

An abstract network diagram on the left side of the slide. It features a collection of nodes represented by circles in various colors: bright blue, dark blue, green, brown, and orange. These nodes are interconnected by a web of thin, light blue lines, creating a complex, interconnected pattern that suggests a network or a web structure.

# Front-end Essencial HTML

Prof. Yuri Weilemann

# O que é HTML?



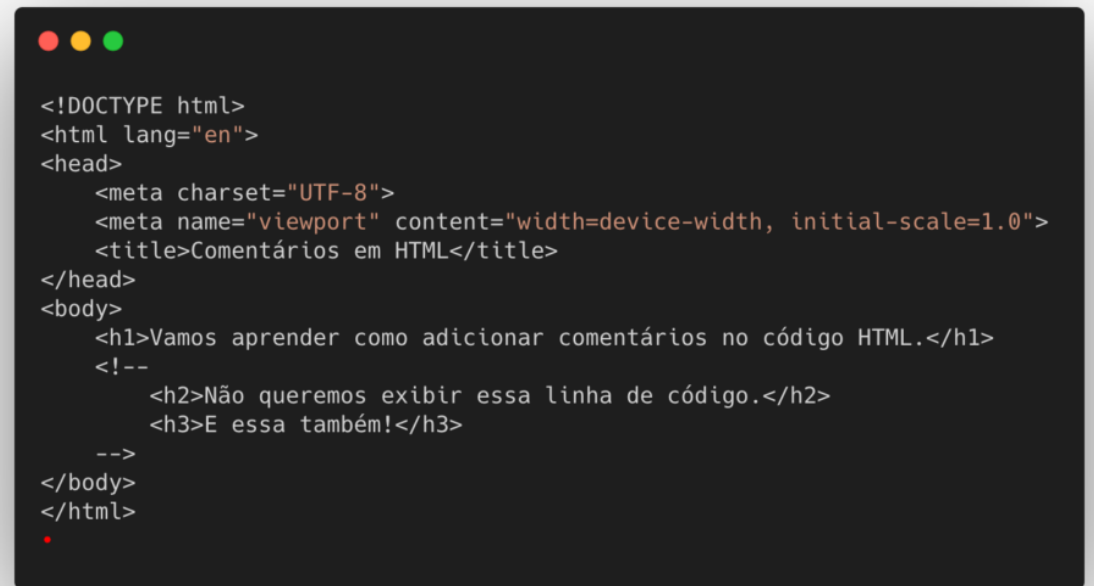
HTML significa Hypertext Markup Language. Diferentemente do Javascript, C# e outras linguagens de programação, o HTML é uma linguagem de marcação.

HTML nada mais é do que um "jeito de escrever". Tem semântica, sintaxe, etc, para marcar textos. É a única linguagem que o navegador consegue interpretar para exibição de conteúdo.

# Comentários no Código

É sempre importante adicionar comentários aos códigos para que lembremos o porquê de determinadas funções, trechos de códigos e afins serem necessários, tal como para fins estudantis também.

Adicionamos comentários utilizando o símbolo `<!-- -->`. É importante ser cauteloso com a inclusão de comentários pois embora estes comentários não sejam exibidos em tela, eles serão acessíveis através do código fonte.

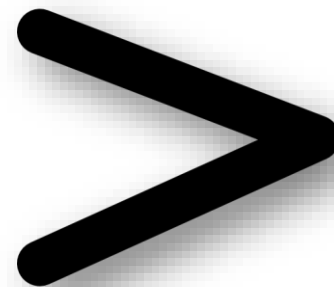
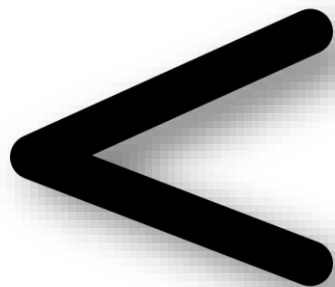


```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Comentários em HTML</title>
</head>
<body>
  <h1>Vamos aprender como adicionar comentários no código HTML.</h1>
  <!--
    <h2>Não queremos exibir essa linha de código.</h2>
    <h3>E essa também!</h3>
  -->
</body>
</html>
```

# Anatomia das Tags

O que constitui a Markup são as tags. Se quisermos escrever um título, por exemplo, adicionaremos uma tag `<h1> </h1>`. Notem que na segunda parte da tag há uma barra ( / ) simbolizando o fechamento daquele elemento.

Existem também os elementos vazios, por exemplo uma imagem `<img>`. Estes elementos podem vir acompanhados com a formatação de "self-closings tags" (`<img />`), porém não é algo obrigatório no HTML.



# Atributos

Elementos HTML possuem também atributos. Atributos servem como informações extras para colocarmos no elemento ou como configurações.

Existem também os **Atributos Booleanos**, como por exemplo o "disabled" nas tags inputs.

```
<input type="text" disabled>
```

# Aspas

Outra questão importante são as aspas. As aspas são utilizadas para inserir cadeias de caracteres dentro delas, de forma que o valor seja compreensível. Basicamente, podemos ter 3 "tipos de aspas":

- Simples - ' '
- Duplas - " "
- Omissas

A large, bold, black graphic of a double quotation mark, consisting of two opening quotes (‘ ’) and two closing quotes (” ”), arranged in a pair.

# Aspas Simples

As aspas simples ( ' ') podem ser utilizadas para inserção de um ou mais caracteres. Entretanto, ela apresenta um problema quando há a necessidade do uso de apóstrofes nessa mesma cadeia de caracteres, como por exemplo:

```
<a href="" title='This is not a description'>Link</a>  
<a href="" title='This isn't a description'>Link</a>
```

# Aspas Duplas

As aspas duplas (" ") podem ser utilizadas também para a inserção de um ou mais caracteres. Este tipo de aspas resolve o problema apresentado anteriormente com as aspas simples. O uso de aspas duplas tende a ser a melhor opção de modo geral e acaba por ser considerado uma boa prática.

```
<a href="" title="This is not a description">Link</a>  
<a href="" title="This isn't a description">Link</a>
```



# Omissas

Em uma tag de âncora, por exemplo, é possível definirmos uma string sem o uso das aspas para referenciar um endereço web.

Entretando, essa não é uma medida recomendada, porque ao adicionar outros atributos a tag, o browser pode interpretar de forma errônea causando um comportamento indesejado. Cada browser interpreta o HTML de uma forma, por isso é recomendado o uso de aspas mesmo nestes casos.

```
<a href=https://www.facebook.com>Facebook</a>
```

# Atributos Globais

Os atributos globais são atributos que podem ser atribuídos a qualquer elemento HTML. Para sabermos quais são os atributos globais é bem simples, basta criarmos uma tag qualquer e pressionar Control + Espaço dentro da tag de abertura. Alguns dos atributos globais mais utilizados são:

- class
- contenteditable
- hidden
- id
- style
- tabindex
- title

```
<div class="container"></div>
```

# Atributos Globais

- **class** - Define uma classe para um ou mais elementos. Essa classe será utilizada para estilização destes elementos e também para a inserção de funções no javascript.
- **contenteditable** - Permite tornar o conteúdo daquela tag editável para o usuário.
- **hidden** - Atributo booleano que esconde o elemento HTML.
- **id** - Define um id para um elemento. Esse id será utilizado para estilização destes elementos e também para a inserção de funções no javascript.
- **style** - Permite adicionar estilizações aos elementos HTML diretamente na tag.
- **tabindex** - Insere a ordem de transição de elementos pelo pressionamento da tecla TAB.
- **title** - Define um título para quando o mouse descansar em cima do elemento.

Mais informações sobre os atributos globais [aqui](#).

# Aninhamento Hierárquico

Podemos aninhar tags dentro de outras tags. Por exemplo:

```
<div>  
  <p>Olá, me chamo <strong>Yuri Weilemann</strong></p>  
</div>
```

Neste exemplo, a tag `<strong>` é uma tag filha da tag `<p>`. A tag `<p>` é filha da tag `<div>`.

# Conteúdo do texto e caracteres reservados

Perceba que o HTML não leva em consideração os espaços e as quebras de linha que inserimos no editor:

```
<p>Testando  
    essa  
parada de  
quebrar      as linhas  
e          adicionar um      monte de espaço  
</p>
```

Isso se dá pela própria interpretação do HTML. Se eu quiser adicionar espaços, efetivamente, eu preciso utilizar caracteres especiais, por exemplo, `&nbsp;` ou uma tag `<br>` no caso de quebras de linha.

O HTML possui alguns caracteres reservados como `<`, `"`, `>`, `&`... Esses caracteres não devem ser utilizados de forma solta nos textos, porque mais cedo ou mais tarde pode desencadear um problema de interpretação no browser. Para isso, devemos usar os caracteres especiais, como por exemplo `&lt;`, `&gt;`, `&amp;`, `&quot;`, `&apos;`, etc.

# Anatomia de um Documento HTML

```
<!DOCTYPE html>
<html lang="en">
<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>Document</title>
</head>
<body>

</body>
</html>
```

# Anatomia de um Documento HTML

- `<DOCTYPE html>` - Serve para que alguns navegadores entendam que estamos trabalhando com HTML 5.
- `<html>` - Início da cadeia de elementos. Contém dois importantes elementos filhos: `<head>` e `<body>`.
- `<head>` - Contém configurações importantes para a página, mas não para o que usuário está vendo de fato (com algumas exceções, como por exemplo, o title).
  - `<title>` - Altera o título exibido na aba da página.
  - `<meta>` - Permite a inserção de vários metadados que servem para configurar a sua página HTML, como no exemplo do próximo slide.

# Anatomia de um Documento HTML

```
<head>  
  <!-- Essa configuração permite a maior parte de caracteres possíveis -->  
  <meta charset="utf-8">  
  <!-- Essa configuração permite que sua página seja visível tanto em dispositivos  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
</head>
```

**<body>** - É o corpo do documento. Consiste no conteúdo visual que será exibido para o usuário em tela.



# Títulos (Cabeçalhos) e Parágrafos

Se você pensar em uma revista ou um jornal, quanto melhor formatado e estruturado for o texto, melhor a experiência de leitura. O mesmo acontece com HTML.

Se você pegar um texto sem formatação (sem tags), dificilmente você vai conseguir distinguir as seções e o que há dentro de cada uma delas.

Além disso, existem os motores de busca (google, yahoo, bing, etc) que estão o tempo todo buscando páginas. Quanto mais semântica a sua página for, melhor ranqueada ela será.

Outro motivo pra um HTML bem estruturado é a acessibilidade. Extensões que leem páginas, por exemplo, terão dificuldade em ler um texto mal formatado.

# Títulos (Cabeçalhos) e Parágrafos

```
<body>
  Sobre mim
  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod ...

  Nascimento
  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod ...

  Trabalho
  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod ...

  Saiba mais
</body>
```

Utilizamos o `<h1>` para títulos das páginas. Geralmente, temos apenas um `h1` por página. Embora isso não seja uma regra, é uma boa prática.

No exemplo, colocaríamos o "Sobre mim" como um `h1` e o Nascimento e trabalho como `h2`.

# Listas

Imagine que tenhamos uma lista de ingredientes. Como podemos fazer para formartá-la como lista utilizando HTML?

Para criarmos uma lista, basta inserirmos os elementos dentro de uma tag `<li>`. Essas listas podem ser ordenadas ou não ordenadas, portanto, é importante sempre especificar qual modelo de lista estamos trabalhando. Podemos envolver as tags `<li>` em uma tag `<ol>` (Ordered List) para transformar a lista em uma lista ordenada. Para obtermos uma lista não ordenada, basta envolvermos as tags `<li>` em uma tag `<ul>` (Unordered List).

# Citações

Podemos inserir citações no HTML de três formas simples:

```
<blockquote cite="https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/blockquote">  
  O <strong>Elemento HTML <code>&lt;blockquote&gt;</code></strong> (ou <em>HTML Block  
  Quotation Element</em>) indica que um texto foi citado.  
</blockquote>
```

```
<p>De acordo com <a href="https://developer.mozilla.org/  
  pt-BR/docs/Web/HTML/Element/blockquote">  
  <cite>página MDN blockquote</cite></a>  
</p>
```

```
<p>O elemento quote – <code>&lt;q&gt;</code> – é <q cite="https://developer.  
  mozilla.org/pt-BR/docs/Web/HTML/Element/blockquote">usado para citações  
  curtas que não precisam de parágrafos ou quebras de linha.</q> --  
  <a href="https://developer.mozilla.org/pt-BR/docs/Web/HTML/Element/  
  q"><cite>MDN q Page</cite></a></p>
```

# Abreviações

```
<p>
  Usamos <abbr title="Hypertext Markup Language">HTML</abbr> para estruturar
  nossos documentos da web.
</p>
```

## Detalhes de Contato

Serve para inserção de detalhes de contato de quem criou a página HTML.

```
<address>
  <p>Yuri Weilemann <br><strong>Petrópolis, RJ</strong></p>
</address>
```

# Listas de Descrição

Listas de descrição são úteis para quando queremos construir glossários, por exemplo. Podemos utilizar a tag `<dl>` pra declarar uma lista de descrição, `<dt>` para um título de descrição e `<dd>` para a própria descrição.

```
<h2>Glossário</h2>
<dl>
  <dt>Hypertext</dt>
  <dd>É um hipertexto com possibilidades...</dd>

  <dt>Markup</dt>
  <dd>Marcação do texto</dd>

  <dt>Language</dt>
  <dd>Linguagem com sua semântica e sintaxe...</dd>
</dl>
```

# Representação de código

A tag `<code>` pode ser usado para inserir partes genéricas de código.

A tag `<pre>` pode ser usada para inserir blocos de código, pois essa tag mantém os espaços em branco e recuos que eu colocar no meu código.

```
<pre>  
  <code>if(isItTime == true) {  
    console.log("JOHN CENAAAAA!")  
  }</code>  
</pre>
```

Outra coisa importante é que nesses blocos de código não é possível a inclusão dos caracteres especiais de forma normal, então é necessário utilizarmos a sintaxe `&lt;` `&gt;` etc. Existe uma tag chamada `<xmp>` que está caindo em desuso mas que permite o uso desses caracteres normalmente, entretanto eu não aconselho a utilização pois ela não funcionará corretamente em alguns casos.

# Elementos Genéricos

Os elementos genéricos são elementos que não possuem uma característica própria bem definida, como por exemplo divs e spans.

As divs inserem uma quebra de linha no seu fechamento e o span não. Normalmente, elas são vinculadas a uma classe ou a um id para que possamos trabalhar com eles de forma separada utilizando o CSS ou o JavaScript.



# Links

Além dos atributos globais, tags de âncora podem receber atributos de href e target.

**Hrefs** - podem ser URLs completa, fragmentos (#fragmento), email (mailto:email@hotmail.com), telefone (tel:+552412345678), entre outros.

**Target** - \_self por padrão ou posso inserir o \_blank para abrir em uma nova aba.

# Links

- Fragmentos

```
<!-- Fragmentos -->
<ul>
  <li><a href="#about">Sobre mim</a></li>
  <li><a href="#history">História</a></li>
  <li><a href="#works">Trabalhos</a></li>

  <h1 id="about">Sobre mim</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit. Asperiores quis itaque maxi

  <h1 id="history">História</h1>
  <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Voluptatibus, cupiditate lib

  <h1 id="works">Trabalhos</h1>
  <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Dolores, voluptatibus natus
</ul>
```

- Email

```
<!-- Email -->
<a href="mailto:yuri_weilemann@hotmail.com">Enviar E-mail</a>
```

- Telefone

```
<!-- Telefone-->
<a href="tel:+55988147885">Ligar</a>
```

Podemos inserir qualquer tag dentro de uma anchor tag. Desta forma, estes elementos acionarão a função da anchor tag ao serem clicados.

# Tabelas

Podemos referenciar páginas e arquivos em diversos elementos HTML através do path ou da URL do objeto que queremos referenciar.

```
<body>
  <table>
    <caption>Pessoas por idade</caption>

    <thead>
      <tr>
        <th>Nome</th>
        <th>Idade</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Yuri</td>
        <td>27</td>
      </tr>
      <tr>
        <td>João</td>
        <td>22</td>
      </tr>
    </tbody>
  </table>
</body>
```

- table – Início da tabela
- caption – Título da tabela
- thead – Título das colunas
- tbody – Corpo da tabela
- tr – Linha da tabela (table row)
- td – Dados da tabela (table data)

# Meta SEO

Existem alguns metadados que podemos inserir no nosso código que auxiliam no ranqueamento dos motores de busca. Alguns exemplos são:

```
<meta name="author" content="Yuri Weilemann">
<!-- A descrição que aparece na pesquisa do google -->
<meta name="description" content="Esse site é bem bacana">
<!-- Não seguir os links da página -->
<meta name="robots" content="nofollow">
<!-- Seguir os links da página -->
<meta name="robots" content="follow">
<!-- Não indexar a página porém seguir os links da página -->
<meta name="robots" content="noindex, follow">
<!-- Indexar a página porém não seguir os links da página -->
<meta name="robots" content="index, nofollow">
```