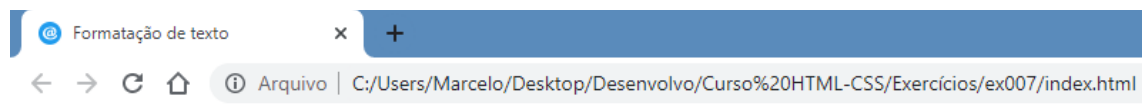
## Não semântica

- <b> bold
- <i> italic
- <u>unarticulated annotation (underline)

## Semântica

- <strong> forte
- <em> emphasis
- <small> pequeno
- <mark> marca
- <del> delete
- <ins> inserted text
- <sup> superscript
- <sub> subscript

Assim, obtivemos este resultado:



## Principais formatações

### Negrito/Destaque

Aqui--> termo em negrito

Aqui--> termo em destaque

### Itálico/Ênfase

Aqui--> termo em itálico

Aqui--> termo em ênfase

### Marcação

Aqui--> um texto marcado

### Texto grande e pequeno

Aqui--> texto grande

Aqui--> texto pequeno

### Texto deletado, inserido e sublinhado

Aqui--> tachado

Aqui--> inserido

Aqui--> sublinhado

### Texto sobrescrito e subscrito

Aqui-->  $x^{20}+3$

Aqui-->  $H_2O$



Você pode usar os comentários para dizer algo sem que apareça visualmente (como no exemplo anterior), com o `<!--comentário-->`

## Citações e códigos

Primeiramente, vimos como funciona a tag `<code>`, que é a representação de códigos para o navegador.

```
<p>document.getElementById('teste')</p>
<p><code>document.getElementById('teste')</code></p>
```

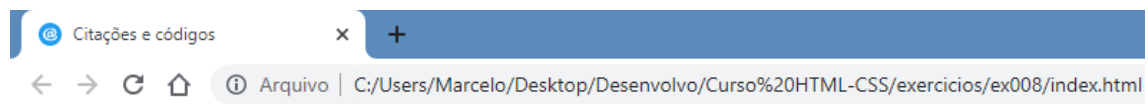
1. Um parágrafo normal na primeira linha sem formatação.
2. Na segunda linha a tag `<code>` ajusta o formato do texto.

Em seguida, implementamos a tag `<pre>` (preformatted text), juntamente com o código a seguir:

```
<pre>
  <code>
    num = int(input('Digite um npumero'))
    if num % 2 == 0:
        print(f'0 número {num} é PAR')
    else:
        print(f'0 número {num} é ÍMPAR')
    PRINT('Fim do programa')
  </code>
</pre>
```

Usamos a tag `<pre>` para preservar o texto pré-formatado, que normalmente é ignorado pelos navegadores.

Agora, observe este resultado:



## Outras formatações

### Código-fonte

```
document.getElementById('teste')
```

```
document.getElementById('teste')
```

```
num = int(input('Digite um npumero'))
if num % 2 == 0:
    print(f'0 número {num} é PAR')
else:
    print(f'0 número {num} é ÍMPAR')
PRINT('Fim do programa')
```

Em seguida, no mesmo documento trabalhamos com outros três tipos de formatação.

## Citação simples

A tag `<q>` (quotation) especifica uma citação curta.

```
<p><q>O computador é um buro muito rápido</q></p>
```

## Citação completa

A tag `<blockquote>` especifica uma seção citada de outra fonte.

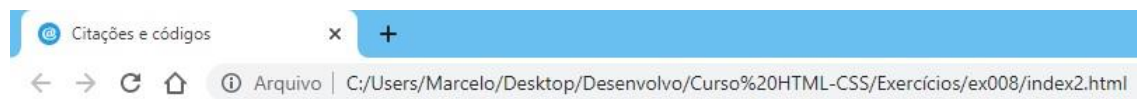
```
<p>Segundo Jeff Noble, no seu livro <em>HTML para Leigos</em></p>
<blockquote>
  A diferença entre elementos inline e um bloco de texto é importante.
</blockquote>
```

## Abreviação

A tag `<abbr>` (abbreviation) define uma abreviatura ou acrônimo.

```
Estou estudando <abbr title="HyperText Markup Language">HTML</abbr> e <abbr
title="Cascading Style Sheets">CSS</abbr>. Estou adorando!
```

Geramos este resultado:



## Outras formatações

### Citações simples

“O computador é um buro muito rápido”

### Citações completas

Segundo Jeff Noble, no seu livro *HTML para Leigos*

A diferença entre elementos inline e um bloco de texto é importante.

### Abreviações

Estou estudando HTML e CSS. Estou adorando!

HyperText Markup Language



## Listas OL e UL

Trabalhamos aqui, utilizando listas no documento HTML e vimos seus diferentes formatos. Para cada item da lista utilizamos a tag `<li>` (list item), que é colocada dentro dessas listas:

### Lista ordenada/não ordenada

Uma lista ordenada pode ser numérica ou alfabética. É representada pela tag `<ol>` (ordered lists), e uma lista não ordenada pela tag `<ul>` (unordered lists).

```
<ol>
  <li>Acordar</li>
  <li>Tomar café</li>
</ol>

<ul>
  <li>Pão</li>
  <li>Leite</li>
</ul>
```

Segundo a W3C, a tag `<li>` não necessariamente precisa de fechamento, isso é opcional.

### Lista mista

É basicamente, uma enumeração com subdivisões.

```
<ol>
  <li>Linguagens Novas</li>
  <ol type="a">
    <li>PHP</li>
    <li>Python</li>
    <li>JavaScript</li>
  </ol>
</ol>
```

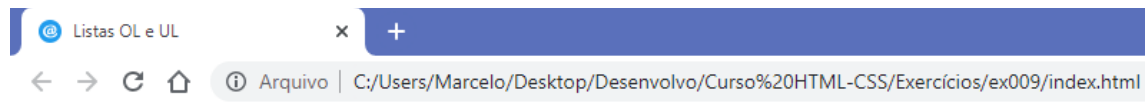
Você também pode juntar listas `<ul>` com `<ol>` e vice-versa.

### Lista de definição

É representada pela tag `<dl>` (description list). Conjuntamente, usada com as tags `<dt>` (definition term) e `<dd>` (definition description).

```
<dl>
  <dt>HTML<!--Termo--></dt>
  <dd>Linguagem de marcação para a criação do conteúdo de um site.
  <!--Descrição--></dd>
</dl>
```

Veja o resultado:



## Trabalhando com listas

### Listas Ordenadas

1. Acordar
2. Tomar café

### Listas não Ordenadas

- Pão
- Leite

### Listas mistas

1. Novas Linguagens
  - a. PHP
  - b. Python
  - c. JavaScript

### Listas de definição

HTML  
Linguagem de marcação para a criação do conteúdo de um site.

Podemos ainda usar o atributo `type` para adicionar tipos as listas.

- OL
  1. 1 – Padrão, cria listas numeradas. Ex.: 1, 2, 3,...
  2. a – Cria listas alfabéticas em minúsculas. Ex.: a, b, c, ...
  3. A – Cria listas alfabéticas em maiúsculas. Ex.: A, B, C, ...
  4. i – Cria listas com algarismos romanos em minúsculas. Ex.: i, ii, iii, ...
  5. I – Cria listas com algarismos romanos em maiúsculas. Ex.: I, II, III, ...

Para formatar o início da contagem, usa-se o atributo `start`.

Com a tag de exemplo `<ol type="a" start="4">` indicamos a contagem do quarto elemento (d) em diante.

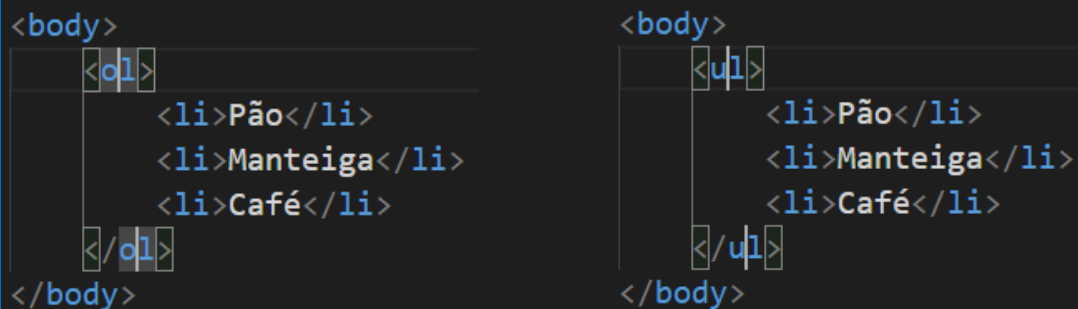
```
<ol type="a" start="4">  
  <li>PHP</li>  
  <li>Python</li>  
  <li>JavaScript</li>  
</ol>
```

- UL
  1. Disc – Padrão, cria listas de círculo preto
  2. Circle – Cria listas de circunferências
  3. Square – Cria listas de quadrado preto

## DICA 5

Você pode fazer seleções simultâneas para editar, pressionando alt.

Como neste exemplo de alterar a tag `<ol>` para `<ul>`, veja a seguir:



```
<body>
  <ol>
    <li>Pão</li>
    <li>Manteiga</li>
    <li>Café</li>
  </ol>
</body>
```

```
<body>
  <ul>
    <li>Pão</li>
    <li>Manteiga</li>
    <li>Café</li>
  </ul>
</body>
```

## Links

Como foi visto, os links são pontos de ligamentos, que ligam um documento a outro, permitindo a navegação.

Usaremos, portanto, a tag `<a>` (anchor element) para criar os hyperlinks. O principal atributo dessa tag é o `href`, que cria uma referência de hipertexto.

## Link Externo

```
<p><a href="https://gustavoguanabara.github.io">Repositório público no GitHub</a></p>
```

A ideia é redirecionar as pessoas que clicarem no link para outra aba, utilizando os seguintes requisitos:

```
<p><a href="https://gustavoguanabara.github.io" target="_blank"
rel="external">Repositório público no GitHub</a></p>
```

`<a`

`target="_blank"` Especifica outra aba para o link ser aberto

`rel="external"` Faz um link de relação com outro arquivo separado e vinculado a uma página da web em HTML >

## Link Interno

Falamos da relação de links vinculados à uma página de um mesmo site. Logo, tivemos que usar alguns conceitos anteriores para pôr em prática esse novo formato.

Criamos, portanto, dois documentos HTML: index.html e pag002.html, com os seguintes dados:

```
<> index.html
```

```
<p><a href="pag002.html" target="_self" rel="next">Segunda página</a></p>
```

```
<> pag002.html
```

```
<p><a href="index.html" target="_self" rel="prev">Volte</a></p>
```

<a

target="\_self"      Especifica uma mesma aba para o link ser aberto (padrão)

rel="next e prev"      Indica a relação de próximo e anterior, que o link vai seguir >

## Link para Download

Por fim, mas não menos importante os links para download.

```
<p><a href="Livro/Alongamento de inferiores.pdf" download="Alongamento de inferiores.pdf" type="application/pdf">Baixar o livro em PDF</a></p>
```

<a

download      Especifica que o destino será baixado quando um usuário clicar no hiperlink

type="application/pdf"      Aplicação/pdf >

# Imagem dinâmica

Nessa aula, foi mostrado a importância de imagens flexíveis na indexação dos mecanismos de busca. E surgiram duas novas tags para colocar isso em prática.

- `<picture>`  
Dá aos desenvolvedores da web mais flexibilidade na especificação de recursos de imagem
- `<source>`  
É usado para especificar vários recursos de mídia para elementos como: `<video>`, `<audio>` e `<picture>`.

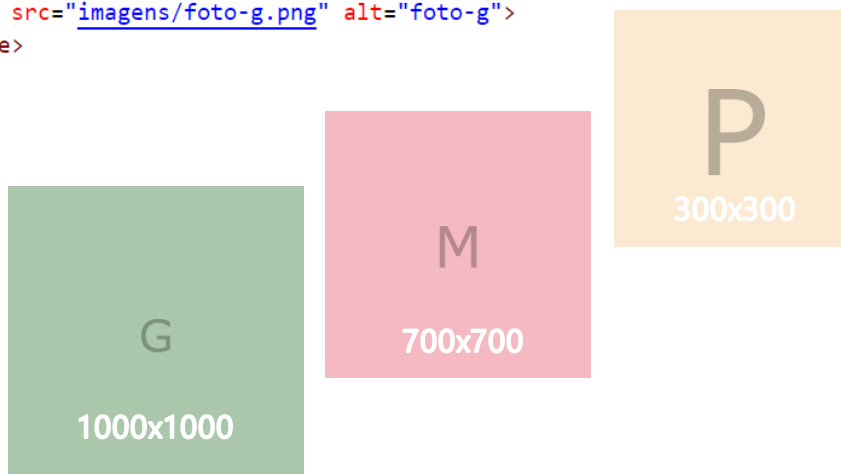
Primeiramente, geramos a base com a tag `<img>`, que vai ser o tamanho original. Em seguida, aplicando o `<source>` redefinimos as configurações para que a imagem "encurte" em telas menores.

```
<picture>
  <source media="(max-width: 1050px)" srcset="imagens/foto-m.png" type="image/png">
  
</picture>
```

<code>&lt;source</code>	
<code>media="width"</code>	Indica a largura da área de exibição do destino
<code>srcset</code>	Configura a imagem, que será carregada quando o tamanho indicado for atingido
<code>type</code>	Indica o tipo de mídia da imagem vinculada >

Para adicionar novas imagens, devemos respeitar uma hierarquia do maior para o menor, começando da base e empilhando as imagens menores subsequentes.

```
<picture>
  <source media="(max-width: 750px)" srcset="imagens/foto-p.png" type="image/png">
  <source media="(max-width: 1050px)" srcset="imagens/foto-m.png" type="image/png">
  
</picture>
```



# Áudio

Passando para áudio, podem ser adicionados de maneira simples com a tag <audio> e alguns <sources>

```
<audio preload="metadata" controls>
  <source src="midia/audio/song.mp3" type="audio/mpeg">
  <source src="midia/audio/song.ogg" type="audio/ogg">
  <source src="midia/audio/song.wav" type="audio/wav">
</audio>
```

<audio	
preload=" metadata"	Pré carrega o arquivo de mídia em apenas informações sobre o arquivo (tamanho, tempo, etc.)
controls	Apresenta o player na tela >

Adicionando outros formatos á tag <source> garante que todos poderão acessar o mesmo áudio. Caso isso falhe, criaremos uma opção de download.

# Vídeo

De maneira fácil, adicionamos vídeos, utilizando a tag <video> com alguns <source> para a mesma finalidade de antes.

```
<video width="560" poster="midia/imagens/thumb.jpg" controls>
  <source src="midia/video/meu-video.mp4" type="video/mp4">
  <source src="midia/video/meu-video.webm" type="video/webm">
  <source src="midia/video/meu-video.ogv" type="video/ogv">
</video>
```

<video	
width	Indica a largura do vídeo na tela
poster	Especifica uma imagem como capa do vídeo. Se não estiver incluso, o primeiro quadro do vídeo será usado >

# Estilos

## CSS inline style

Começamos pela forma mais simples para aplicar os estilos. Consiste em configurações pontuais, aquelas usadas em cada elemento individualmente. Porém, pouco usual em documentos longos com muitos elementos.

```
<body style="background-color: ■rgb(103, 240, 240); font-family: 'Lora';">
  <h1 style="text-align: center;">Capítulo 1</h1>
  <h2>Capítulo 1.1</h2>
  <p style="text-align: justify;">Conteúdo</p>
```

Basicamente, adiciona-se atributos e valores, separadamente, em cada elemento do corpo do documento.

## CSS internal style

Em seguida, passamos para um método mais eficiente. As configurações gerais aplicam um estilo comum a todas as tags. Esse método tende a ser mais preciso, isoladamente, em páginas separadas do conteúdo padrão.

```
<head>
  <style>
    body{
      background-color: ■rgb(103, 240, 240);
      font-family: 'Lora';
      font-size: 20px;
    }
    h1{
      text-align: center;
    }
    p{
      text-align: justify;
    }
  </style>
</head>
```

Dessa forma, é ajustado o estilo do conteúdo de acordo com as informações presentes na cabeça do documento. Porém, será mantido o estilo daquele elemento antes configurado em inline.

## CSS external style

Por fim, chegamos ao método mais usual. Isso, pois, sua finalidade é configurar, globalmente, o seu site. Constitui num arquivo separado pela CSS, interligando suas configurações a todos os documentos desejados. Dessa maneira, explica-se o motivo pela sua preferência em relação aos outros métodos.



```
# style.css > ...
1  @charset "UTF-8";
2  body{
3      background-color: rgb(103, 240, 240);
4      font-family: 'Lora';
5      font-size: 20px;
6  }
7  h1{
8      text-align: center;
9  }
10 p{
11     text-align: justify;
12 }
```

@charset "UTF-8"

Especifica a compatibilidade de codificação, com o padrão UTF-8, assim como fizemos com o arquivo HTML5. Essa linha não é gerada automaticamente, porém, é útil configurá-la de forma que evite erros ;

Primeiramente, iremos adicionar o `link:css` para solicitar o arquivo `style.css` . Após seguir o link e aceitar a mensagem de confirmação o arquivo será criado.

```
<head>
|   <link rel="stylesheet" href="style.css">
</head>
```

Siga o link (cmd + clique)

Assim, adicionando esse link a outros arquivos esses se interligarão, como na imagem.



## DICA 6

Você pode abrir todos seus arquivos com o Code. Isso ajuda, quando você está trabalhando com mais de um ao mesmo tempo.

