



HTML

Sumário

Termo de uso	3
Objetivo do documento	3
Introdução à HTML	3
Estrutura Básica	5
Editores HTML (bloco de notas, VSCode)	8
Navegador e DevTools	9
Semântica HTML	11
Títulos	16
Parágrafos	17
Texto	19
Listas	20
Iframes	21
Âncoras	22
Links	23
Imagens	24
URLs absolutas e relativas	25
Tabelas	26

Termo de uso

Todo o conteúdo deste documento é propriedade da Growdev. O mesmo pode ser utilizado livremente para estudo pessoal.

É proibida qualquer utilização desse material que não se enquadre nas condições acima sem o prévio consentimento formal, por escrito, da Growdev. O uso indevido está sujeito às medidas legais cabíveis.

Objetivo do documento

Este material tem como objetivo introduzir o estudo da linguagem HTML, servindo como apoio às aulas e atividades propostas no Programa Starter.

Introdução à HTML

HTML é uma sigla para representar o termo **Linguagem de Marcação de HiperTexto**, originado do inglês *HyperText Markup Language*. Portanto a HTML é uma linguagem para criar documentos com textos e outros elementos do meio digital, tais como imagens, sons e vídeos. Todos estes elementos serão ligados por meio de *tags* ou elementos, que vamos estudar nesta unidade/disciplina. As *tags* são recursos fundamentais da HTML. Abaixo segue exemplo de uma *tag*.

```
<img src=“foto1.jpg” >
```

A *tag* apresentada acima é a *tag img*, que indica uma abreviação de *image*, que traduzindo é imagem. Note que a *tag img* tem o atributo *src*, que indica uma abreviação de *source*, que traduzindo é origem. Os atributos são recursos presentes na maioria das *tags*.

Resumindo o conceito de *tag* com relação ao HTML:



Figura 1 Exemplo de tag HTML

tag – etiqueta, rótulo ou marcação, exemplo `` é um exemplo de tag de imagem

atributo – característica que desejamos especificar, geralmente um atributo é seguido por sinal de igualdade, exemplo `src=`

valor – geralmente fica entre aspas duplas. Em alguns casos, o valor pode ficar entre aspas simples, dependendo do padrão adotado. Vamos manter o padrão de aspas duplas.

Notas importantes.

--Perceba que a *tag* inicia com o símbolo de menor, ou seja <. Observe também um sinal de maior no final, ou seja >.

--Todas as *tags* tem uma representação de finalização opcionais. Veremos ao logo desta disciplina/unidade tais particularidades.

--Conheça mais *tags* HTML visitando este site: <https://www.w3schools.com/tags/default.asp>

O HTML foi proposto no início de 1990 e recebeu novas funcionalidades ao longo do tempo visando atender às novas exigências de comunicação e representação da informação. Atualmente o padrão HTML 5 é amplamente utilizado, no momento já temos o padrão HTML 5.3, lançado em janeiro de 2021.

Conheça mais sobre o HTML, visitando o site da Organização Mozilla, neste site você vai encontrar um manual completo e detalhado com exemplos: <https://developer.mozilla.org/pt-BR/docs/Web/HTML>

Se desejar praticar e avançar mais em seus estudos o site W3Schools oferece uma ferramenta para codificar e visualizar os resultados: <https://www.w3schools.com/html/>

O objetivo do HTML é oferecer um padrão comprehensível por nós humanos e que pode ser interpretado pelos navegadores. Um dos fatores responsáveis pelo sucesso do HTML é sua simplicidade de desenvolvimento, pois exige apenas um editor de texto e um navegador web. Além disso, ao navegar por um site qualquer na internet podemos facilmente explorar o código HTML que está sendo apresentado. Para explorar o código de uma página qualquer vamos utilizar o *DevTools*, que é apresentado aqui neste documento na seção [Navegador e DevTools](#).

O HTML é uma linguagem interpretada pelos navegadores. Mas o que exatamente significa isso?

Significa que todas as *tags* que desenvolvemos são executadas e geram algum resultado para ser exibido ao usuário. Isso significa que as *tags* não são “validadas”, nem mesmo são reportados erros caso ocorram. Este é um fator que define o HTML como linguagem de marcação e não como linguagem de programação. Uma linguagem de programação passa por um processo de validação dos comandos, onde cada um deles é executado podendo ou não gerar uma saída.

De forma resumida isso significa que quando nosso código HTML estiver incorreto, não vai apresentar erro, simplesmente não vai funcionar.

Existem diversos padrões que os programadores podem adotar, então é fundamental você conheça bastante a linguagem HTML pois vai se deparar com códigos desenvolvidos por outros programadores ao longo de sua jornada profissional e antes mesmo de executar o código é importante que você faça a validação do mesmo e entenda o que o código deve realizar.

O HTML geralmente é utilizado com outros recursos para lhe aplicar estilos, animações, transições e ventos. Para colocar todas estas funcionalidades no HTML vamos estudar também o Folhas de Estilo representado pela sigla **CSS** (do inglês **Cascading Style Sheets**) para que nosso HTML tenha estilos, animações e transições. O CSS é amplamente utilizado em sistemas de interação com o usuário, pois permite uma facilidade na geração de interfaces elegantes e modernas. Também vamos explorar o **JavaScript** que será o elo entre nossas páginas usuário e demais aplicações externas, tais como controles, eventos, bancos de dados, API e serviços.

Front-end: O termo front-end é empregado para designar o profissional que desenvolve interfaces para o usuário interagir, com controles, tais como botões, caixas de busca, formulários, telas com paginação e demais elementos que são acessados pelos usuários. Para que sejam desenvolvidas tais “telas”, vamos usar HTML, CSS e JS. Então se prepare que o HTML vai lhe preparar para se tornar um programador front-end que conhece os fundamentos desta área da programação.

Estrutura Básica

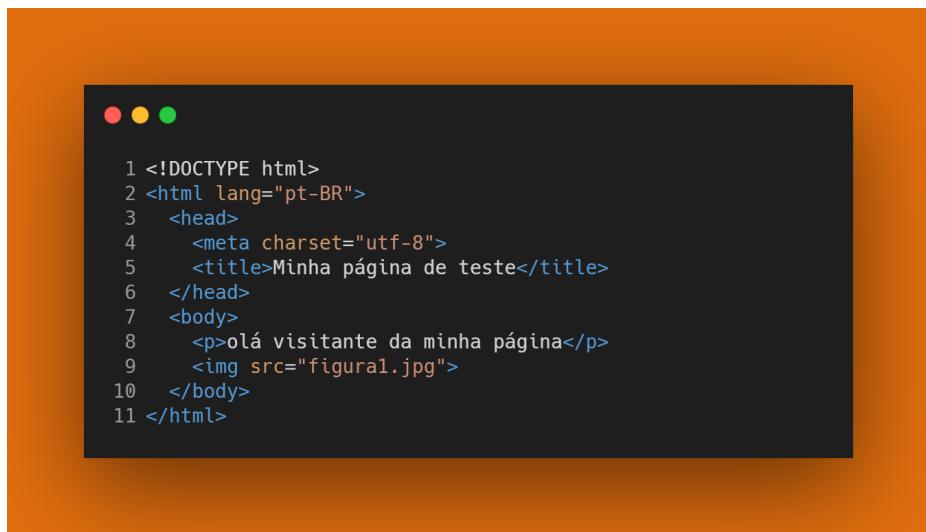
Agora que já sabemos sobre *tags*, vamos explorar a estrutura básica de um documento HTML. Um documento HTML deve ter a declaração da *tag* inicial `<!DOCTYPE html>`, embora um documento HTML possa ser interpretado sem a declaração, ela é altamente recomendada. Abaixo segue *tag* DOCTYPE:



Figura 2 Tag DOCTYPE

A *tag* DOCTYPE pode ser escrita em minúscula, ou seja, não é *case sensitive*. Algumas linguagens exigem a escrita usando letras maiúsculas em certas situações e não funcionam caso escrevemos uma letra minúscula no lugar de uma letra maiúscula, por exemplo em JavaScript uma variável definida como `pi` é diferente de uma variável definida como `PI`. Podemos afirmar que HTML não é *case sensitive*, ou seja, tanto faz se escrevermos com letra maiúscula ou minúscula, já a linguagem JavaScript é *case sensitive*. Sempre que possível vamos obedecer ao padrão e digitar corretamente as letras.

Vamos explorar a estrutura básica de uma página HTML. Veja na figura abaixo um exemplo de página HTML.



```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3   <head>
4     <meta charset="utf-8">
5     <title>Minha página de teste</title>
6   </head>
7   <body>
8     <p>Olá visitante da minha página</p>
9     
10   </body>
11 </html>
```

Figura 3 Exemplo de estrutura de página

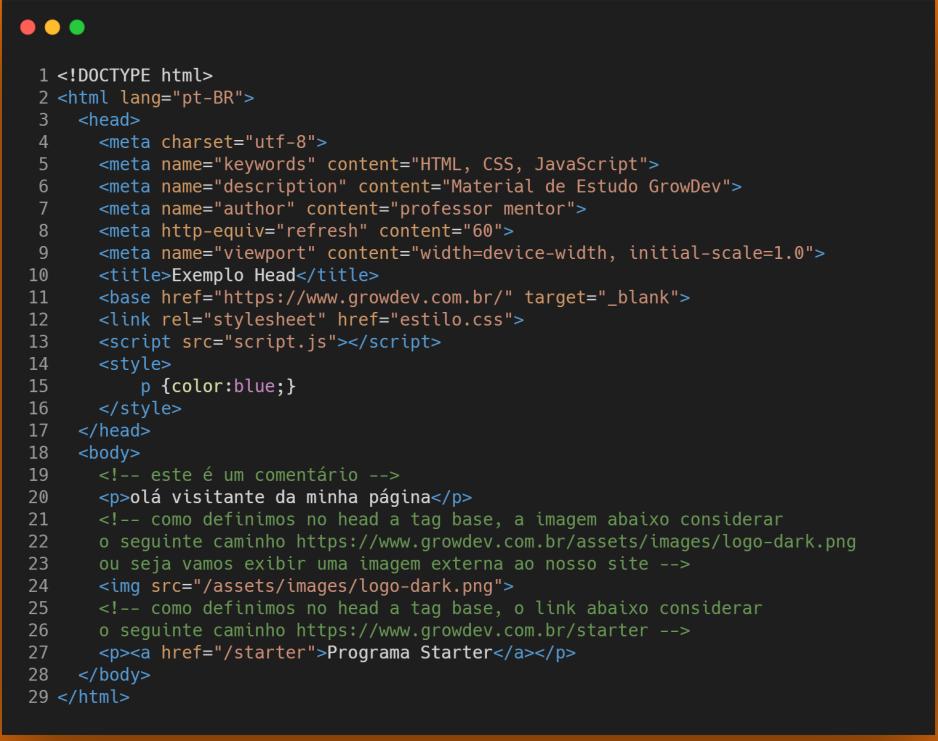
Todos os códigos apresentados neste material serão disponibilizados no material do curso. O exemplo acima pode ser obtido no arquivo **exemplo1.html**. No exemplo apresentado na figura acima temos um documento básico HTML. Note que estamos usando diversas tags, são elas: `<html>`, `<head>`, `<meta>`, `<title>`, `<body>`, `<p>` e ``. Vamos explorar cada uma delas:

A tag `<html>` é a primeira a ser aberta, antes dela sempre teremos apenas a tag `!DOCTYPE`. A tag `<html>` é fechada na linha 11, da seguinte forma `</html>`. A tag `img` não fechamos, apenas abrimos. Mais adiante veremos exemplos de tag `img` com fechamento.

A tag `<head>` é utilizada para definir configurações do cabeçalho de nossa página tais como seu conteúdo, título, palavras-chave e outras configurações que são processadas logo que a página é carregada. Na tag `<head>` temos também inclusão de arquivos CSS e JS. Explore o exemplo de página com todas as possíveis configurações do cabeçalho de nossa página, disponibilizado no arquivo **exemploHead.html** do material do curso.

Os **títulos** são representados dentro da tag `<head>` e são considerados pelos motores de busca, isso significa que se você deve sempre criar títulos relevantes e criativos para suas páginas.

Abaixo é apresentado o código do **exemploHead.html**, no qual podemos observar o uso de `<meta>` e `<title>`. A fim de preparar nossos estudos para os próximos conteúdos vamos considerar um exemplo completo com outras tags que podem aparecer dentro do `<head>`, que são: `<base>`, `<link>`, `<script>` e `<style>`.



```

1 <!DOCTYPE html>
2 <html lang="pt-BR">
3   <head>
4     <meta charset="utf-8">
5     <meta name="keywords" content="HTML, CSS, JavaScript">
6     <meta name="description" content="Material de Estudo GrowDev">
7     <meta name="author" content="professor mentor">
8     <meta http-equiv="refresh" content="60">
9     <meta name="viewport" content="width=device-width, initial-scale=1.0">
10    <title>Exemplo Head</title>
11    <base href="https://www.growdev.com.br/" target="_blank">
12    <link rel="stylesheet" href="estilo.css">
13    <script src="script.js"></script>
14    <style>
15      p {color:blue;}</style>
16    </head>
17  <body>
18    <!-- este é um comentário -->
19    <p>Olá visitante da minha página</p>
20    <!-- como definimos no head a tag base, a imagem abaixo considerar
21        o seguinte caminho https://www.growdev.com.br/assets/images/logo-dark.png
22        ou seja vamos exibir uma imagem externa ao nosso site -->
23    
24    <!-- como definimos no head a tag base, o link abaixo considerar
25        o seguinte caminho https://www.growdev.com.br/starter -->
26    <p><a href="/starter">Programa Starter</a></p>
27  </body>
28 </html>
29

```

Figura 4 ExemploHead.html

Um documento HTML deve ter apenas uma tag `<html>`. Tanto a tag `<head>` quanto a tag `<body>` são únicas por padrão, podemos mudar isso se for necessário, mas não é uma boa prática. Note que no exemplo apresentado acima temos diversas tags `<meta>`, isso significa que algumas tags podem aparecer mais de uma vez.

A próxima tag que vamos explorar é `<body>`. Nela podemos colocar tudo que desejamos exibir para nosso usuário, tais como textos, figuras e links. No exemplo anterior colocamos dentro da tag `<body>` a tag `<p>` e tag ``. Perceba que temos uma parte em verde no exemplo, todo conteúdo em verde está dentro de uma **tag de comentário, a tag `<!-- -->`**. Os comentários são úteis para colocar explicação e organizar as partes da nossa página.

Usamos também uma tag de âncora, que é a tag `Programa Starter`, dentro da tag `<p>`. Sempre que usamos uma tag `<p>` nossa página inicializa uma nova linha para colocar o conteúdo.

No exemplo temos algumas funcionalidades otimizadas, tanto a imagem quanto o link foram simplificados com o uso da tag `<base>` que vamos explorar quando desenvolvermos sistemas mais estruturados, por hora considere apenas que temos possibilidade de colocar tags internas dentro da tag `<head>` e `<body>`.

Editores HTML (bloco de notas, VSCode)

Para que possamos criar nossas páginas ou documentos HTML utilizamos um editor de texto qualquer. Embora possamos adotar até mesmo o bloco de notas, é mais aconselhável escolher um editor mais amigável que nos auxilie na produtividade, com recursos de autocompletar, tabulação, formatação em cores de acordo com as funcionalidades etc. Para ilustrar este cenário, considere a comparação abaixo:

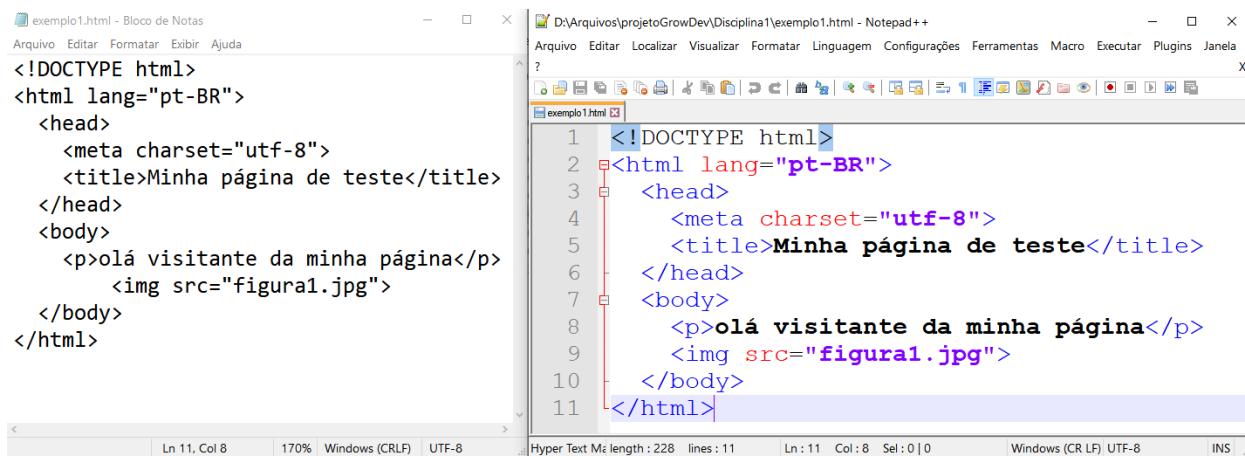


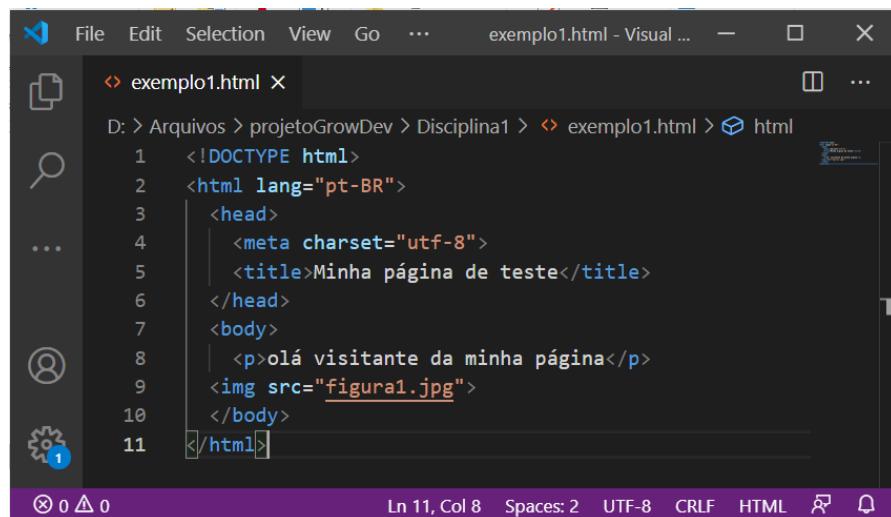
Figura 5 Comparativo Bloco de Notas versus Notepad++

No comparativo acima podemos perceber que o editor da esquerda, o Bloco de Notas do Windows 10 nos mostra apenas algumas informações na barra de status (bem abaixo) tais como linha, coluna tamanho do Zoom, formato e a codificação do arquivo, respectivamente Ln 11, Col 8, 170%, Windows (CRLF) e UTF-8. Já o editor NotePad++ a direita mostra uma barra de status bem completa, as tags em cores azuis, os atributos em vermelho, os valores dos atributos em roxo e negrito, além de mostrar o número de cada linha.

Além do Bloco de Notas e o NotePad++ existem diversos editores mais profissionais tais como Atom, Sublime Text e VSCode. No programa Starter da GrowDev vamos adotar o VSCode, mas sinta-se livre em instalar e conhecer outros editores.

O VSCode atualmente atende bem às demandas dos profissionais front-end, oferecendo suporte a autocompletar, correção de digitação, tabulação inteligente, suporte a transferência de arquivos (FTP e SFTP), área de edição compartilhada entre mais de um programador, ocultação de código, salvamento automático, ferramenta de busca e substituição entre outras funcionalidades. Além de tudo o VSCode tem as extensões que agregam opções diversas.

Confira abaixo nosso exemplo de documento HTML sendo editado no VSCode.



```

<!DOCTYPE html>
<html lang="pt-BR">
  <head>
    <meta charset="utf-8">
    <title>Minha página de teste</title>
  </head>
  <body>
    <p>Olá visitante da minha página</p>
    
  </body>
</html>
  
```

Ln 11, Col 8 Spaces: 2 UTF-8 CRLF HTML ⚙️ 🔍

Figura 6 Exemplo de documento HTML no VSCode

Confira no link a seguir uma demonstração de produtividade no VSCode com relação a auto completar o código: https://youtu.be/8jLftDn3_TM

Link para baixar o VSCode: <https://code.visualstudio.com/download>

Editores HTML e Porque utilizar editores: <https://kinsta.com/pt/blog/editores-html-gratis/>

Navegador e DevTools

Ao longo do Programa Starter da GrowDev vamos considerar o navegador Chrome. No entanto a maioria das funcionalidades e comandos apresentados funcionam nos seguintes navegadores:



Se desejar instalar outro navegador clique nos ícones acima

O Navegador Chrome oferece uma ferramenta muito útil para desenvolvedores tanto Front-end quanto Back-end, que é a “Ferramentas do programador” também conhecido como *DevTools*.

Para acessá-la, basta abrir o navegador e clicar na tecla F12. Existem mais duas formas de acessar tal ferramenta, uma delas é pelo atalho nas seguintes teclas: Ctrl+Shift+i, a outra forma é clicando nos **três pontinhos**, no final da barra de navegação bem em cima a direita, depois em **“Mais ferramentas”** e por fim em **“Ferramentas do programador”** conforme mostra a figura abaixo:

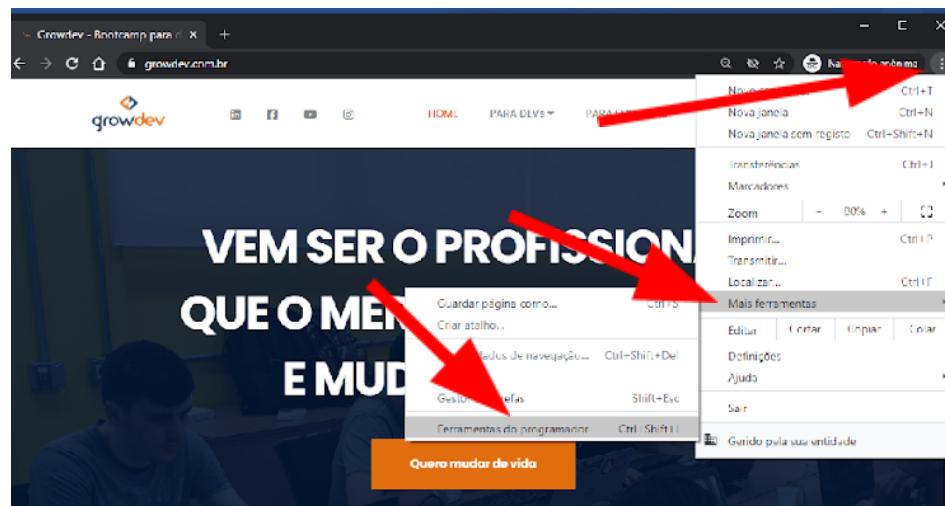


Figura 7 Abrir DevTools do Chrome

O *DevTools* permite visualizar o código fonte e muitas outras coisas. Veja abaixo a janela que é exibida ao abrir o *DevTools*.

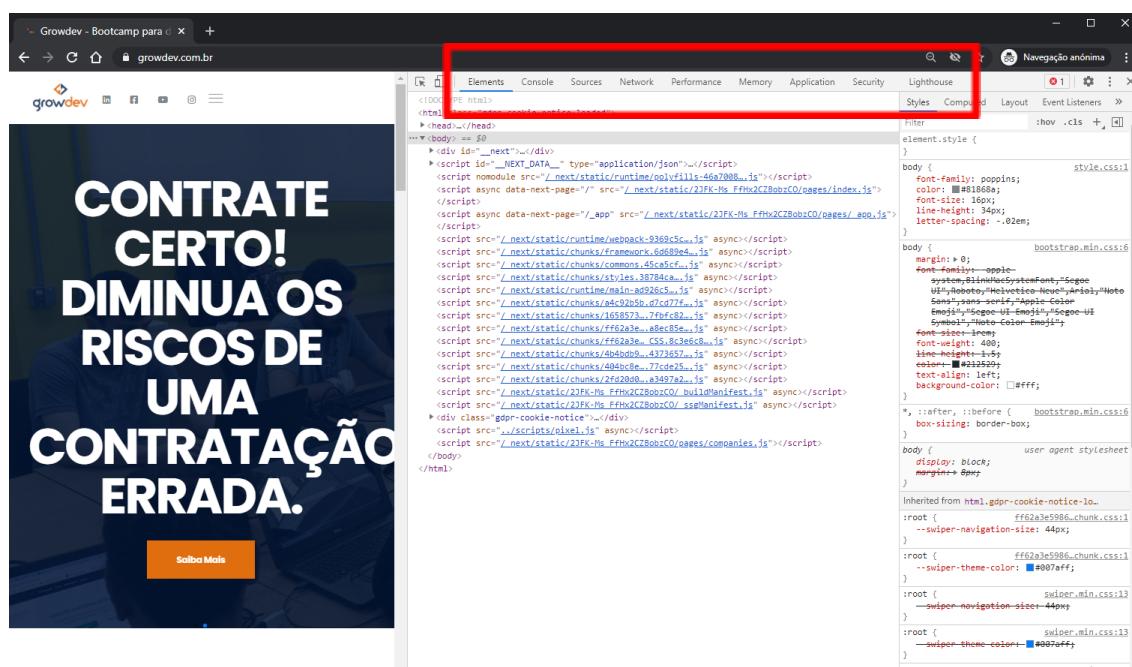


Figura 8 Janela do DevTools

Na janela que é exibida junto ao navegador temos uma sequência de abas, destacada na Figura acima pelo quadro vermelho. A primeira delas é a aba *Elements*, a qual vamos explorar bastante neste programa. O código fonte dos documentos HTML pode ser visualizado acessando a aba *Elements*.

O código fonte também pode ser visualizado sem acessar o *DevTools*. Para isso basta clicar com o botão direito do mouse em qualquer ponto da página e clicar na opção “Ver fonte da

página” ou pelas teclas de atalho Ctrl+U. Note que vai abrir uma nova aba no navegador com o código fonte do documento HTML.

Semântica HTML

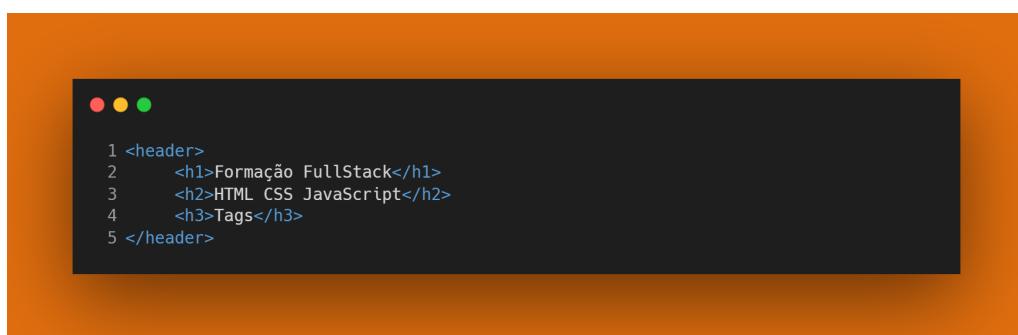
As *tags* HTML devem ser utilizadas de forma que seja compreensível para nós seres humanos e por computadores. Dar semântica a um documento significa simplificar sua codificação e fazer com o documento seja compreensível sem muito esforço. As páginas HTML são interpretadas por outras aplicações, pois uma vez estando “hospedadas” e acessíveis, qualquer aplicação pode abrir nossa página e obter as informações que precisa. Para organizar as informações de uma página existe um grupo de *elementos*, listados abaixo:

```
<header>
<section>
<article>
<nav>
<aside>
<main>
<figure>
<footer>
<a>
<em>
<strong>
<cite>
<q>
<time>
```

Vamos explorar cada um destes elementos semânticos e no final vamos olhar os exemplos todos juntos em um mesmo arquivo, o **exemploSemanticoCompleto.html**.

O elemento `<header>` é utilizado para representar o cabeçalho de um documento HTML. Dentro desta tag é possível inserir outros elementos, tais como `<p>`, ``, `<a href>`, `<h1>` etc.

Veja um exemplo do elemento `<header>`

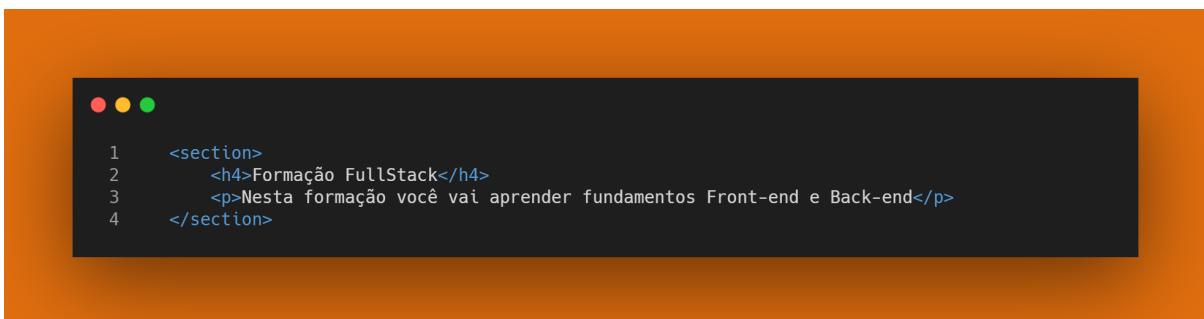


```
● ● ●
1 <header>
2   <h1>Formação FullStack</h1>
3   <h2>HTML CSS JavaScript</h2>
4   <h3>Tags</h3>
5 </header>
```

Figura 9 Elemento Header

No exemplo criamos 3 títulos, com a tag `<h1>`, `<h2>` e `<h3>` cada um deles com um tamanho. Podemos usar de 1 ao 6 para os tamanhos dos títulos.

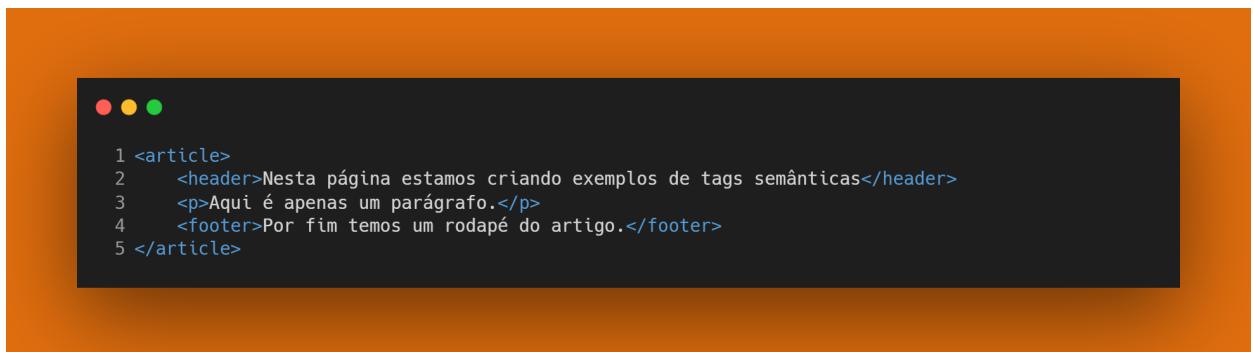
O elemento `<section>` é utilizado em uma seção depois do elemento `<header>`. Geralmente contém um título com um tamanho menor do que os títulos do `<header>`. Confira um exemplo do elemento `<section>` abaixo:



```
● ● ●
1 <section>
2   <h4>Formação FullStack</h4>
3   <p>Nesta formação você vai aprender fundamentos Front-end e Back-end</p>
4 </section>
```

Figura 10 Elemento Section

O elemento `<article>` é semelhante ao `<section>` e é utilizado quando o conteúdo é independente e não tem relação com os demais elementos da página, por exemplo, um artigo de outra página. Geralmente dentro de `<article>` temos um título. Dentro de `<article>` podemos colocar o elemento `<header>`, `<section>` e outros. Dentro do elemento `<section>` também podemos ter o elemento `<article>`. Confira um exemplo do elemento `<article>` abaixo:



```
● ● ●
1 <article>
2   <header>Nesta página estamos criando exemplos de tags semânticas</header>
3   <p>Aqui é apenas um parágrafo.</p>
4   <footer>Por fim temos um rodapé do artigo.</footer>
5 </article>
```

Figura 11 Elemento Article

Note que o elemento `<article>` tem elementos `<header>`, `<p>` e `<footer>`. O `<footer>` indica um fechamento ao qual é inserido.

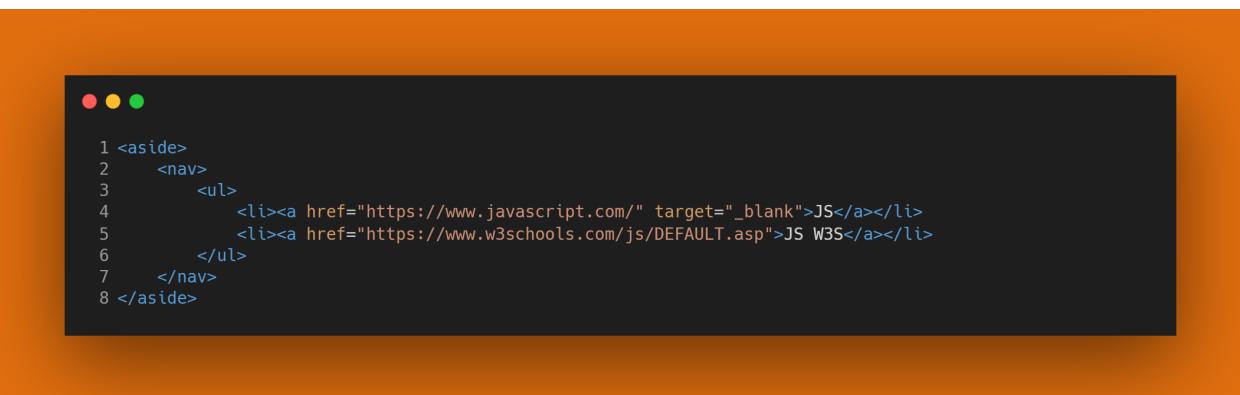
O elemento `<nav>` é utilizado para criar um grupo de links para navegação. Dentro do elemento `<nav>` geralmente temos elementos ``, `` e ``. O elemento `` é utilizado para iniciar uma lista de links, já o elemento `` é empregado para definir cada item da lista. Confira um exemplo do elemento `<nav>` abaixo:



```
1 <nav>
2   <ul>
3     <li><a href="https://www.google.com/chrome/">Chrome</a></li>
4     <li><a href="https://www.microsoft.com/pt-br/download/internet-explorer.aspx">Internet Explorer</a></li>
5     <li><a href="https://www.mozilla.org/pt-BR/firefox/">FireFox</a></li>
6     <li><a href="https://support.apple.com/downloads/safari">Safari</a></li>
7     <li><a href="https://www.opera.com/pt-br/download">Opera</a></li>
8   </ul>
9 </nav>
```

Figura 12 Elemento Nav

O elemento `<aside>` é utilizado para criar um conteúdo adicional. Por exemplo, ao falar de JavaScript podemos indicar links adicionais para que o visitante da página encontre mais fontes de informação. Confira um exemplo do elemento `<aside>` abaixo:

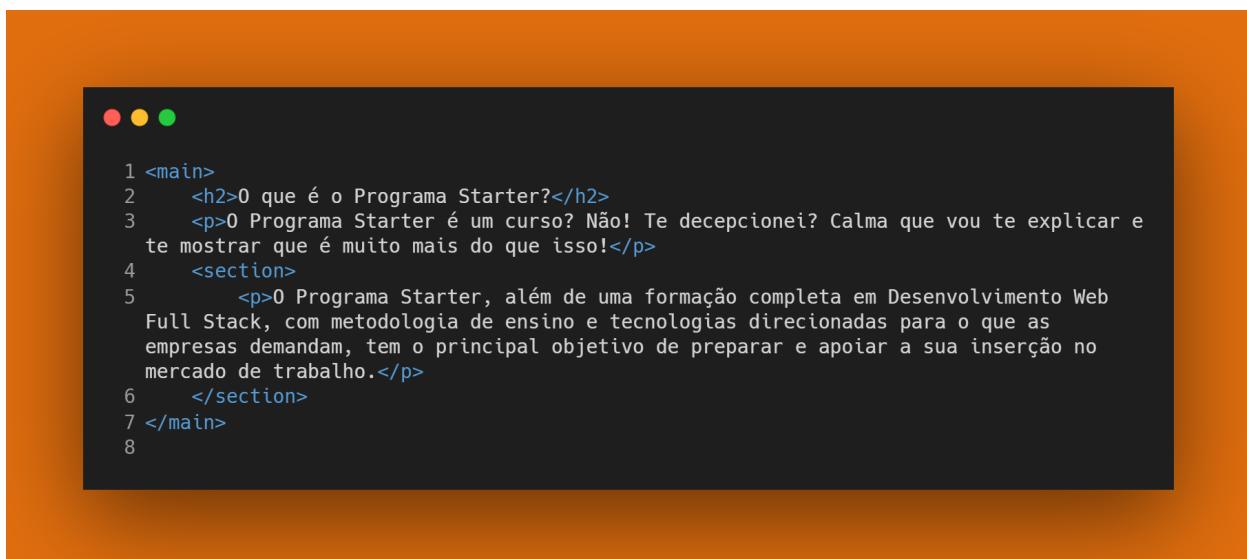


```
1 <aside>
2   <nav>
3     <ul>
4       <li><a href="https://www.javascript.com/" target="_blank">JS</a></li>
5       <li><a href="https://www.w3schools.com/js/DEFAULT.asp">JS W3S</a></li>
6     </ul>
7   </nav>
8 </aside>
```

Figura 13 Elemento Aside

Note no exemplo que o primeiro link definimos o seu atributo `target` com o valor “`_blank`”. Isso faz com que seja aberta uma nova aba no navegador, isso é relevante quando desejamos deixar a nossa página aberta em uma aba e abrir o link em uma aba separada.

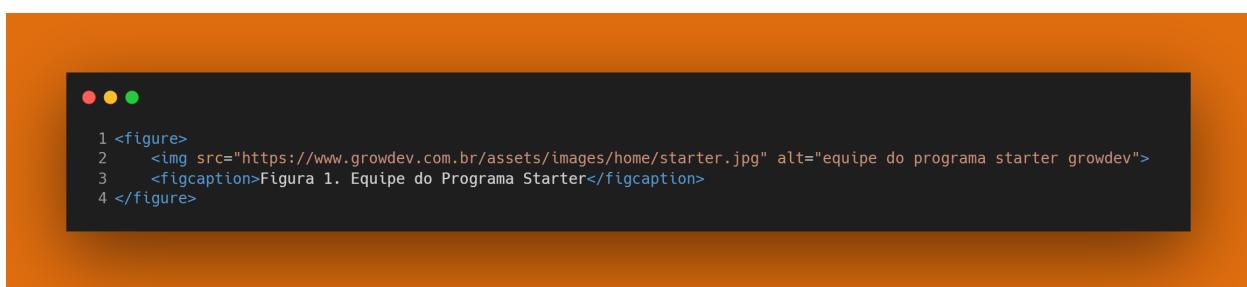
O elemento `<main>` é utilizado para definir qual o principal conteúdo de uma página. Uma página deve apresentar apenas um elemento `<main>`. Confira um exemplo do elemento `<main>` abaixo:



```
1 <main>
2   <h2>O que é o Programa Starter?</h2>
3   <p>O Programa Starter é um curso? Não! Te decepcionei? Calma que vou te explicar e
te mostrar que é muito mais do que isso!</p>
4   <section>
5     <p>O Programa Starter, além de uma formação completa em Desenvolvimento Web
Full Stack, com metodologia de ensino e tecnologias direcionadas para o que as
empresas demandam, tem o principal objetivo de preparar e apoiar a sua inserção no
mercado de trabalho.</p>
6   </section>
7 </main>
8
```

Figura 14 Elemento Main

O elemento `<figure>` é utilizado para aprimorar a interpretação de figuras. Sempre termos uma tag `` dentro do elemento `<figure>`. Geralmente temos também a tag `<figcaption>` dentro de `<figure>`. Confira um exemplo do elemento `<figure>` abaixo:



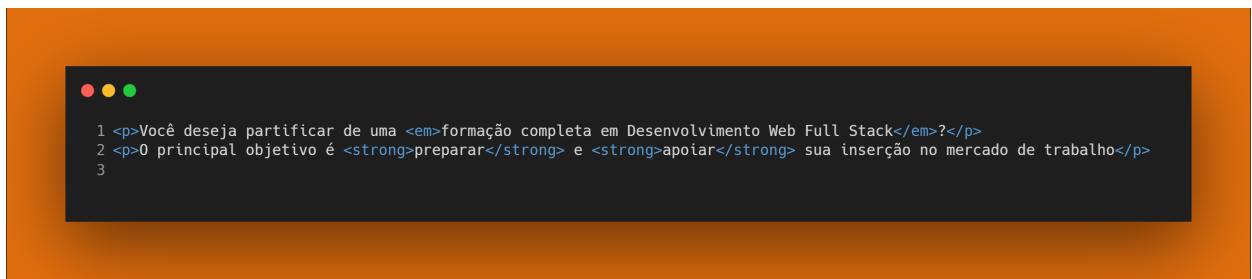
```
1 <figure>
2   
3   <figcaption>Figura 1. Equipe do Programa Starter</figcaption>
4 </figure>
```

Figura 15 Elemento Figure

Note no exemplo que definimos o atributo `alt` com o valor “equipe do programa starter growdev”. Isso faz com que nossa imagem possa ser interpretada por motores de busca e em caso de falha ou lentidão na sua transferência. Ser interpretada por motores de busca significa que você deseja que sua página seja famosa e que outros usuários encontrem sua página, você deve criar tag `alt` de acordo com seu conteúdo.

O atributo `alt` também pode ser aplicado ao elemento de ancora ``. O texto da tag `<figcaption>` é apresentado logo abaixo da imagem.

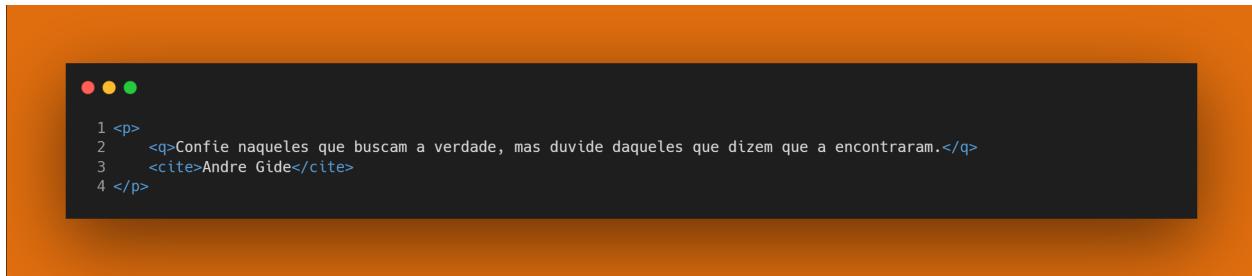
O elemento `` é utilizado para destacar a uma palavra ou expressão de nossa página, sendo que tal palavra será apresentada em itálico. Os motores de busca interpretam a palavra em destaque como mais relevante. Já o elemento `` utilizamos para dar destaque textual e o texto fica em negrito. Confira um exemplo do elemento `` e `` abaixo:



```
● ● ●
1 <p>Você deseja particiar de uma <em>formação completa em Desenvolvimento Web Full Stack</em>?</p>
2 <p>O principal objetivo é <strong>preparar</strong> e <strong>apoiar</strong> sua inserção no mercado de trabalho</p>
3
```

Figura 16 Elemento E Em e Strong

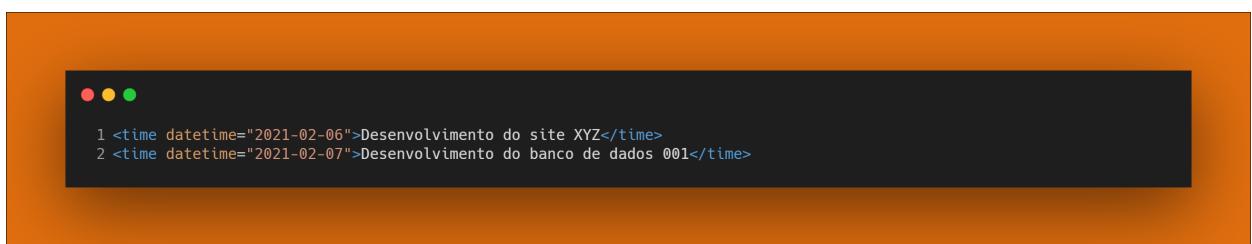
O elemento `<cite>` é utilizado para citar um trecho que foi criado por outra pessoa ou retirado de outro site, livro, revista, jornal etc. Geralmente o elemento `<cite>` é precedido do texto original dentro do elemento `<q>`. Confira um exemplo do elemento `<cite>` e `<q>` abaixo:



```
● ● ●
1 <p>
2   <q>Confie naqueles que buscam a verdade, mas duvide daqueles que dizem que a encontraram.</q>
3   <cite>Andre Gide</cite>
4 </p>
```

Figura 17 Elemento Q e Cite

O elemento `<time>` é muito utilizado pelos motores de busca e serve para representar datas. O A elemento `<time>` aceita o atributo `datetime`. Confira um exemplo do elemento `<time>` abaixo:



```
● ● ●
1 <time datetime="2021-02-06">Desenvolvimento do site XYZ</time>
2 <time datetime="2021-02-07">Desenvolvimento do banco de dados 001</time>
```

Figura 18 Elemento Time

Explore o exemplo de página com tags semânticas, disponibilizado no arquivo **exemploSemanticoCompleto.html** do material do curso.

Um exemplo prático com *tags* semânticas é apresentado na figura abaixo:



```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3   <head>
4     <meta charset="utf-8">
5     <title>Página Exemplo Semântico</title>
6   </head>
7   <body>
8     <header>
9       <h1>Programa Starter</h1>
10    </header>
11    <main>
12      <section>
13        <h2>HTML</h2>
14        <article>
15          <h3>Desenvolvimento de páginas simples</h3>
16          <p>Vamos explorar as tags e seus exemplos.</p>
17        </article>
18      </section>
19      <section>
20        <h2>CSS</h2>
21        <article>
22          <h3>Criação de estilos</h3>
23          <p>Personalização de cores tamanhos e demais elementos visuais.</p>
24        </article>
25        <article>
26          <h3>Arquivos css</h3>
27          <p>Separação de estilos em arquivos específicos.</p>
28        </article>
29      </section>
30      <section>
31        <h2>JavaScript</h2>
32        <article>
33          <h3>Lógica de programação</h3>
34          <p>Vamos explorar conceitos básicos de lógica de programação.</p>
35        </article>
36        <article>
37          <h3>Eventos e Funções</h3>
38          <p>Vamos explorar eventos e funções.</p>
39        </article>
40      </section>
41    </main>
42    <footer>
43      <p>GrowDev - Programa Starter.</p>
44    </footer>
45  </body>
46 </html>
```

Figura 19 Exemplo Semântico

O exemplo apresentado acima com tags semânticas foi disponibilizado no arquivo **exemploSemantico.html** do material do curso.

Explore mais sobre semântica HTML na prática visitando o link abaixo:

<https://medium.com/collabcode/meu-html-%C3%A9-sem%C3%A2ntico-e-o-seu-4e97c81c0c49>

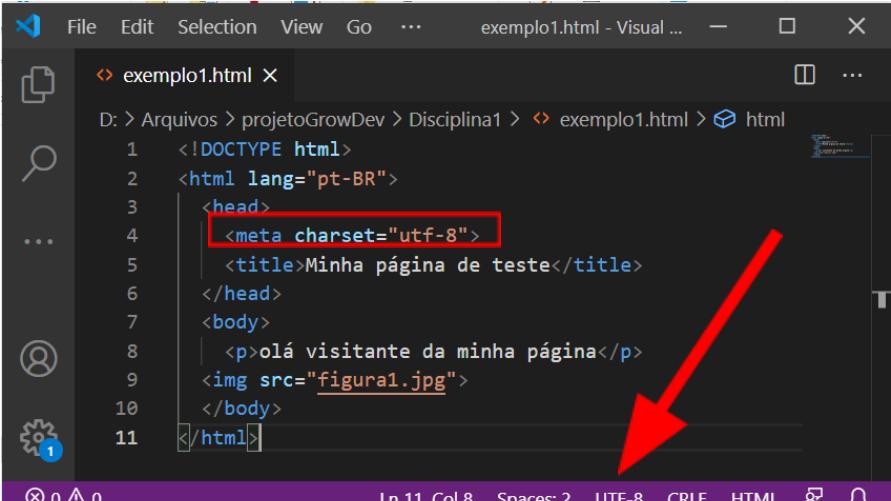
Para explorar mais exemplos dos elementos semânticos, visite o link abaixo:

<https://www.devmedia.com.br/html-semanntico-conheca-os-elementos-semannticos-da-html5/38065#tag-nav>

Títulos

A tag `<title>` deve estar presente na tag `<head>` para que nosso site exiba um título na aba do navegador e para que ele possa ser encontrado nas buscas que são realizadas pelos usuários da internet.

Vamos analisar novamente a figura com Exemplo de documento HTML no VSCode. Agora observe o Encoding que o editor está utilizando: UTF-8. Isso significa que devemos definir na tag `<meta>` o atributo `charset= "utf-8"`.



```
File Edit Selection View Go ... exemplo1.html - Visual ... D: > Arquivos > projetoGrowDev > Disciplina1 > exemplo1.html > html
1  <!DOCTYPE html>
2  <html lang="pt-BR">
3  |   <head>
4  |   |   <meta charset="utf-8"> [red box]
5  |   |   <title>Minha página de teste</title>
6  |   |>
7  |   </head>
8  |   <body>
9  |   |   <p>Olá visitante da minha página</p>
10 |   |   
11 |   </body>
|> </html>
```

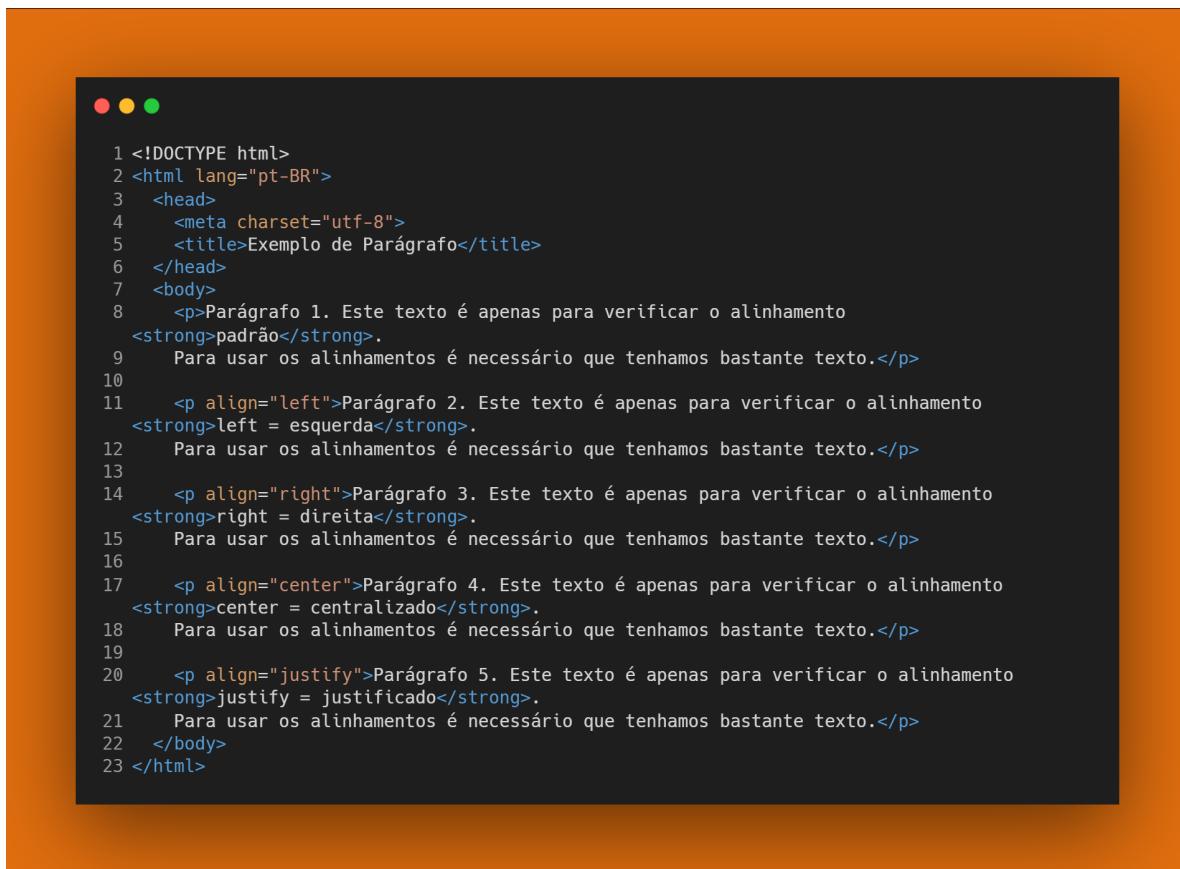
Ln 11, Col 8 Spaces: 2 UTF-8 CRLF HTML ⚙️ ⚙️

Figura 20 Exemplo de Documento HTML no VSCode - Observando o Encoding

Desta forma cuidado importe a ser considerado é que caso o título tenha acentos devemos informar a codificação de nosso documento de acordo com nosso editor.

Parágrafos

Ao usar a tag `<p>` a linha é quebrada, ou seja, o conteúdo que colocarmos entre `<p>` e `</p>` vai ficar em uma linha exclusiva. A tag `<p>` pode receber diversos atributos bem como adotar um estilo pré-definido. Outras formas de quebrar linha é utilizando a tag `
` e `<hr>`. Confira alguns exemplos da tag `<p>` abaixo:



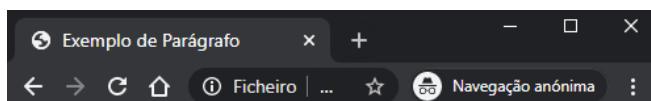
```

1 <!DOCTYPE html>
2 <html lang="pt-BR">
3   <head>
4     <meta charset="utf-8">
5     <title>Exemplo de Parágrafo</title>
6   </head>
7   <body>
8     <p>Parágrafo 1. Este texto é apenas para verificar o alinhamento
9       <strong>padrão</strong>.
10      Para usar os alinhamentos é necessário que tenhamos bastante texto.</p>
11
12     <p align="left">Parágrafo 2. Este texto é apenas para verificar o alinhamento
13       <strong>left = esquerda</strong>.
14      Para usar os alinhamentos é necessário que tenhamos bastante texto.</p>
15
16     <p align="right">Parágrafo 3. Este texto é apenas para verificar o alinhamento
17       <strong>right = direita</strong>.
18      Para usar os alinhamentos é necessário que tenhamos bastante texto.</p>
19
20     <p align="center">Parágrafo 4. Este texto é apenas para verificar o alinhamento
21       <strong>center = centralizado</strong>.
22      Para usar os alinhamentos é necessário que tenhamos bastante texto.</p>
23   </body>
24 </html>

```

Figura 21 Exemplo da Tag de Parágrafo <p>

O exemplo apresentado acima com tags de parágrafo foi disponibilizado no arquivo **exemploParagrafo.html** do material do curso. O resultado pode ser observado na figura abaixo, onde temos o parágrafo 1 com alinhamento padrão, parágrafo 2 com alinhamento na esquerda, parágrafo 3 com alinhamento na direita, parágrafo 4 com alinhamento centralizado e parágrafo 5 com justificado:



Parágrafo 1. Este texto é apenas para verificar o alinhamento **padrão**. Para usar os alinhamentos é necessário que tenhamos bastante texto.

Parágrafo 2. Este texto é apenas para verificar o alinhamento **left = esquerda**. Para usar os alinhamentos é necessário que tenhamos bastante texto.

Parágrafo 3. Este texto é apenas para verificar o alinhamento **right = direita**. Para usar os alinhamentos é necessário que tenhamos bastante texto.

Parágrafo 4. Este texto é apenas para verificar o alinhamento **center = centralizado**. Para usar os alinhamentos é necessário que tenhamos bastante texto.

Parágrafo 5. Este texto é apenas para verificar o alinhamento **justify = justificado**. Para usar os alinhamentos é necessário que tenhamos bastante texto.

Figura 22 Resultado dos alinhamentos

Texto

Para trabalhar com texto, além da tag `<p></p>` podemos utilizar as seguintes tag:

`` - É semelhante a tag `<p>`, mas geralmente é utilizada para pequenas informações, tais como legenda de outro elemento.

`<pre></pre>` - É utilizada para representar texto pré-formatado. Muito utilizada com CSS e para inserir códigos.

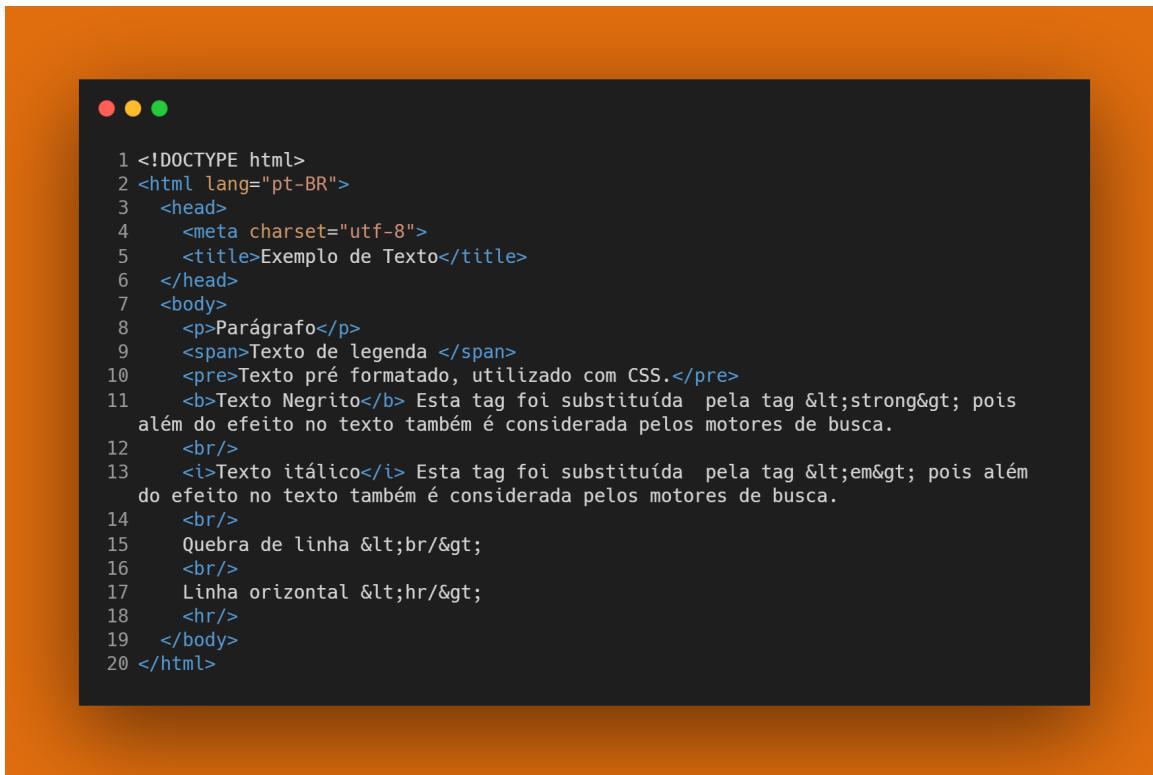
`` - Define o conteúdo em negrito. Esta tag foi substituída pela tag `` pois além do efeito no texto também é considerada pelos motores de busca.

`<i></i>` - Define o conteúdo em itálico; Esta tag foi substituída pela tag `` pois além do efeito no texto também é considerada pelos motores de busca.

`
` - Essa tag não exige fechamento, ela executa a função de quebra de linha.

`<hr/>` – Essa tag não exige fechamento, ela forma uma linha horizontal.

Confira alguns exemplos de texto no exemplo abaixo:



```
● ● ●
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3   <head>
4     <meta charset="utf-8">
5     <title>Exemplo de Texto</title>
6   </head>
7   <body>
8     <p>Parágrafo</p>
9     <span>Texto de legenda </span>
10    <pre>Texto pré formatado, utilizado com CSS.</pre>
11    <b>Texto Negrito</b> Esta tag foi substituída pela tag &lt;strong&gt; pois
12      além do efeito no texto também é considerada pelos motores de busca.
13    <br/>
14    <i>Texto itálico</i> Esta tag foi substituída pela tag &lt;em&gt; pois além
15      do efeito no texto também é considerada pelos motores de busca.
16    <br/>
17    Quebra de linha &lt;br&gt;;
18    <br/>
19    Linha orizontal &lt;hr&gt;;
20  </body>
21 </html>
```

Figura 23 Exemplo de Textos, tags `<p>`, ``, `<pre>`, ``, `<i>`, `
` e `<hr>`

O exemplo apresentado acima com tags de texto foi disponibilizado no arquivo **exemploTexto.html** do material do curso. Note que nas linhas 11, 13, 15 e 17 usamos símbolos HTML para exibir respetivamente ``, ``, `
` e `<hr/>`. Para saber demais símbolos HTML, visite o link abaixo:

<https://tutorialdehtml.com/pt/html-tabela-simbolos-caracteres/>

Listas

Podemos organizar listas em nossa página com as tags: ``, `` e `<dl>`. Vamos explorar cada uma delas:

Listas ordenadas = ``. Podemos escolher qual tipo de ordenação. Por padrão a ordenação será numérica. As listas ordenadas aceitam o atributo `type=` com valores diversos tais como `a`, `A`, `i`, `I`. Além disso podemos adotar o atributo `reversed`. Consulte o exemplo abaixo para conhecer as listas ordenadas.

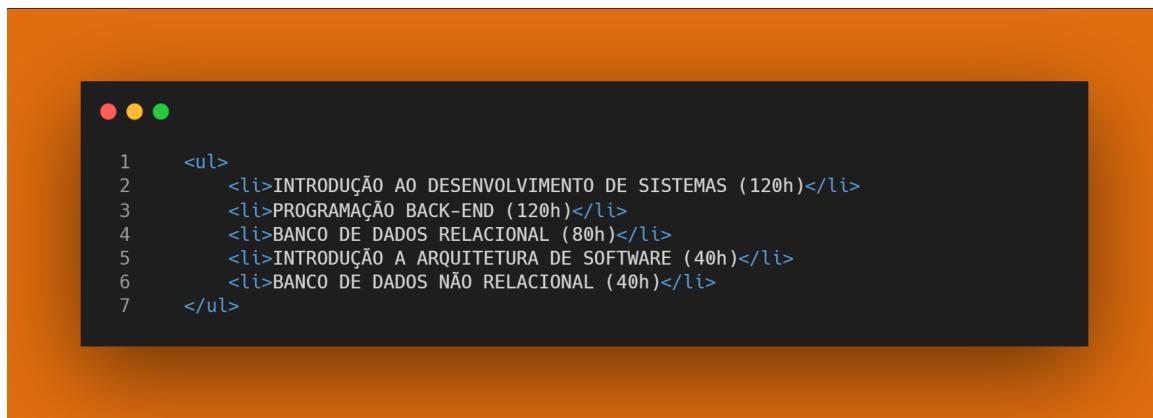


```
● ● ●

1 <ol>
2   <li>INTRODUÇÃO AO DESENVOLVIMENTO DE SISTEMAS (120h)</li>
3   <li>PROGRAMAÇÃO BACK-END (120h)</li>
4   <li>BANCO DE DADOS RELACIONAL (80h)</li>
5   <li>INTRODUÇÃO A ARQUITETURA DE SOFTWARE (40h)</li>
6   <li>BANCO DE DADOS NÃO RELACIONAL (40h)</li>
7 </ol>
8
9 <ol type="a">
10  <li>INTRODUÇÃO AO DESENVOLVIMENTO DE SISTEMAS (120h)</li>
11  <li>PROGRAMAÇÃO BACK-END (120h)</li>
12  <li>BANCO DE DADOS RELACIONAL (80h)</li>
13  <li>INTRODUÇÃO A ARQUITETURA DE SOFTWARE (40h)</li>
14  <li>BANCO DE DADOS NÃO RELACIONAL (40h)</li>
15 </ol>
16
17 <ol type="A">
18  <li>INTRODUÇÃO AO DESENVOLVIMENTO DE SISTEMAS (120h)</li>
19  <li>PROGRAMAÇÃO BACK-END (120h)</li>
20  <li>BANCO DE DADOS RELACIONAL (80h)</li>
21  <li>INTRODUÇÃO A ARQUITETURA DE SOFTWARE (40h)</li>
22  <li>BANCO DE DADOS NÃO RELACIONAL (40h)</li>
23 </ol>
24
25 <ol type="I">
26  <li>INTRODUÇÃO AO DESENVOLVIMENTO DE SISTEMAS (120h)</li>
27  <li>PROGRAMAÇÃO BACK-END (120h)</li>
28  <li>BANCO DE DADOS RELACIONAL (80h)</li>
29  <li>INTRODUÇÃO A ARQUITETURA DE SOFTWARE (40h)</li>
30  <li>BANCO DE DADOS NÃO RELACIONAL (40h)</li>
31 </ol>
32
33 <ol type="i" reversed>
34  <li>INTRODUÇÃO AO DESENVOLVIMENTO DE SISTEMAS (120h)</li>
35  <li>PROGRAMAÇÃO BACK-END (120h)</li>
36  <li>BANCO DE DADOS RELACIONAL (80h)</li>
37  <li>INTRODUÇÃO A ARQUITETURA DE SOFTWARE (40h)</li>
38  <li>BANCO DE DADOS NÃO RELACIONAL (40h)</li>
39 </ol>
40
```

Figura 24 Exemplo de Listas Ordenadas com a tag ``

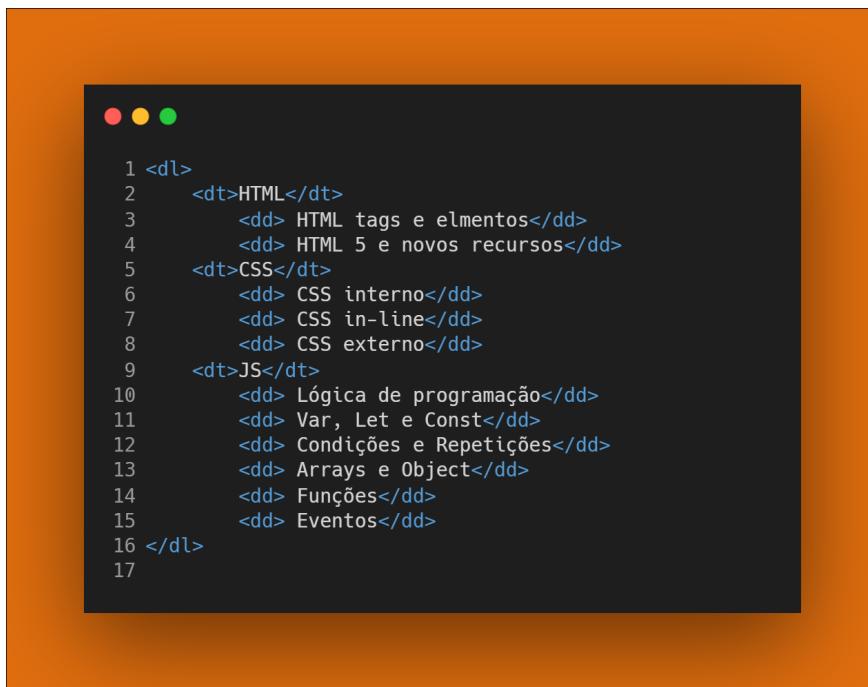
Além das listas ordenadas `` temos as listas desordenadas com a tag ``. Consulte o exemplo abaixo para conhecer as listas desordenadas.



```
1 <ul>
2   <li>INTRODUÇÃO AO DESENVOLVIMENTO DE SISTEMAS (120h)</li>
3   <li>PROGRAMAÇÃO BACK-END (120h)</li>
4   <li>BANCO DE DADOS RELACIONAL (80h)</li>
5   <li>INTRODUÇÃO A ARQUITETURA DE SOFTWARE (40h)</li>
6   <li>BANCO DE DADOS NÃO RELACIONAL (40h)</li>
7 </ul>
```

Figura 25 Exemplo Listas Desordenadas, com a tag

Temos ainda as listas por definição, que são representadas pela tag <dl>, seguido da tag <dt> e <dd>, confira abaixo um exemplo de lista por definição:



```
1 <dl>
2   <dt>HTML</dt>
3     <dd> HTML tags e elementos</dd>
4     <dd> HTML 5 e novos recursos</dd>
5   <dt>CSS</dt>
6     <dd> CSS interno</dd>
7     <dd> CSS in-line</dd>
8     <dd> CSS externo</dd>
9   <dt>JS</dt>
10    <dd> Lógica de programação</dd>
11    <dd> Var, Let e Const</dd>
12    <dd> Condições e Repetições</dd>
13    <dd> Arrays e Object</dd>
14    <dd> Funções</dd>
15    <dd> Eventos</dd>
16 </dl>
17
```

Figura 26 Exemplo de Lista por definição, com a tag <dl>, <dt> e <dd>

Os três exemplos apresentados acima com tags de listas foi disponibilizado no arquivo **exemploLista.html** do material do curso.

Iframes

A tag <iframe> permite inserirmos um conteúdo em nossa página, o objetivo desta tag é criar um frame interno. Geralmente o conteúdo vem de outro arquivo. Para informar a origem do conteúdo utilizamos o atributo src. É recomendado utilizar o atributo title para definir um título ao nosso frame. Confira abaixo um exemplo da tag iframe:



```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3   <head>
4     <meta charset="utf-8">
5     <title>Exemplo Iframe</title>
6   </head>
7   <body>
8     <iframe src="exemploLista.html" title="Iframe Listas"></iframe>
9   </body>
10 </html>
```

Figura 27 Exemplo da tag *Iframe*

Os Iframe são utilizados para interação de conteúdo em uma página e para segmentar nossa página em vários frames separados, por exemplo, podemos ter um frame para o menu e um frame para o conteúdo principal. Note que o exemplo apresentado acima tem a inserção da página “exemploLista.html”. O exemplo apresentado acima foi disponibilizado no arquivo **exemplolframes.html** do material do curso.

Âncoras

A tag `<a>`, que já usamos anteriormente sempre vem com o atributo `href`. Ela permite criar link para locais dentro do mesmo documento ou outras páginas do mesmo site e até mesmo páginas em outros sites. Vejamos os principais atributos desta tag.

href = indica qual página ou local vamos direcionar

target = indica onde vamos abrir o local indicado no `href`, as opções para o atributo `target` são: `_blank`, `_parent`, `_self` e `_top`

download = podemos definir o download do local indicado no atributo `href`, ao invés de abrir no navegador

rel = indica a relação que entre o documento e o local indicado no `href`, existem várias opções para o atributo `rel`, que veremos ao longo de nossa formação.

Existem outros atributos da tag `<a>` que vamos explorar quando for mais apropriado, caso desejar explorar ou outros atributos visite o site: https://www.w3schools.com/tags/tag_a.asp

Perceba que a tag `<a>` sempre tem um fechamento, isso significa que o conteúdo entre sua abertura e seu fechamento será o objeto que pode ser clicado, ou seja, o objeto “clicável” do nosso site. Confira exemplos da tag `<a>` abaixo:

```

1 <!DOCTYPE html>
2 <html lang="pt-BR">
3   <head>
4     <meta charset="utf-8">
5     <title>Exemplo de Âncoras</title>
6   </head>
7   <body>
8     <section>
9       <h1>Atributo href com símbolo # e atributo target="_self"</h1>
10      <p>Link para uma seção dentro do mesmo documento, neste caso no documento exemploAncora.html.</p>
11      <p>Perceba que estamos usando o sinal #. Estamos usando também o target="_self" para permanecer na
12        mesma aba no navegador.</p>
13      <p><a href="#agradecimentos" target="_self">Agradecimentos</a></p>
14    </section>
15    <section>
16      <h1>Atributo href e atributo target="_blank"</h1>
17      <p>Link para o site da GrowDev, abrindo em uma nova aba do navegador</p>
18      <p><a href="https://www.growdev.com.br/" target="_blank">Visite o site da GrowDev em nova aba</a></p>
19    </section>
20    <section>
21      <h1>Atributo Download </h1>
22      <p>Link para Download. Este exemplo funciona apenas se sua página estiver em um servidor Web. Veremos
23        detalhes sobre servidores ao lado de nossa formação.</p>
24      <p><a href=".//img.png" download="imagem.png" >Baixar imagem</a></p>
25    </section>
26    <section>
27      <h1>Juntando a tag de ancora com img para criar um link com imagem</h1>
28      <p>Link imagem para Download.</p>
29      <p><a href=".//img.png" download="imagem.png" ></a></p>
30    </section>
31    <p id="agradecimentos">Agradecemos aos alunos participantes do programa Starter Edição 1,2,3 e 4.</p>
32    Link para o site da GrowDev, abrindo na mesma aba do navegador, deixamos esta por último pois o visitante
33    do site vai abrir um novo local e não precisa mais ficar na página inicial.
34    <a href="https://www.growdev.com.br/">Visite o site da GrowDev nesta mesma aba</a>
35  </body>
36 </html>

```

Figura 28 Exemplo de Âncoras com tag <a> e seus atributos

Perceba que na âncora da linha 12 temos o símbolo # que indica uma seção definida pelo atributo id. Isso significa que dentro deste documento temos um elemento HTML com o atributo id="agradecimentos", como podemos observar na linha 29.

O exemplo apresentado acima com tag <a> foi disponibilizado no arquivo **exemploAncoras.html** do material do curso.

Links

Os links são criados com a tag <a>. Confira abaixo alguns exemplos de link.

```

1 <!DOCTYPE html>
2 <html lang="pt-BR">
3   <head>
4     <meta charset="utf-8">
5     <title>Exemplo Link</title>
6   </head>
7   <body>
8     <h2>Link para seção do mesmo documento</h2>
9     <p><a href="#rodape">Visitar o rodapé</a></p>
10    <p><a href="exemploLista.html">CSS Tutorial</a></p>
11
12    <h2>Link Absoluto, isto é, com URL completa</h2>
13    <p><a href="https://www.google.com/">Google</a></p>
14
15    <h2>Link relativo, isto é para arquivos no mesmo <b>local</b> do documento</h2>
16    <p><a href="exemploAncoras.html">Exemplo Âncoras</a></p>
17    <p><a href="exemploLista.html">Exemplo Listas</a></p>
18
19    <h2>Link para envio de e-mail, vai abrir o cliente de e-mail padrão do usuário</h2>
20    <p><a href="mailto:pessoa@servidor.com.br">Enviar e-mail</a></p>
21
22    <h2>Link com imagem </h2>
23    <p>
24      <a href="https://www.growdev.com.br">
25        
26      </a>
27    </p>
28
29    <h2>Link com Título </h2>
30    <p>deixe o mouse em cima do link para visualizar o título do link.</p>
31    <p>
32      <a href="https://www.growdev.com.br" title="Visitar site GrowDev">Visite nosso site.</a>
33    </p>
34
35    <p id="rodape">Grato por visitar o nosso site!</p>
36  </body>
37 </html>

```

Figura 29 Exemplos de Link

Imagens

Podemos inserir imagens em nossas páginas usando a tag `` como já estudamos anteriormente. Agora vamos explorar mais atributos desta tag.

```

1 <!DOCTYPE html>
2 <html lang="pt-BR">
3   <head>
4     <meta charset="utf-8">
5     <title>Exemplo Imagens</title>
6   </head>
7   <body>
8     
9     <br>
10    
11    <br>
12    
13  </body>
14 </html>

```

Figura 30 Exemplos da tag img

Existem outros atributos que vamos explorar ao longo do curso, caso desejar conhecê-los, visite o site: https://www.w3schools.com/tags/tag_img.asp

É possível exibir imagens através do arquivo binário, veja este exemplo: <https://html.com/attributes/img-src/>

Tag img no HTML 5: <https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/img>

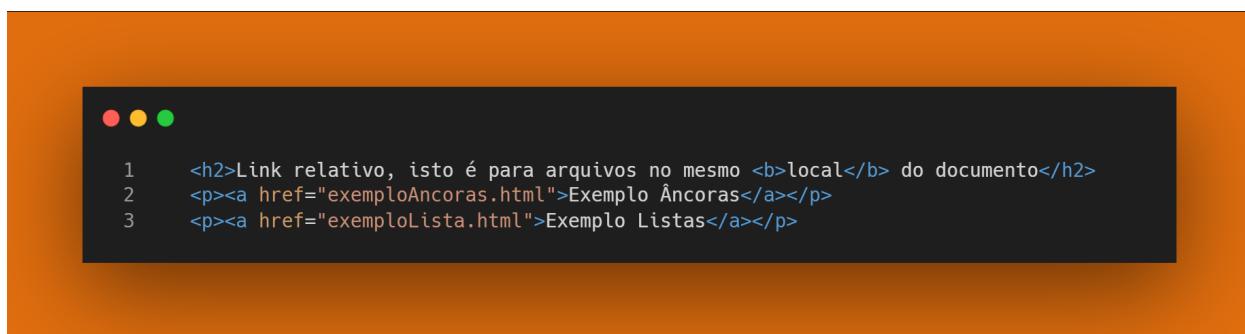
Como fechar a tag img?
<https://stackoverflow.com/questions/14860492/how-to-close-img-tag-properly>

URLs absolutas e relativas

A sigla URL significa o endereço virtual de uma página, do inglês *Uniform Resource Locator*. As URL absolutas e relativas já foram utilizadas no exemplo de Links. Agora vamos estudar com mais detalhe este conceito.

Definir o caminho absoluto significa indicar o caminho completo de um arquivo, seja ele html, jpg, png, pdf ou qualquer outro. Note que o caminho completo deve ser algo como href="https://www.google.com/"

Para abrir um arquivo HTML que está no mesmo local, isto é, na mesma pasta do arquivo, podemos indicar a URL relativa. Para entender este conceito considere o exemplo abaixo:



```
Link relativo, isto é para arquivos no mesmo <b>local</b> do documento</h2>
<p><a href="exemploAcoras.html">Exemplo Âncoras</a></p>
<p><a href="exemploLista.html">Exemplo Listas</a></p>
```

Figura 31 URL relativa

Considere que o arquivo com o código acima deve estar na mesma pasta dos arquivos exemploAcoras.html e exemploLista.html.

Para indicar caminhos podemos informar os nomes das pastas. Por exemplo imagine que em nosso site temos uma pasta chamada pagina_produto e dentro desta pasta temos outras páginas que desejamos criar link, a página importados.html e nacionais.html, para isso basta adicionar o caminho. Imagine também que temos uma pasta em um nível acima com o arquivo contato.html. Confira no exemplo abaixo os links relativos para as importados.html, nacionais.html e contato.html.



```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3   <head>
4     <meta charset="utf-8">
5     <title>Exemplo Relativo</title>
6   </head>
7   <body>
8     <h2>Link relativo, isto é considerando o mesmo <b>local</b> do documento</h2>
9     <p><a href="pagina_produto/importados.html">Importados</a></p>
10    <p><a href="pagina_produto/nacionais.html">Nacionais</a></p>
11    <p><a href="../contato.html">Contato</a></p>
12  </body>
13 </html>
```

Figura 32 Exemplo de URL Relativas

O exemplo apresentado acima com URL relativas foi disponibilizado na pasta **site** do material do curso. Note que estamos navegando pela estrutura de pastas de nosso site. Nos links das linhas 10 e 11 estamos entrando em uma pasta, já no link da linha 12 estamos subindo uma pasta. Dependendo da situação usamos caminhos absolutos ou relativos, veremos ao longo do curso a melhor abordagem de cada caso.

Tabelas

As tabelas são utilizadas para organizar o conteúdo de uma página. Para criar uma tabela simples usamos as seguintes tags `<table>` `<tr>` `<th>` e `<td>`. Vamos construir uma tabela com os dias da semana, isto é, cinco colunas (segunda a sexta) e duas linhas.



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Exemplo Tabela</title>
6 </head>
7 <body>
8   <table border="1">
9     <tr>
10       <th>Seg</th>
11       <th>Ter</th>
12       <th>Qua</th>
13       <th>Qui</th>
14       <th>Sex</th>
15     </tr>
16     <tr>
17       <td>limpar o quarto</td>
18       <td>Participar da aula</td>
19       <td>Ir no mercado</td>
20       <td>Participar da aula</td>
21       <td>Limpar o banheiro</td>
22     </tr>
23   </table>
24 </body>
25 </html>
```

Figura 33 Exemplo de Tabela, tag table

Note que a tabela inicia na linha 8 e finaliza na linha 23. Como desejamos duas linhas abrimos a tag `<tr>` na linha 9 e na linha 16. Cada uma das tags `<tr>` define uma linha em nossa tabela. A primeira linha é nosso cabeçalho, por isso cada célula foi criada com a tag `<th>`. Note que definimos o atributo *border* para que pudéssemos identificar a tabela no navegador. O exemplo apresentado acima com tag `<table>` foi disponibilizado no arquivo **exemploTabela.html** do material do curso. Explore o código do arquivo **exemploTabela.html** e implemente modificações, criando mais linhas e colunas.

Agora vamos explorar mais atributos e tags relacionadas as tabelas HTML

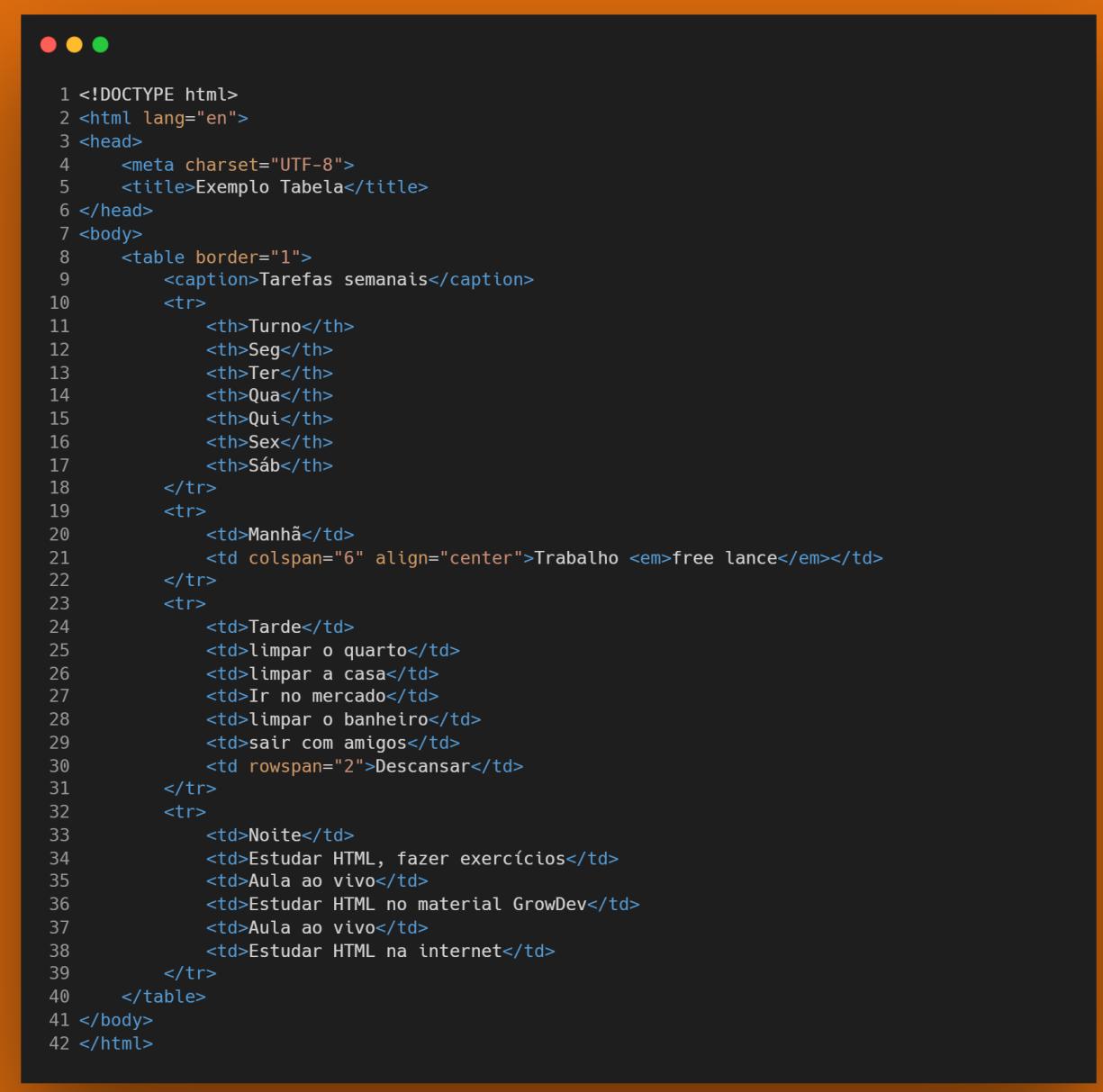
Tag **<caption>** é utilizada para definir um título para tabela

rowspan é um atributo que pode ser colocado na tag `<td>` utilizado para mesclar duas ou mais linhas, no exemplo vamos juntar a linha do turno da manhã com o turno da noite.

colspan é um atributo que pode ser colocado na tag `<td>` utilizado para mesclar colunas, no exemplo vamos juntar as colunas dos 6 dias definidos em nossa tabela (segunda a sábado).

aling e **valign** são atributos para definir o alinhamento horizontal e vertical respectivamente.

Confira no exemplo abaixo a tabela com as tags caption, rowspan, colspan e align.



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Exemplo Tabela</title>
6 </head>
7 <body>
8   <table border="1">
9     <caption>Tarefas semanais</caption>
10    <tr>
11      <th>Turno</th>
12      <th>Seg</th>
13      <th>Ter</th>
14      <th>Qua</th>
15      <th>Qui</th>
16      <th>Sex</th>
17      <th>Sáb</th>
18    </tr>
19    <tr>
20      <td>Manhã</td>
21      <td colspan="6" align="center">Trabalho <em>free lance</em></td>
22    </tr>
23    <tr>
24      <td>Tarde</td>
25      <td>limpar o quarto</td>
26      <td>limpar a casa</td>
27      <td>Ir no mercado</td>
28      <td>limpar o banheiro</td>
29      <td>sair com amigos</td>
30      <td rowspan="2">Descansar</td>
31    </tr>
32    <tr>
33      <td>Noite</td>
34      <td>Estudar HTML, fazer exercícios</td>
35      <td>Aula ao vivo</td>
36      <td>Estudar HTML no material GrowDev</td>
37      <td>Aula ao vivo</td>
38      <td>Estudar HTML na internet</td>
39    </tr>
40  </table>
41 </body>
42 </html>
```

Figura 34 Exemplo de Tabela com tags `caption`, `rowspan`, `colspan` e `align`

O exemplo apresentado acima com a tag `<table>`, suas demais tags e atributos foi disponibilizado no arquivo **exemploTabelaTags.html** do material do curso.

Links de referência

Resumo das tags HTML

<https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element>