



# Introdução à HTML

Prof. Jesiel Viana

# Roteiro

---

- Introdução
- Elementos HTML
- Estrutura do HTML
- Doctype
- Encoding
- Atributos de elementos
- HTML Semântico

# Pré-requisitos

---

Básico de informática, softwares básicos (**VS Code e Chrome**) instalado e conhecimento básico de como trabalhar com arquivos.

# Objetivos

---

Obter uma familiaridade básica com o HTML e adquirir um pouco de prática escrevendo alguns elementos HTML.

# Web page

```
graph TD; WP[Web page] --- HTML[HTML]; WP --- CSS[CSS]; WP --- JS[Javascript]; HTML --- CS[Content & Structure]; CS --- P[Paragraphs]; CS --- L[Lists]; CS --- T[Tables]; CS --- TX[Texts]; CS --- I[Images]; CSS --- PR[Presentation]; PR --- BC[Background color]; PR --- CO[Color]; PR --- F[Font]; PR --- B[Border]; PR --- A[Alignment]; JS --- BE[Behavior]; BE --- DD[Dynamic display]; BE --- UI[User iteration]; BE --- DM[DOM manipulation];
```

## HTML

### Content & Structure

Paragraphs  
Lists  
Tables  
Texts  
Images

## CSS

### Presentation

Background color  
Color  
Font  
Border  
Alignment

## Javascript

### Behavior

Dynamic display  
User iteration  
DOM manipulation

# HTML

---

Hypertext Markup Language (HTML) é o código que você usa para estruturar seu conteúdo web, dando significado e propósito. Por exemplo, meu conteúdo é organizado em parágrafos, ou em uma lista de pontos? Eu tenho imagens na minha página? Eu tenho uma tabela de dados?

HTML não é realmente uma linguagem de programação; é uma linguagem de marcação, usada para dizer ao navegador como exibir as páginas que são visitadas. Ela consiste de uma série de **elementos**, que você junta ao redor de diferentes partes do seu conteúdo para que ele apareça ou aja de um determinado jeito.

# Por exemplo:

---

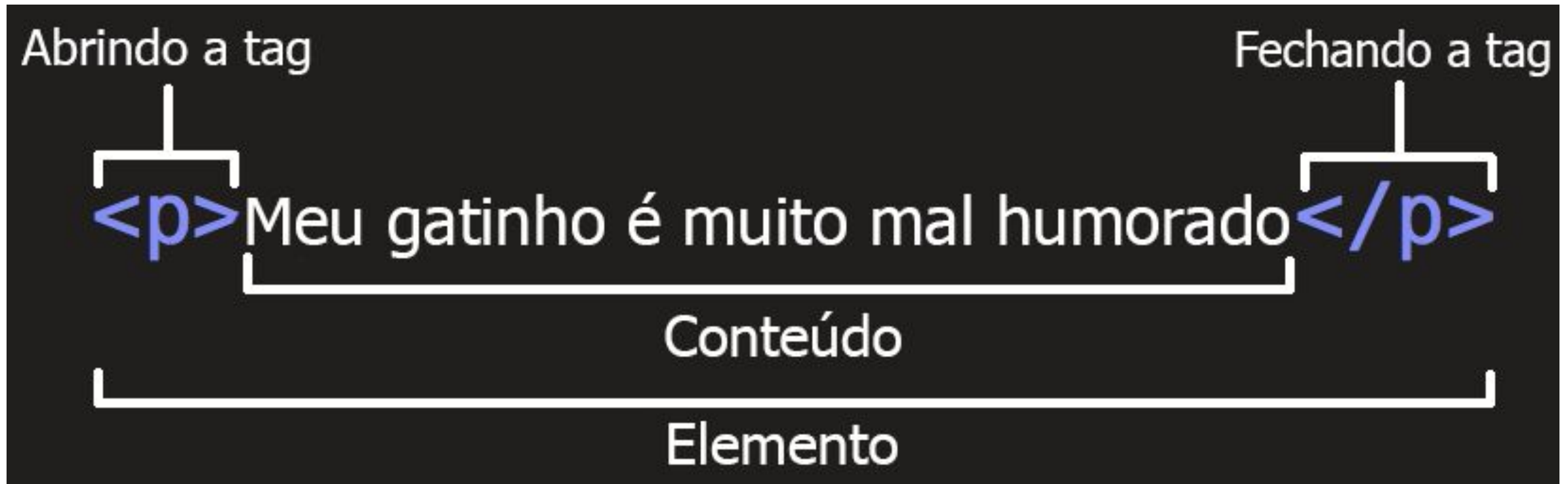
Meu gatinho é muito mal humorado

No HTML se nós queremos oficialmente dizer que isso é um parágrafo, nós podemos envolver o elemento em um código ([<p>](#)):

```
<p>Meu gatinho é muito mal humorado</p>
```

# Anatomia de um elemento HTML

---





# Anatomia de um elemento HTML

---

As principais partes de um elemento são:

**Tag de abertura:** Consiste no nome do elemento (no caso, p), envolvido em chaves de abertura e fechamento. Isso demonstra onde o elemento começa, ou onde seu efeito se inicia — nesse caso, onde é o começo do parágrafo.

**Tag de fechamento:** Isso é a mesma coisa que a tag de abertura, exceto que inclui uma barra antes do nome do elemento. Isso demonstra onde o elemento acaba — nesse caso, onde fica o fim do parágrafo. Esquecer de incluir uma tag de fechamento é um dos erros mais comuns de iniciantes, e pode levar a resultados estranhos.


**Conteúdo:** Esse é o conteúdo do elemento, que nesse caso é só texto.

**Elemento:** A tag de abertura + tag de fechamento + conteúdo formam o elemento.

## Elementos também podem ter atributos, conforme abaixo:

Attribute

```
<p class="editor-note">My cat is very grumpy</p>
```



Atributos contém informação extra sobre o elemento, que você não quer que apareça no elemento de fato. Nesse caso, o atributo classe (**class**) permite que você dê ao elemento um nome de identificação (editor-note), que depois pode ser usado para dar ao elemento informações de estilo ou outras coisas.

Um atributo sempre deve ter:

1. Um espaço entre ele e o nome do elemento (ou o atributo anterior).
2. O nome do atributo, seguido por um sinal de igual
3. Aspas de abertura e fechamento, envolvendo todo o valor do atributo

# Quiz 1

---

## HTML é?

1. Uma linguagem de programação.
2. Uma linguagem de marcação.



# Elementos vazios

---

Alguns elementos não tem conteúdo, e são chamados elementos vazios.

Considere o elemento `<img>` que temos na nossa HTML:

```

```

Ele contém dois atributos, mas não há tag `</img>` de fechamento, e não há conteúdo interno. Isso acontece porque uma imagem não envolve conteúdo para ter efeito em si mesma. Sua proposta é incorporar uma imagem na página HTML, no lugar que o código aparece.

# Quiz 2

---

**Todo elemento HTML deve ter:**

1. Uma tag de abertura.
2. Uma tag de fechamento.
3. Um conteúdo.
4. Um atributo.



# Elementos aninhados

---

Você pode colocar elementos dentro de outros elementos também — isso é chamado aninhar (nesting). Se quisermos declarar que o gato é MUITO rabugento, podemos envolver a palavra "muito" em um `<strong>`, o que significa que a palavra deve ser fortemente enfatizada:

```
<p>Meu gatinho é <strong>muito</strong> mal humorado.</p>
```

Contudo, você precisa ter certeza que seus elementos estão devidamente aninhados: no exemplo acima, nós abrimos o elemento `<p>` primeiro, depois o `<strong>`, então temos que fechar o `</strong>` primeiro e depois o `</p>`.

# Estrutura semântica de uma página HTML

# Exemplo de documento HTML

```
<!DOCTYPE html>
<html>

  <head>
    <meta charset="UTF-8">
    <title>Título da Página</title>
  </head>

  <body>
    <h1>Meu primeiro Cabeçalho</h1>
    <p>Meu Primeiro Parágrafo</p>
  </body>

</html>
```

- `<!DOCTYPE html>` declaração que define que este documento é HTML5
- `<html>` é o elemento raiz de uma página HTML
- `<head>` elemento que contém meta informações sobre o documento
- `<meta charset>` especifica a codificação de caracteres para o documento HTML.
- `<title>` elemento que especifica um título para o documento
- `<body>` elemento que contém o conteúdo da página visível
- `<h1>` elemento que define um grande título
- `<p>` elemento que define um parágrafo



```
<html>
```

```
<head>
```

```
<title>Page title</title>
```

```
</head>
```

```
<body>
```

```
<h1>This is a heading</h1>
```

```
<p>This is a paragraph.</p>
```

```
<p>This is another paragraph.</p>
```

```
</body>
```

```
</html>
```

# DOCTYPE

---

- O DOCTYPE não é uma tag HTML, mas uma instrução especial. Ela indica para o navegador qual versão do HTML deve ser utilizada para renderizar a página. Utilizaremos `<!DOCTYPE html>`, que indica para o navegador a utilização da versão mais recente do HTML - a versão 5, atualmente.
- Versões anteriores do HTML e XHTML tinham diferentes DOCTYPE.
- Importante sempre usar a última versão do HTML.

# Encoding

---

- HTML <meta> charset atributo é usado para indicar o formato de codificação de caracteres usado no documento HTML, *sempre deve ser colocada dentro do <head>.*
- Para exibir uma página HTML corretamente, um navegador da web deve saber qual o conjunto de caracteres (codificação de caracteres) a ser usado.
- Em 2014 foi constatado que cerca de 83,6% das páginas web estavam utilizando o encode **UTF-8** (4).
- Exemplo de uso: <meta charset="UTF-8">

# Até agora...

Todo documento HTML deve começar com uma declaração, qual?

```
<!DOCTYPE html>
```

Qual a tag inicial de um documento HTML?

```
<html>
```

Qual tag define a parte visível de um documento HTML?

```
<body>
```



# Use o inspetor de código do seu navegador

Para visualizar a estrutura da página codificada.

# HTML Semântico

---

Nas versões anteriores do HTML não haviam tags com uma semântica apropriada para cada uma das divisões de uma página web. Dessa forma, os desenvolvedores usavam a tag `<div>` para todas as situações, e criando seus próprios padrões de nomenclaturas através dos atributos `id` ou `class` [\[3\]](#).

# HTML Semântico

---

Semântica é o estudo dos significados de palavras e frases em uma linguagem.

Elementos semânticos = elementos com um significado.

Alguns elementos semânticos:

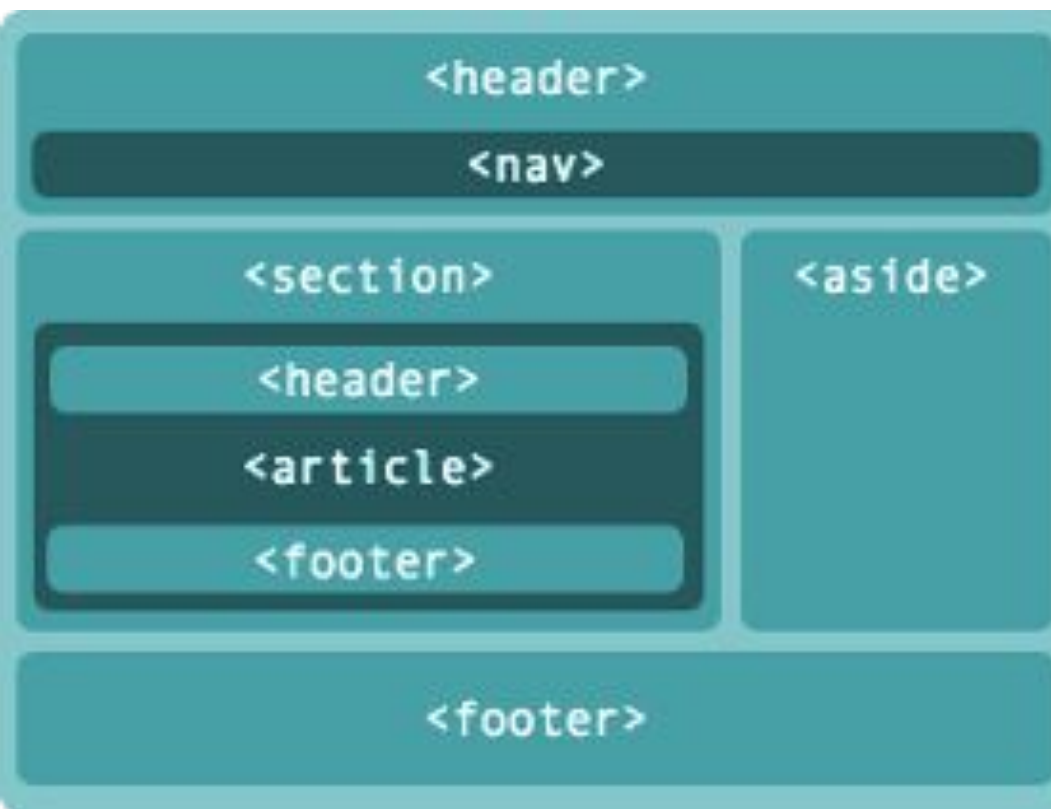
- `<header>` - Define o cabeçalho de um documento ou seção
- `<main>` - Define o conteúdo principal de um documento
- `<nav>` - Define os links de navegação (Menu)
- `<section>` - Define uma seção no documento
- `<article>` - Define um artigo
- `<aside>` - Define um conteúdo lateral
- `<footer>` - Define o rodapé de um documento ou seção

A partir do HTML5 foram criadas diversas tags semânticas para indicar aos user-agents quais conteúdos estão sendo inseridos em cada uma das divisões da página, organizando e padronizando o desenvolvimento [\[3\]](#).

#### HTML 4



#### HTML 5





# Faça a marcação semântica correta de seu código HTML

1. Comunica a intenção do código (pra você e outros)
2. Acessibilidade (leitores de tela)
3. Rastreabilidade (mecanismos de busca)

# Bibliografia

---

- [1] - CAELUM. Desenvolvimento Web com HTML, CSS e JavaScript. Disponível em: <<https://www.caelum.com.br/apostila-html-css-javascript>>. Acesso em: 17 maio 2016.
- [2] - W3SCHOOLS. HTML(5) Tutorial. Disponível em: <<http://www.w3schools.com>>. Acesso em: 17 maio 2016.
- [3] - <http://www.uxdesign.blog.br/front-end/html5-estrutura-semantica/>
- [4] - <https://tableless.com.br/html-encode-utf-8/>
- [5] - [https://developer.mozilla.org/pt-BR/docs/Aprender/Getting\\_started\\_with\\_the\\_web/HTML\\_basico](https://developer.mozilla.org/pt-BR/docs/Aprender/Getting_started_with_the_web/HTML_basico)



 [jesiel@ifpi.edu.br](mailto:jesiel@ifpi.edu.br)

 [@jesielviana](https://twitter.com/jesielviana)

 <https://github.com/prof-jesielviana>

# Obrigado!!!