

**Pós-Graduação Lato Sensu**  
Curso de Especialização em Tecnologia Java

# Frameworks Web

## HTML

**Prof. Esp. Hugo Baker Goveia**



# 1. O que é HTML?

É a linguagem que descreve a estrutura do conteúdo semântico de um documento da web.

O HTML não é uma linguagem de programação, ela é considerada uma linguagem de marcação(**HyperText Markup Language**).

O HTML é um padrão internacional mantido e especificado pela **W3C**(World Wide Web Consortium) e **WHATWG**(What Working Group), portanto são as entidades que definem e melhoram o HTML.

É importante frisar também que temos na internet a **W3SCHOOLS** e eles não são a entidade **W3C** citada acima, a **W3SCHOOLS** é simplesmente um site de referências e tutoriais.

## **W3C**

O W3C (World Wide Web Consortium) é uma organização internacional que desenvolve padrões abertos para a web, com o objetivo de garantir o crescimento a longo prazo da web e promover a sua interoperabilidade. Fundado em 1994 por Tim Berners-Lee, o W3C é composto por uma equipe dedicada de engenheiros, especialistas técnicos e membros da comunidade web de todo o mundo.

As atividades do W3C incluem o desenvolvimento e a publicação de especificações técnicas e recomendações que abrangem uma ampla gama de tecnologias web, como HTML, CSS, XML, JavaScript, entre outras. Essas especificações ajudam a garantir que os desenvolvedores de software possam criar conteúdo e aplicativos web que funcionem de forma consistente em diferentes navegadores e dispositivos.

Além disso, o W3C promove a educação e o entendimento público sobre os padrões da web, realiza conferências e eventos, e colabora com outras organizações e grupos de interesse para promover a inovação e a evolução contínua da web.

É uma entidade que desempenha um papel crucial na definição e na manutenção dos padrões da web, trabalhando para garantir que a web permaneça aberta, acessível e interoperável para todos.

## **WHATWG**

A WHATWG (Web Hypertext Application Technology Working Group) é um grupo de trabalho formado por um conjunto de empresas e indivíduos que se uniram para desenvolver e manter tecnologias relacionadas à web. Foi fundado em 2004 como uma resposta à falta de progresso na evolução das tecnologias web, particularmente no que diz respeito ao HTML e ao DOM (Document Object Model).

A WHATWG concentra-se principalmente na evolução e no desenvolvimento de padrões abertos para a web, com o objetivo de impulsionar a inovação e melhorar a experiência do usuário na web. Seu trabalho é realizado de forma aberta e colaborativa, com contribuições de membros da comunidade web em todo o mundo.

Uma das contribuições mais significativas da WHATWG foi a especificação do HTML5, uma revisão importante da linguagem HTML que introduziu uma série de novos recursos e melhorias para a web, incluindo suporte a vídeo e áudio nativos, APIs para armazenamento offline, canvases para gráficos interativos, entre outros.

A WHATWG continua a desempenhar um papel importante no desenvolvimento da web, trabalhando em estreita colaboração com o W3C (World Wide Web Consortium) para manter e evoluir os padrões da web de forma aberta e interoperável.

## HTML5

É a quinta versão da linguagem de marcação HTML. Lançada em 2014, trouxe uma série de novos recursos, melhorias e funcionalidades em comparação com suas versões anteriores, tornando-se uma base sólida para o desenvolvimento de aplicativos web modernos e dinâmicos.

Temos muito mais semântica após essa versão.

*Alguns dos recursos e melhorias mais significativos introduzidos pelo HTML5 são:*

- **Novas Tags Semânticas:** tags como <header>, <footer>, <nav>, <article>, <section>, que tornam mais fácil e intuitivo estruturar e organizar o conteúdo de uma página web de forma significativa para os motores de busca e para os desenvolvedores.
- **Suporte a Vídeo e Áudio Nativos:** suporte nativo para a reprodução de vídeo e áudio em navegadores, sem a necessidade de plugins adicionais como o Flash. As tags <video> e <audio> permitem a incorporação de conteúdo multimídia diretamente em páginas web, oferecendo uma experiência mais rica e interativa para os usuários.
- **Canvas e Gráficos 2D/3D:** introduziu a tag <canvas>, que permite a renderização de gráficos 2D e 3D diretamente no navegador usando JavaScript. Isso possibilita a criação de jogos, visualizações de dados e animações complexas diretamente na web, sem a necessidade de plugins externos.
- **Armazenamento Local:** oferece suporte para armazenamento local de dados do cliente, permitindo que aplicativos web armazenem e acessem dados no navegador do usuário. Isso inclui recursos como o Web Storage (localStorage e sessionStorage) e o IndexedDB, que oferecem uma alternativa ao uso de cookies para armazenar dados do lado do cliente.
- **Geolocalização:** introduziu APIs para obter a localização geográfica do usuário através do navegador, permitindo que aplicativos web forneçam serviços baseados em localização, como mapas interativos, serviços de geolocalização e muito mais.

## 2. O que é DOM?

O DOM (Document Object Model) é uma representação estruturada em forma de árvore de um documento HTML ou XML, que define a estrutura e o conteúdo de uma página web ou de um documento XML como um conjunto de objetos interconectados.

Quando um navegador carrega uma página web, ele interpreta o código HTML da página e cria uma representação em memória da estrutura da página usando o DOM. Essa representação em memória permite que o navegador manipule dinamicamente a estrutura, o conteúdo e o estilo da página por meio de scripts, como JavaScript.

O DOM organiza os elementos HTML da página em um conjunto de nós (nodes), que formam uma hierarquia de pais e filhos, representando a estrutura aninhada dos elementos HTML. Cada elemento HTML é representado por um nó no DOM, e os atributos, texto e outros conteúdos associados a cada elemento são representados por propriedades desses nós.

Por exemplo, considere o seguinte trecho de código HTML:

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemplo de DOM</title>
</head>
<body>
  <h1>Olá, mundo!</h1>
  <p>Este é um exemplo simples de DOM.</p>
</body>
</html>
```

No DOM correspondente a essa página, teríamos uma estrutura de nós semelhante a esta:

```
Document
├── html
│   ├── head
│   │   └── title
│   │       └── "Exemplo de DOM"
│   └── body
│       ├── h1
│       │   └── "Olá, mundo!"
│       └── p
│           └── "Este é um exemplo simples de DOM."
```

O DOM é uma parte fundamental da plataforma web e é amplamente utilizado por desenvolvedores para manipular dinamicamente o conteúdo, a estrutura e o estilo de páginas web por meio de scripts do lado do cliente, como JavaScript. Ele permite a interação e a atualização de páginas web sem a necessidade de recarregar a página inteira, tornando a web mais dinâmica e interativa.

### 3. Tags

Basicamente são **palavras-chave** cercadas pelos sinais de **menor e maior** "<>". Elas normalmente são usadas em **pares** como por exemplo `<h1>` e `</h1>`.

Onde podemos observar que há uma Tag de abertura: `<comando>`

E uma Tag de fechamento: `</comando>`

Existem também Tags chamadas de **singulares** ou **auto-fechadas**: `<br />` ou `<br>`

Portanto uma tag é um elemento fundamental da linguagem de marcação HTML que é usada para definir e estruturar o conteúdo de uma página web. As tags HTML são usadas para marcar diferentes partes do conteúdo, como texto, imagens, links, formulários e muito mais, permitindo que os navegadores entendam como exibir e interagir com o conteúdo da página.

Cada tag HTML é composta por um nome de tag delimitado por colchetes angulares < >. Por exemplo, a tag `<p>` é usada para definir um parágrafo de texto, a tag `<img>` é usada para inserir uma imagem e a tag `<a>` é usada para criar um link.

Então é importante ressaltar o que foi dito acima, que as tags HTML geralmente têm uma estrutura de abertura e fechamento. A estrutura de abertura consiste no nome da tag envolto por colchetes angulares, e a estrutura de fechamento consiste no nome da tag precedido por uma barra / dentro de colchetes angulares.

*Exemplo de uso de tags HTML em um documento HTML:*

```

1  <!DOCTYPE html>
2  <html lang="pt-BR">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Exemplo de Tags HTML</title>
7  </head>
8  <body>
9      <h1>Este é um cabeçalho de nível 1</h1>
10     <p>Este é um parágrafo de texto.</p>
11     
12     <a href="https://www.utfpr.edu.br/campus/cornelioprocopio">Este é um link para o site da UTFPR, campus de Cornélio Procopio</a>
13 </body>
14 </html>
15

```

Neste exemplo acima, as tags HTML como `<h1>`, `<p>`, `<img>` e `<a>` são usadas para definir e estruturar o conteúdo da página. Cada uma dessas tags tem um propósito específico e ajuda a criar uma página web funcional e interativa.

*Abaixo podemos observar o resultado das Tags do exemplo acima em um browser:*

## Este é um cabeçalho de nível 1

Este é um parágrafo de texto.



[Este é um link para o site da UTFPR, campus de Cornélio Procópio](#)

### ATRIBUTOS DE UMA TAG

Os atributos de uma tag HTML fornecem informações adicionais sobre como a tag deve ser processada ou exibida pelo navegador. Eles são usados para modificar o comportamento ou a aparência de uma tag e geralmente são especificados dentro do par de colchetes da tag, após o nome da tag.

*Características dos atributos das tags HTML:*

- **Nome do Atributo:** Cada atributo tem um nome específico que indica sua função. Por exemplo, em `<a href="https://www.example.com">`, o nome do atributo é **"href"**.
- **Valor do Atributo:** Os atributos podem ter um valor associado que define sua propriedade. No exemplo acima, `"https://www.example.com"` é o **valor** do atributo **"href"**.
- **Sintaxe:** A maioria dos atributos é especificada usando a sintaxe **nome="valor"**, onde o nome do atributo é seguido por um sinal de igual `"=`" e o valor do atributo é colocado entre aspas duplas `"`.
- **Atributos Obrigatórios vs. Opcionais:** Alguns atributos são obrigatórios para determinadas tags, enquanto outros são opcionais. Por exemplo, a tag `<img>` requer o atributo **"src"** para especificar o caminho da imagem, enquanto o atributo **"alt"** é opcional e fornece texto alternativo para a imagem.
- **Atributos Globais:** Existem também os atributos globais, que podem ser usados em qualquer tag HTML. Alguns exemplos de atributos globais incluem **"id"**, **"class"**, **"style"** e **"title"**. Esses atributos podem ser usados em qualquer tag para fornecer identificação, estilo ou informações adicionais.

Exemplo focado no uso de atributos em tags HTML:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplo de Atributos em Tags HTML</title>
</head>
<body>
  <a href="https://www.utfpr.edu.br/campus/cornelioprocopio" title="Visitar site da UTFPR">
    Este é um link para o site da UTFPR, campus de Cornélio Procópio
  </a>
  
</body>
</html>
```

No exemplo acima, a tag `<a>` possui os atributos `"href"` e `"title"`, enquanto a tag `<img>` possui os atributos `"src"`, `"alt"`, `"width"` e `"height"`. Cada atributo desempenha um papel específico na definição do comportamento ou da aparência da tag correspondente.

## 4. DOCTYPE e Codificação

### DOCTYPE

Todos os documentos HTMLs precisam ter uma declaração que indica para o navegador qual é a versão do HTML que a gente está usando.

Essa declaração fica na primeira linha do documento HTML e usamos o `<!DOCTYPE html>` para essa definição.

Até a versão 4 do HTML, era necessário informar mais dados nessa instrução, mas a partir da versão 5 do HTML não é mais necessário, somente informar `<!DOCTYPE html>`.

Importante sabermos que isso não é uma TAG HTML, é apenas uma instrução para o browser indicando qual é o tipo do documento.

Ela indica que o documento HTML está escrito de acordo com as especificações do HTML5, a versão mais recente e amplamente adotada da linguagem de marcação HTML. Ela deve ser colocada no topo do documento HTML, antes de qualquer outra tag, e não requer um elemento de fechamento.

Esta instrução é importante porque ajuda os navegadores a interpretar o código HTML corretamente, garantindo que eles usem os padrões mais recentes e mantenham a compatibilidade com versões anteriores da linguagem. Além disso, ao utilizar o `<!DOCTYPE html>`, os navegadores sabem que devem tratar o documento como um documento **HTML5**, o que é essencial para a renderização correta de novos recursos e funcionalidades introduzidos nesta versão.

## CODIFICAÇÃO

Dentro do head (cabeçalho do documento) nós temos que especificar através do **META** a informação do encoding. Para isso usamos o **<meta charset='utf-8'>**

O elemento meta não precisa ser fechado usando o **</meta>** e basta apenas fechar a tag meta usando o **>**

Essa codificação é usada para evitar distorção nas acentuações do texto de nosso site e se refere ao conjunto de regras que especifica como os caracteres no documento HTML serão interpretados e exibidos pelos navegadores. Em outras palavras, define como os caracteres Unicode (letras, números, símbolos) serão convertidos em bytes e interpretados pelo navegador para renderizar o conteúdo da página web corretamente.

Existem várias codificações de caracteres disponíveis, como UTF-8, UTF-16, ISO-8859-1 (Latin-1), entre outras. A codificação **UTF-8** é a mais amplamente utilizada e recomendada para documentos HTML, pois suporta uma ampla gama de caracteres e é capaz de representar qualquer caractere Unicode.

Ao definir a codificação do documento HTML, você está informando ao navegador como os caracteres no documento devem ser interpretados. Isso é feito utilizando a meta tag **<meta charset="nome\_da\_codificação">** no cabeçalho do documento HTML conforme exemplificado acima.

## 5. Head

O elemento **<head>** é uma parte fundamental de um documento HTML que contém metadados e informações adicionais sobre o documento, mas não é exibido na página web renderizada pelo navegador. Em vez disso, o conteúdo dentro do elemento **<head>** fornece informações sobre o documento HTML e suas características, como título da página, codificação de caracteres, estilos CSS, scripts JavaScript e outras informações importantes para a renderização e funcionamento correto da página.

*Principais elementos que podem estar presentes dentro do elemento <head>:*

- **<title>**: Este elemento define o título da página, que é exibido na barra de título do navegador ou na aba da janela do navegador. O texto dentro do elemento **<title>** é o que os usuários veem quando a página é marcada como favorita ou compartilhada nas redes sociais.
- **<meta>**: Elementos **<meta>** são usados para fornecer metadados adicionais sobre o documento HTML. Isso pode incluir metadados como a descrição da página, palavras-chave para mecanismos de busca, codificação de caracteres, autor e outros. Um exemplo comum é a meta tag **<meta charset="UTF-8">** mencionada anteriormente, que define a codificação de caracteres do documento como UTF-8.
- **<link>**: Este elemento é usado para vincular o documento HTML a recursos externos, como arquivos de estilo CSS, fontes externas, ícones de favicons e muito mais. Por exemplo, **<link rel="stylesheet" href="estilos.css">** é usado para vincular um arquivo CSS chamado "estilos.css" ao documento HTML.



- **<script>**: Elementos `<script>` são usados para incluir scripts JavaScript dentro do documento HTML. Isso pode incluir scripts inline diretamente no documento ou links para arquivos de script externos. Por exemplo, `<script src="script.js"></script>` é usado para vincular um arquivo JavaScript chamado "script.js" ao documento HTML.

Além desses elementos mencionados acima, o `<head>` também pode conter outras informações adicionais, como metadados específicos de redes sociais, instruções de rastreamento para mecanismos de busca e muito mais. Portanto, o elemento `<head>` é onde todas as informações e configurações importantes sobre o documento HTML são definidas, ajudando a garantir uma experiência de usuário adequada e otimizada.

## VIEWPORT

A meta tag `<meta name="viewport" content="width=device-width, initial-scale=1.0">` é uma instrução utilizada em documentos HTML para controlar a forma como o navegador de dispositivos móveis (como em smartphones e tablets) renderizam o conteúdo da página, definem as dimensões iniciais e a escala de visualização.

O atributo `name` com valor **viewport** especifica o tipo de metadado que estamos definindo. Neste caso, estamos informando ao navegador que se trata de configurações de **visualização da página**. O atributo `content="width=device-width, initial-scale=1.0"` está especificando o conteúdo do metadado. O `width=device-width` define a largura inicial da janela de visualização para ser igual à largura do dispositivo (a largura da tela do dispositivo em pixels). Isso garante que a página se ajuste automaticamente à largura da tela do dispositivo, impedindo que a página seja redimensionada ou ampliada desnecessariamente.

E o `initial-scale=1.0` define o nível inicial de escala de visualização da página. Um valor de 1.0 indica que a página deve ser exibida com uma escala de 100%, ou seja, sem zoom. Isso garante que a página seja exibida em tamanho real quando carregada inicialmente.

Juntas, essas configurações garantem uma experiência de visualização consistente e otimizada em dispositivos móveis, permitindo que os usuários naveguem facilmente pelo conteúdo da página sem a necessidade de redimensionamento manual ou zoom. Essa meta tag é particularmente importante para o desenvolvimento de páginas web responsivas, que se adaptam dinamicamente a diferentes tamanhos de tela e dispositivos.

## 6. Tag <H1>

O elemento `<h1>` é uma tag HTML usada para definir o cabeçalho de maior importância em uma página web. Ele representa o título principal ou o cabeçalho de nível mais alto em uma hierarquia de títulos. Geralmente o título `<h1>` só aparece uma vez em uma página HTML.

### Hierarquia de Títulos

Em HTML, os títulos são organizados em uma hierarquia, onde o `<h1>` representa o título principal ou o cabeçalho de nível mais alto, seguido pelo `<h2>` para subseções importantes, `<h3>` para subseções dentro de `<h2>`, e assim por diante até `<h6>`, que é usado para títulos de menor importância.

## Importância Semântica

O elemento <h1> não apenas fornece um estilo de texto maior e mais destacado, mas também possui uma importância semântica especial. Os motores de busca e os leitores de tela usam a hierarquia de títulos para entender a estrutura e o contexto do conteúdo da página. Portanto, é importante usar o elemento <h1> de forma apropriada, reservando-o para o título principal da página ou para a seção mais importante.

## Apresentação Visual

Por padrão, os navegadores aplicam um estilo de texto maior e mais negrito ao conteúdo dentro do elemento <h1>, tornando-o visualmente destacado em relação ao restante do conteúdo da página. No entanto, o estilo exato pode variar dependendo do CSS aplicado ao documento.

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemplo de Título Principal</title>
</head>
<body>
  <h1>Este é o Título Principal da Página</h1>
  <p>Este é um parágrafo de texto que segue o título principal.</p>
  <br>
  <h2>Este é o Título Secundário da Página</h2>
  <p>Este é um parágrafo de texto que segue o título secundário.</p>
  <br>
  <h3>Este é o Título h3</h3>
  <p>Este é um parágrafo de texto que segue o título h3.</p>
  <br>
  <h4>Este é o Título h4</h4>
  <p>Este é um parágrafo de texto que segue o título h4.</p>
  <br>
  <h5>Este é o Título h5</h5>
  <p>Este é um parágrafo de texto que segue o título h5.</p>
  <br>
  <h6>Este é o Título h6</h6>
  <p>Este é um parágrafo de texto que segue o título h6.</p>
</body>
</html>
```

# Este é o Título Principal da Página

Este é um parágrafo de texto que segue o título principal.

## Este é o Título Secundário da Página

Este é um parágrafo de texto que segue o título secundário.

### Este é o Título h3

Este é um parágrafo de texto que segue o título h3.

#### Este é o Título h4

Este é um parágrafo de texto que segue o título h4.

##### Este é o Título h5

Este é um parágrafo de texto que segue o título h5.

###### Este é o Título h6

Este é um parágrafo de texto que segue o título h6.

No exemplo acima, `<h1>Este é o Título Principal da Página</h1>` representa o título principal da página, enquanto `<p>Este é um parágrafo de texto que segue o título principal.</p>` é um parágrafo de texto que segue o título principal. O elemento `<h1>` é usado para definir o título principal da página, indicando sua importância semântica e visual. Logo após são apresentados os outros níveis de título para se usar em um documento HTML.

## 7. Tag <p>

O elemento <p> é uma tag HTML usada para definir um parágrafo de texto em uma página web. Ele é usado para agrupar blocos de texto em um único parágrafo, criando assim uma estrutura visual e semântica que facilita a leitura e a compreensão do conteúdo.

### Agrupamento de Texto

O elemento <p> é usado para agrupar texto em blocos coesos, representando uma unidade lógica de conteúdo. Cada <p> contém um único parágrafo de texto.

### Formatação Padrão

Por padrão, os navegadores aplicam um estilo de texto específico aos elementos <p>, geralmente incluindo um espaçamento vertical entre os parágrafos para melhorar a legibilidade do conteúdo.

### Importância Semântica

Embora o elemento <p> seja comumente usado para representar parágrafos de texto simples, ele também possui uma importância semântica significativa. Os motores de busca e outros agentes de processamento de conteúdo usam os elementos <p> para entender a estrutura e o significado do conteúdo da página.

### Conteúdo Permitido

O elemento <p> pode conter uma variedade de elementos HTML, como texto simples, imagens, links e até mesmo outras tags <p> aninhadas. No entanto, é importante manter o conteúdo dentro de um <p> coeso e relevante para garantir a clareza e a legibilidade do texto.

## 8. Comentário no código HTML

Para adicionar comentário no código HTML, basta usar antes do comentário o "<!--" e ao finalizar "-->" conforme exemplo baixo:

```
<!--Inicio do artigo -->
```

## 9. Tag <br>

Usamos a TAG <br> (**break**) para efetuar a quebra de linha, porém não podemos ficar usando de forma descontrolada para aumentar o espaço entre um parágrafo e outro. NÃO É UMA BOA PRÁTICA, uma boa prática é usar o CSS para isso.

## 10. Tag <img>

A tag <img> em HTML é usada para inserir uma imagem em uma página web. Ela é uma tag vazia, o que significa que não possui uma tag de fechamento correspondente e pode ser escrita como uma única tag.

*Atributos importantes da tag <img>:*

- **Src (Source):** O atributo src é obrigatório e especifica o caminho para o arquivo de imagem que deve ser exibido. Isso pode ser uma URL de uma imagem hospedada na web ou um caminho para uma imagem em seu servidor local.
- **Alt (Alternative Text):** O atributo alt é usado para fornecer um texto alternativo para a imagem. Este texto é exibido quando a imagem não pode ser carregada pelo navegador ou quando um leitor de tela é usado por usuários com deficiência visual. O texto alternativo também é importante para mecanismos de busca e acessibilidade web.
- **Width (Largura) e Height (Altura):** Os atributos width e height podem ser usados opcionalmente para especificar as dimensões da imagem em pixels. Isso pode ser útil para definir o tamanho da imagem na página, controlando sua aparência e layout.
- **Title (Título):** O atributo title é opcional e pode ser usado para fornecer um título ou uma descrição adicional da imagem. Quando o usuário passa o mouse sobre a imagem, o título é exibido como uma dica de ferramenta (tooltip), fornecendo mais informações sobre a imagem.

## 11. Tag <a>

A tag <a> em HTML é usada para criar links (ou âncoras) entre páginas da web, permitindo que os usuários naveguem de uma página para outra ou a diferentes seções da mesma página. Ela é uma das tags mais importantes para a navegação e interação em páginas web.

*Atributos e características importantes da tag <a>:*

- **Href (Hypertext Reference):** O atributo href é obrigatório e especifica o destino do link. Isso pode ser uma URL para outra página web, o endereço de e-mail de um destinatário ou um identificador dentro do próprio documento HTML (usado para links internos).
- **Texto Ancorado:** O texto entre as tags <a> e </a> é conhecido como texto ancorado e é o que os usuários veem na página como o link clicável. Esse texto fornece contexto sobre o destino do link e é clicável para iniciar a ação de navegação.
- **Target (Alvo):** O atributo target é opcional e especifica onde o destino do link será aberto. Os valores comuns para este atributo incluem \_blank para abrir o link em uma nova janela ou guia, \_self para abrir o link na mesma janela (o comportamento padrão) e outros valores como \_parent e \_top para controlar o contexto de abertura do link em frames ou iframes.
- **Title (Título):** O atributo title também é opcional e fornece um título ou descrição adicional do link. Quando o usuário passa o mouse sobre o link, o título é exibido como uma dica de ferramenta (tooltip), fornecendo informações adicionais sobre o destino do link.

## 12. Tag <div>

A tag <div> em HTML é um elemento de bloco usado para criar uma divisão ou seção em uma página web. É uma das tags mais versáteis e amplamente utilizadas em HTML e CSS, permitindo

aos desenvolvedores agrupar e organizar o conteúdo da página de forma semântica e estruturada.

### **Divisão de Conteúdo**

A tag <div> é usada para dividir o conteúdo de uma página web em seções lógicas e independentes. Cada <div> representa uma área separada da página que pode conter qualquer tipo de conteúdo, como texto, imagens, formulários, listas, outros elementos HTML, etc.

### **Elemento de Bloco**

O <div> é um elemento de bloco, o que significa que ele ocupa todo o espaço horizontal disponível na página, é renderizado em uma nova linha antes e depois do conteúdo que ele contém. Isso permite que os desenvolvedores controlem o layout e a aparência de seções individuais da página.

### **Estilo e Layout**

A tag <div> é frequentemente usada em conjunto com CSS para aplicar estilos de formatação e layout ao conteúdo da página. Os desenvolvedores podem atribuir classes ou IDs às divs para aplicar estilos específicos usando regras de CSS correspondentes.

### **Agrupamento de Elementos**

As divs são frequentemente usadas para agrupar vários elementos relacionados juntos, criando assim uma unidade lógica de conteúdo. Por exemplo, uma div pode ser usada para envolver um cabeçalho, um menu de navegação e um conteúdo principal, agrupando-os visualmente e estruturalmente.

```

<!DOCTYPE html>
<html>
<head>
  <title>Exemplo de Tag DIV</title>
  <style>
    .conteudo {
      background-color: #f0f0f0;
      padding: 20px;
      border: 1px solid #ccc;
      border-radius: 5px;
    }
    .conteudo2 {
      background-color: #836FFF;
      padding: 20px;
      border: 5px solid #00008B;
      border-radius: 5px;
    }
  </style>
</head>
<body>
  <div class="conteudo">
    <h1>Divisão de Conteúdo - classe CSS conteudo</h1>
    <p>Esta é uma divisão de conteúdo em uma página HTML.</p>
  </div>
  <br>
  <div class="conteudo2">
    <h1>Divisão de Conteúdo - classe CSS conteudo2</h1>
    <p>Esta é uma divisão de conteúdo em uma página HTML.</p>
  </div>
</body>
</html>

```

No exemplo acima podemos ver o <div> usando a classe no atributo “class”. Observe que cada uma das divs usa uma classe diferente e isso impacta na formatação delas.

Observe na imagem abaixo o resultado da implementação do código mostrado acima.

**Divisão de Conteúdo - classe CSS conteudo**

Esta é uma divisão de conteúdo em uma página HTML.

**Divisão de Conteúdo - classe CSS conteudo2**

Esta é uma divisão de conteúdo em uma página HTML.

## 13. Tag <header>

A tag <header> em HTML é um elemento de bloco usado para representar o cabeçalho de uma seção ou de toda a página em uma estrutura HTML. Ele é projetado para conter elementos introdutórios ou de navegação que são comuns em muitas páginas da web, como logotipos, menus de navegação, títulos principais e outros elementos relacionados ao topo da página.

### Cabeçalho da Página

O elemento <header> é geralmente usado para representar o cabeçalho de uma página, embora também possa ser usado dentro de outras seções da página para representar cabeçalhos de seção.

### Elemento de Bloco

Assim como a tag <div>, <header> é um elemento de bloco, o que significa que ele ocupa todo o espaço horizontal disponível e é renderizado em uma nova linha antes e depois do conteúdo que ele contém.

### Conteúdo Introdutório

A tag <header> é usada para agrupar e organizar elementos introdutórios ou de navegação que são relevantes para o início da página ou de uma seção específica. Isso pode incluir elementos como logotipos, títulos principais, menus de navegação, links de retorno à página inicial, etc.

### Semântica

O uso da tag <header> ajuda a fornecer uma estrutura semântica para o conteúdo da página, permitindo que os navegadores, mecanismos de busca e leitores de tela entendam a importância e a função do conteúdo contido dentro do cabeçalho.

Sendo assim usamos no lugar do div que identificaria o cabeçalho de nossa página, com isso podemos implementar uma melhor semântica em nossa página HTML. Portanto é entendido pelos desenvolvedores e navegadores como o cabeçalho da página.



```

<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplo de uso da Tag HEADER</title>
  <style>
    header {
      background-color: #333;
      color: #fff;
      padding: 20px;
      text-align: center;
    }
  </style>
</head>
<body>
  <header>
    <h1>Título do site</h1>
    <nav>
      <ul>
        <li><a href="#">Página Inicial</a></li>
        <li><a href="#">Sobre</a></li>
        <li><a href="#">Contato</a></li>
      </ul>
    </nav>
  </header>
  <main>
    <h2>Conteúdo Principal</h2>
    <p>Este é o conteúdo principal da página.</p>
  </main>
</body>
</html>

```

No exemplo de código acima podemos verificar o uso da tag <header> proporcionando uma boa semântica ao código da página HTML.

Abaixo podemos observar o resultado do código acima:



### Conteúdo Principal

Este é o conteúdo principal da página.

## 14. Tag <footer>

A tag <footer> em HTML é usada para representar o rodapé de uma página ou de uma seção específica dentro de uma página web. Ele geralmente contém informações de rodapé, como créditos, links para páginas relacionadas, informações de contato, links de navegação secundários, entre outros elementos que são relevantes para o final do conteúdo da página.

### **Conteúdo de Rodapé**

O conteúdo dentro da tag <footer> é destinado a informações que são consideradas o "final" ou a conclusão do conteúdo da página. Isso pode incluir informações como copyright, créditos, links de navegação secundários, informações de contato, links para páginas relacionadas, etc.

### **Localização na Página**

Tradicionalmente, o <footer> é colocado no final da página, após todo o conteúdo principal. No entanto, ele também pode ser usado dentro de seções específicas da página para representar o rodapé dessa seção em particular.

### **Semântica Clara**

O uso da tag <footer> fornece uma semântica clara para o conteúdo que representa, indicando aos navegadores e mecanismos de busca que este é o rodapé da página ou da seção. Isso ajuda na interpretação e na indexação do conteúdo pelos mecanismos de busca.

### **Acessibilidade**

O conteúdo dentro de <footer> pode ser acessado facilmente por usuários finais e por tecnologias assistivas, como leitores de tela, para fornecer informações adicionais e facilitar a navegação.

```

<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplo de página usando a Tag FOOTER</title>
  <style>
    footer {
      background-color: #333;
      color: #fff;
      padding: 20px;
      text-align: center;
    }
  </style>
</head>
<body>
  <header>
    <h1>Meu Site</h1>
    <nav>
      <ul>
        <li><a href="#">Página Inicial</a></li>
        <li><a href="#">Sobre</a></li>
        <li><a href="#">Contato</a></li>
      </ul>
    </nav>
  </header>
  <main>
    <p>Conteúdo principal da página...</p>
  </main>
  <footer>
    <p>© 2024 Meu Site de aprendizado na UTFPR - Todos os direitos reservados</p>
  </footer>
</body>
</html>

```

No exemplo acima a tag <footer> envolve o conteúdo de rodapé da página, que inclui informações de copyright.

O CSS é usado para aplicar estilos de formatação ao rodapé, definindo cor de fundo, cor do texto, preenchimento e alinhamento do texto. Isso demonstra como os estilos CSS podem ser aplicados para personalizar a aparência do rodapé da página. Na imagem abaixo podemos observar o resultado do uso da tag <footer>

## Meu Site

- [Página Inicial](#)
- [Sobre](#)
- [Contato](#)

Conteúdo principal da página...

## 15. Tag <ul>

A tag <ul> (Unordered List) em HTML é usada para criar uma lista não ordenada de itens. Uma lista não ordenada é uma lista de itens onde a ordem dos itens não é importante. Cada item na lista é representado por meio da tag <li> (elemento de lista ou List Item), que é filho direto da tag <ul>.

### Lista Não Ordenada

A <ul> cria uma lista na qual a ordem dos itens não importa. Isso significa que os itens podem ser exibidos em qualquer ordem e não têm uma numeração associada a eles.

### Elementos de Lista <li>

Cada item na lista é representado por uma tag <li>, que é um elemento de lista filho direto da tag <ul>. Os elementos <li> são renderizados como marcadores padrão (geralmente marcadores redondos) à esquerda de cada item na lista.

### Estrutura Hierárquica

A tag <ul> permite criar listas aninhadas, onde uma lista pode conter outra lista como um de seus itens. Isso permite a criação de estruturas de lista hierárquicas ou multiníveis.

### Semântica

O uso da tag <ul> ajuda a fornecer uma estrutura semântica para o conteúdo da página, indicando claramente que o conteúdo dentro dela representa uma lista não ordenada de itens.

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplo página com a Tag UL</title>
</head>
<body>
  <h2>Lista de Disciplinas</h2>
  <ul>
    <li>Sistemas de Banco de Dados</li>
    <li>Frameworks Web</li>
    <li>Teste de Software</li>
  </ul>
</body>
</html>
```

No exemplo acima podemos observar o uso da lista não ordenada para relacionar as disciplinas com a tag <ul> e os itens da lista com <li>.

Na imagem abaixo é possível verificar como ficaria.

## Lista de Disciplinas

- Sistemas de Banco de Dados
- Frameworks Web
- Teste de Software

## 16. Tag <ol>

A tag <ol> (**O**rdere**D** List) em HTML é usada para criar uma lista ordenada de itens. Ao contrário de uma lista não ordenada (<ul>), onde a ordem dos itens não importa, uma lista ordenada (<ol>) exibe os itens em uma ordem específica, geralmente numerando-os automaticamente. Cada item na lista é representado por meio da tag <li> (elemento de lista), que é um filho direto da tag <ol>.

### Lista Ordenada

A <ol> cria uma lista na qual a ordem dos itens é importante. Isso significa que os itens são exibidos em uma sequência específica e geralmente são numerados automaticamente, embora isso possa ser personalizado com CSS.

### Elementos de Lista <li>

Cada item na lista é representado por uma tag <li>, que é um elemento de lista filho direto da tag <ol>. Os elementos <li> são renderizados com números sequenciais (por padrão) à esquerda de cada item na lista.

### Estrutura Hierárquica

Assim como <ul>, a tag <ol> permite a criação de listas aninhadas, onde uma lista pode conter outra lista como um de seus itens. Isso permite a criação de estruturas de lista hierárquicas ou multiníveis.

### Semântica

O uso da tag <ol> ajuda a fornecer uma estrutura semântica para o conteúdo da página, indicando claramente que o conteúdo dentro dela representa uma lista ordenada de itens.

```

<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplo de página usando a Tag OL</title>
</head>
<body>
  <h2>Etapas para desenvolver uma task:</h2>
  <ol>
    <li>Puxar o card no Jira para a coluna "Em andamento"</li>
    <li>Abrir projeto que vai ser feita atuação</li>
    <li>Entrar na Branch "develop"</li>
    <li>Criar nova Branch com o número do card do "jira" + "titulo da task"</li>
    <li>Realizar implementações da feature + testes unitários</li>
    <li>Fazer commit e o push</li>
    <li>Abrir a PR da sua branch para a branch develop</li>
    <li>Mover o card da task no jira para a coluna "Em code-review"</li>
    <li>Aguardar aprovação (code review) dos colegas do time</li>
    <li>Após aprovação, realizar o Merge e rodar a esteira do Jenkins</li>
    <li>Mover o card para coluna "Para testar" e avisar QA do time</li>
  </ol>
</body>
</html>

```

Acima temos um exemplo de implementação usando uma lista ordenada através da tag <ol> e abaixo temos uma imagem do resultado exibido na página web.

## Etapas para desenvolver uma task:

1. Puxar o card no Jira para a coluna "Em andamento"
2. Abrir projeto que vai ser feita atuação
3. Entrar na Branch "develop"
4. Criar nova Branch com o número do card do "jira" + "titulo da task"
5. Realizar implementações da feature + testes unitários
6. Fazer commit e o push
7. Abrir a PR da sua branch para a branch develop
8. Mover o card da task no jira para a coluna "Em code-review"
9. Aguardar aprovação (code review) dos colegas do time
10. Após aprovação, realizar o Merge e rodar a esteira do Jenkins
11. Mover o card para coluna "Para testar" e avisar QA do time

## 17. Tag <form>

A tag <form> em HTML é usada para criar um formulário em uma página da web. Os formulários são usados para coletar informações dos usuários, como nome, endereço, senha, caixas de

seleção, botões de opção, etc. O conteúdo dentro da tag <form> define os campos e os controles que compõem o formulário.

### Coleta de Dados

A tag <form> é usada para definir a estrutura do formulário que será usado para coletar dados dos usuários. Os dados inseridos pelos usuários nos campos do formulário serão enviados para um servidor web para processamento posterior.

### Elementos de Formulário

O conteúdo dentro da tag <form> consiste em diferentes tipos de elementos de formulário, como caixas de texto <input type="text">, caixas de seleção <select>, botões de seleção <input type="checkbox">, botões de envio <input type="submit">, entre outros.

### Atributos do Formulário

A tag <form> pode ter atributos adicionais que especificam o método de envio dos dados (method), o URL para onde os dados serão enviados (action), e outras propriedades relacionadas ao comportamento e à aparência do formulário.

### Validação de Dados

A tag <form> pode ser usada em conjunto com JavaScript para realizar validação de dados no lado do cliente antes do envio do formulário para o servidor. Isso ajuda a garantir que os dados inseridos pelos usuários estejam corretos e em um formato adequado antes de serem processados.

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Formulário de Contato</title>
</head>
<body>
  <h2>Entre em Contato</h2>
  <form action="/enviar-contato" method="post">
    <label for="nome">Nome:</label>
    <input type="text" id="nome" name="nome" required>
    <br>
    <label for="email">E-mail:</label>
    <input type="email" id="email" name="email" required>
    <br>
    <label for="mensagem">Mensagem:</label>
    <textarea id="mensagem" name="mensagem" rows="4" required></textarea>
    <br>
    <input type="submit" value="Enviar Mensagem">
  </form>
</body>
</html>
```

Para entender o exemplo da implementação acima devemos observar que:

- A tag **<form>** envolve os campos e controles que compõem o formulário de contato.
- O atributo **action** especifica o URL para onde os dados do formulário serão enviados quando o usuário clicar no botão de envio.
- O atributo **method** especifica o método HTTP que será usado para enviar os dados do formulário para o servidor (geralmente "GET" ou "POST").
- Os campos de entrada incluem um campo de texto para o nome, um campo de email para o endereço de e-mail e uma área de texto para a mensagem.
- O botão de envio **<input type="submit">** é usado para enviar o formulário quando o usuário clicar nele.

### IMPORTANTE

**GET=>** ele envia os dados pela URL

Não é muito bom para usar em cadastros pois expõe os dados, pois consigo acessar rotas do tipo GET pela URL

- Os dados do formulário são anexados à URL como uma string de consulta (query string).
- Os dados são visíveis na barra de endereços do navegador.
- É adequado para envio de pequenas quantidades de dados.
- É geralmente utilizado para solicitar dados do servidor.

**POST=>** ele não passa os dados pela URL

Não consigo acessar rota do tipo POST pela url, isso evita expor os dados

- Os dados do formulário são incluídos no corpo da requisição HTTP.
- Os dados não são visíveis na barra de endereços do navegador.
- É adequado para envio de grandes quantidades de dados.
- É geralmente utilizado para enviar dados sensíveis ou quando a quantidade de dados é grande demais para ser convenientemente incluída na URL.

## 18. Semântica

Semântica em HTML se refere ao uso apropriado das tags HTML para transmitir o significado e a estrutura correta do conteúdo de uma página web. Em outras palavras, trata-se de utilizar as tags HTML de forma que elas descrevam o propósito do conteúdo, em vez de apenas definir sua aparência visual.

Quando falamos em semântica em HTML, estamos nos referindo ao uso de tags que fornecem significado e contexto ao conteúdo, facilitando a compreensão tanto para os desenvolvedores quanto para os mecanismos de busca e navegadores.

**Exemplos de tags semânticas introduzidas no HTML5:**

**<header>** Define o cabeçalho de uma seção ou página.

**<footer>** Define o rodapé de uma seção ou página.



**<nav>** Define uma seção de navegação.

**<article>** Define um conteúdo independente que pode ser distribuído e reutilizado.

**<section>** Define uma seção genérica de uma página.

Ao utilizar essas tags semânticas, os desenvolvedores podem criar páginas web mais claras, organizadas e acessíveis. Além disso, os mecanismos de busca podem entender melhor a estrutura e o significado do conteúdo, o que pode melhorar a classificação nos resultados de pesquisa.

## 19. Ênfase, Importância e Marcação

### Ênfase (Emphasis)

No contexto de HTML, a ênfase refere-se a destacar determinado texto para dar ênfase ou importância ao conteúdo. Isso pode ser feito visualmente com itálico, negrito, sublinhado ou outras formas de destaque.

Em HTML, a tag **<em>** é usada para indicar ênfase no texto. Por padrão, os navegadores geralmente exibem texto enfatizado em itálico, mas a apresentação exata pode variar dependendo do estilo definido pelo CSS.

Exemplo: `<em>Texto enfatizado</em>`

### Importância (Strong Importance)

A importância é semelhante à ênfase, mas indica um nível mais forte de destaque ou relevância para o conteúdo. Isso geralmente é expresso visualmente com negrito ou outra formatação mais proeminente.

Em HTML, a tag **<strong>** é usada para indicar a importância do texto. Por padrão, os navegadores geralmente exibem texto importante em negrito.

Exemplo: `<strong>Texto importante</strong>`

### Marcação (Marking)

A marcação é uma forma de realçar ou destacar um trecho específico de texto para torná-lo mais visível ou significativo para o usuário. Isso pode ser útil para destacar termos de pesquisa em resultados de pesquisa.

Em HTML, a tag **<mark>** é usada para marcar ou destacar o texto. Por padrão, os navegadores geralmente exibem o texto marcado com uma cor de fundo amarela.

Exemplo: `<mark>Texto marcado</mark>`

Juntos, esses elementos de marcação permitem que os desenvolvedores forneçam ênfase, importância e destaque a diferentes partes do conteúdo de uma página HTML, melhorando assim a compreensão e a usabilidade para os usuários, dando semântica ao código HTML.