

# Desenvolvimento Web

## Aula 3

---

Eduardo Mangeli

26 de agosto de 2024

# Agenda

1. Introdução aos Elementos HTML
2. Estrutura de um documento HTML
  - Raiz principal
  - Metadados
  - Raiz do conteúdo
3. Organização de conteúdo
4. Conteúdo
  - Listas
5. Exercícios Parte I
6. Marcação semântica de texto
  - Marcando Edições
7. Mídia
8. Formulários
9. Atributos
10. Referência
11. Exercício

# Introdução aos Elementos HTML

---

## Considerações iniciais

- CASE sensitive?

# Considerações iniciais

- CASE sensitive?
  - no conteúdo

# Considerações iniciais

- CASE sensitive?
  - no conteúdo
  - no nome do arquivo

# Considerações iniciais

- CASE sensitive?
  - no conteúdo
  - no nome do arquivo
- `<!DOCTYPE html>`

# Considerações iniciais

- CASE sensitive?
  - no conteúdo
  - no nome do arquivo
- `<!DOCTYPE html>`
- o comentário `<!-- -->`



# Considerações iniciais

- CASE sensitive?
  - no conteúdo
  - no nome do arquivo
- `<!DOCTYPE html>`
- o comentário `<!-- -->`
- nota sobre editores de texto

# Considerações iniciais

- CASE sensitive?
  - no conteúdo
  - no nome do arquivo
- `<!DOCTYPE html>`
- o comentário `<!-- -->`
- nota sobre editores de texto
- W3C especifica 108 elementos

# Considerações iniciais

- CASE sensitive?
  - no conteúdo
  - no nome do arquivo
- `<!DOCTYPE html>`
- o comentário `<!-- -->`
- nota sobre editores de texto
- W3C especifica 108 elementos
- o que são tags?

# Considerações iniciais

- CASE sensitive?
  - no conteúdo
  - no nome do arquivo
- `<!DOCTYPE html>`
- o comentário `<!-- -->`
- nota sobre editores de texto
- W3C especifica 108 elementos
- o que são tags?
- o que são elementos?

# Considerações iniciais

- CASE sensitive?
  - no conteúdo
  - no nome do arquivo
- `<!DOCTYPE html>`
- o comentário `<!-- -->`
- nota sobre editores de texto
- W3C especifica 108 elementos
- o que são tags?
- o que são elementos?
- o que são atributos?

# Estrutura de um documento HTML

---

# Estrutura de um documento HTML

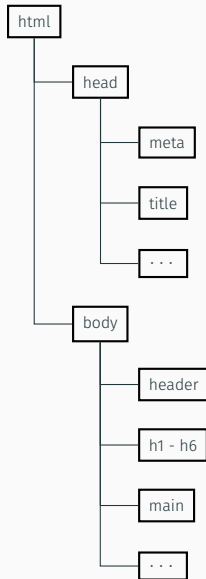
---

Raiz principal

# Nó raiz

`<html>`

é o elemento raiz (*root*) de um documento HTML. Todos os outros elementos devem ser seus descendentes.





# Estrutura de um documento HTML

---

Metadados

`<head>` é o elemento que contém as informações *gerais* do documento, como o título, conjunto de caracteres, folhas de estilo e scripts.

Deve conter, no mínimo, os elementos `<title>` e `<meta>`.

`<title>` é o elemento que contém o título do documento, que é exibido na barra de títulos do navegador web e usado para referenciar a página em mecanismos de busca, por exemplo.

Aceita apenas texto.

`<meta>` é o elemento utilizado para representar outros metadados ou informações adicionais do documento, como o conjunto de caracteres ou o nome do author, por exemplo.

```
1  <!DOCTYPE html>
2  <html lang="pt-br">
3      <head>
4          <title>Minha página</title>
5          <meta charset="utf-8">
6      </head>
7      <body>
8          <h1>Página web</h1>
9          <p>Minha página web</p>
10         <hr>
11         <!-- hr é um elemento vazio -->
12         <p>Com dois parágrafos</p>
13     </body>
14 </html>
```

# Estrutura de um documento HTML

---

Raiz do conteúdo

`<body>` é o elemento que representa o conteúdo do documento HTML.

Só pode haver um único `<body>` em cada documento.

# Organização de conteúdo

---



`<h1>-<h6>` são elementos que representam cabeçalhos. São utilizados para marcar títulos e subtítulos dos documentos.

`<h1>` representa os cabeçalhos de nível mais alto e `<h6>` representa os cabeçalhos de nível mais baixo.

## Outros elementos de organização de conteúdo

- `<address>`
- `<article>`
- `<aside>`
- `<footer>`
- `<header>`
- `<main>`
- `<nav>`
- `<section>`

# Conteúdo

---

`<p>` representa um parágrafo. Geralmente são representados em **mídias visuais** como blocos de texto separados dos seus vizinhos por linhas em branco ou indentação da primeira linha.

`<div>` é um contêiner genérico usado para controlar o fluxo do conteúdo. Ele não tem efeito imediato no conteúdo ou no estilo do documento e necessita ser “estilizado” com o uso de CSS.

`<span>` é um contêiner genérico *inline*. Ele pode ser usado para agrupar elementos para estilização em conjunto ou quando esses elementos compartilham alguma característica e outro elemento não seria adequado.

# Conteúdo

---

Listas

`<ul>` é o elemento que representa uma lista que não apresenta ordenação.



`<ol>` é o elemento que representa uma lista que apresenta sua ordenação.

`<li>` é o elemento usado para representar um item de uma lista ordenada ou sem ordenação.

## Exemplo de listas

```
1  <ul>
2    <li>Um ítem</li>
3    <li>Outro ítem</li>
4    <li>Mais um item
5      <ol>
6        <li>Primeiro item</li>
7        <li>Segundo item</li>
8      </ol>
9    </li>
10 </ul>
```

`<menu>` é uma alternativa semântica para o `<ul>`. Ele representa uma lista sem ordenação na qual cada item (`<li>`) representa um *link* para um comando que pode ser ativado pelo usuário.

## Outros elementos de conteúdo

- `<blockquote>`
- `<figcaption>`
- `<figure>`
- `<hr>`
- `<pre>`
- lista de definições
  - `<dl>` lista de definições
  - `<dt>` itens
  - `<dd>` definições

## Outros elementos de conteúdo

- `<blockquote>`
- `<figcaption>`
- `<figure>`
- `<hr>`
- `<pre>`
- lista de definições
  - `<dl>` lista de definições
  - `<dt>` itens
  - `<dd>` definições

```
<dl>  
  <dt>Notebook</dt>  
    <dd>Mobilidade</dd>  
    <dd>Conforto</dd>  
  
  <dt>Desktop</dt>  
    <dd>Desempenho</dd>  
</dl>
```

## Exercícios Parte I

---

1. Crie uma página **html** com cada um dos elementos principais listados na aula. Não é necessário incluir o “*outros*” elementos.
2. Use as ferramentas de desenvolver do seu navegador para ver o seu código.
3. Utilize as ferramentas de desenvolvedor do seu navegador para mudar o seu código
4. Organize seus exemplos e envie para o seu repositório no github



# Marcação semântica de texto

---

`<a>` é um elemento *âncora* que cria um *hyperlink* para uma página web, um arquivo, um endereço de email, uma posição na mesma página ou qualquer outra coisa que possa ser apontada por uma URL.

```
<a href="http://mangeli.com.br">Mangeli</a>
```

`<strong>` indica conteúdo de grande importância, relevância ou urgência. Os navegadores tipicamente exibem o conteúdo em negrito.

O uso da tag `<b>` é desencorajado.

`<em>` marca um elemento que deve receber ênfase. Os navegadores geralmente exibem o conteúdo em itálico.

O uso da tag `<i>` é desencorajado.

# Marcação semântica de texto

---

Marcando Edições

`<del>` é o elemento utilizado para demarcar texto que foi apagado do documento. É usado para manter o registro de edições. Os navegadores costumam indicar o texto ~~tachado~~. O uso da tag `<s>` é desencorajado.

`<ins>` é o elemento utilizado para demarcar texto que foi inserido no documento. É usado para manter o registro de edições. Os navegadores costumam indicar o texto sublinhado. O uso da tag `<u>` é desencorajado.

## Outros elementos de marcação semântica

- `<abbr>`
- `<bdi>`
- `<bdo>`
- `<br>`
- `<cite>`
- `<code>`
- `<data>`
- `<dfn>`
- `<kbd>`
- `<mark>`
- `<q>`



Mídia

---

O atributo importante para essas tags é o **src**.

- `<img>`
- `<audio>`
- `<video>`

# Formulários

---

`<form>` é o elemento que representa a seção do documento que contém os controles e elementos de um formulário.

`<button>` é um elemento interativo que pode ser ativado pelo usuário usando o mouse, o teclado, o dedo, um comando de voz ou outra tecnologia. Quando ativado ele realiza alguma ação como enviar um formulário.

`<input>` é o elemento utilizado para capturar entradas do usuário. É um dos elementos mais poderosos da HTML devido ao grande número de possibilidades de combinação de tipos de entrada e atributos.

Exemplos: <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input>

`<textarea>` permite a entrada de grande quantidade de texto, em múltiplas linhas.

`<select>` é o controle que fornece uma lista de opções que são representadas pela tag `<option>`



## Outros elementos de formulários

- `<datalist>`
- `<fieldset>`
- `<label>`
- `<legend>`
- `<meter>`
- `<optgroup>`
- `<progress>`

```
1  <!-- Envia uma requisição GET para a URL atual -->
2  <form method="get">
3      <label>Name:
4          <input name="submitted-name" autocomplete="name">
5      </label>
6      <button>Save</button>
7  </form>
8
9  <!-- Envia uma requisição POST para a URL atual -->
10 <form method="post">
11     <label>Name:
12         <input name="submitted-name" autocomplete="name">
13     </label>
14     <button>Save</button>
15 </form>
16
17 <!-- Exemplos dos elementos fieldset, legend e label -->
18 <form method="post">
19     <fieldset>
20         <legend>Title</legend>
21         <label><input type="radio" name="radio"> Select me</label>
22     </fieldset>
23 </form>
```

# Atributos

---

onabort, onautocomplete, onautocompleteerror, onblur, oncancel, oncanplay, oncanplaythrough, onchange, onclick, onclose, oncontextmenu, oncuechange, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, ondurationchange, onemptied, onended, onerror, onfocus, oninput, oninvalid, onkeydown, onkeypress, onkeyup, onload, onloadeddata, onloadedmetadata, onloadstart, onmousedown, onmouseenter, onmouseleave, onmousemove, onmouseout, onmouseover, onmouseup, onmousewheel, onpause, onplay, onplaying, onprogress, onratechange, onreset, onresize, onscroll, onseeked, onseeking, onselect, onshow, onsort, onstalled, onsubmit, onsuspend, ontimeupdate, ontoggle, onvolumechange, onwaiting

[https://developer.mozilla.org/en-US/docs/Web/Events/Event\\_handlers](https://developer.mozilla.org/en-US/docs/Web/Events/Event_handlers)

- class
- id
- name
- hidden
- style
- title

## Referência

---

[https://developer.mozilla.org/en-US/docs/Web/  
HTML/Element](https://developer.mozilla.org/en-US/docs/Web/HTML/Element)

# Exercício

---



1. Crie uma página **html** com cada um dos elementos principais listados na aula. Não é necessário incluir o “*outros*” elementos.
2. Use as ferramentas de desenvolver do seu navegador para ver o seu código.
3. Implemente formulários e faça com que o seu conteúdo seja enviado (DICA: **submit**)
  - Faça ao menos um formulário com o método GET
  - Faça ao menos um formulário com o método POST
  - Observe as requisições realizadas pelos formulários na ferramenta de desenvolvedor do seu navegador
4. Utilize as ferramentas de desenvolvedor do seu navegador para mudar o seu código
5. Implemente um elemento **a** que seja um link interno da própria página
6. Implemente um elemento **a** que seja um link para outra página
7. Organize seus exemplos e envie para o seu repositório no github