

FICHAMENTO

Navegadores (Browsers) e Funcionamento

- Definição de Navegador: É um programa de computador que permite aos usuários visualizar páginas da web.
- Interação: Os navegadores também permitem a interação com as páginas, como enviar formulários, clicar em links e digitar texto.
- Função Principal: Não importa qual navegador seja utilizado, todos fazem a mesma coisa: pegam arquivos HTML, CSS e JavaScript e os renderizam em um site.

A Estrutura dos Sites Modernos e Antigos

- Sites Modernos: A maioria é criada usando a combinação dos três tipos de arquivos (HTML, CSS, JavaScript).
- Regra de Essencialidade: Não é possível criar um site apenas com CSS ou apenas com JavaScript. No entanto, é possível criar um site apenas com um arquivo HTML.
- Primeiros Sites: De fato, os primeiros sites foram criados utilizando apenas HTML.

O que é HTML?

- Definição: HTML é o que define a estrutura do website.
- Significado: HTML é a sigla para Hypertext Markup Language.

A. Hypertext (Hipertexto)

- Conceito: Refere-se a pedaços de texto que podem ser vinculados a outros documentos no website.
- Base do Funcionamento: É a base do funcionamento de um website HTML.
- Exemplo Prático: No primeiro website criado por Tim Berners-Lee (também criador da internet), os Hyperlinks (marcados em azul) levavam a um redirecionamento para outro documento HTML ao serem clicados.

B. Markup Language (Linguagem de Marcação)

- Conceito: Funciona de modo similar ao uso de aspas em português: a presença de elementos informa ao leitor que o texto é, por exemplo, uma citação.
- Implementação: Para usar a marcação com HTML, é necessário recorrer às tags HTML.

HTML Não é Linguagem de Programação

- Razão Principal: A principal finalidade do HTML é estruturar o conteúdo de uma página web, definindo a hierarquia e os elementos.
- Falta de Recursos: HTML não possui recursos para executar algoritmos, tomadas de decisões ou operações matemáticas complexas. Ele não é capaz de realizar cálculos, manipular variáveis ou executar *loops*, que são características essenciais das linguagens de programação.

HTML5 e HTML Semântico

- HTML5: É a versão mais recente do HTML.
- Principais Novidades do HTML5:
 - Introduziu novos elementos semânticos que tornam a estrutura da página mais clara e significativa.
 - Adicionou suporte nativo para elementos de áudio (`<audio>`) e vídeo (`<video>`).
- HTML Semântico: É uma abordagem que consiste em utilizar as tags HTML de forma mais significativa e descritiva, dando um significado mais claro ao conteúdo.
 - Benefícios: Melhora a acessibilidade, SEO (otimização para mecanismos de busca) e a usabilidade do site, tornando o código mais legível para humanos e máquinas.

Elemento Heading: Definição e Tags

- Definição: O elemento *Heading* serve para criar títulos ou subtítulos que você deseja exibir em uma página web.
- Tags: São definidos usando as tags que vão de `<h1>` (o título mais importante) até `<h6>` (o título menos importante).
- Função Estrutural: A ideia é que os *Headings* funcionem como o sumário de um livro, definindo a estrutura hierárquica do conteúdo.

Importância e Boas Práticas

- Estrutura do Documento: É crucial usar os *Headings* para mostrar a estrutura correta do documento.
- Mecanismos de Busca (SEO): Os mecanismos de pesquisa (como o Google) usam esses cabeçalhos para indexar a estrutura e o conteúdo das suas páginas web.
- Experiência do Usuário: Os usuários costumam percorrer uma página rapidamente por seus títulos; a estrutura clara melhora a usabilidade.
- Hierarquia:

- Utilize o `<h1>` para o título principal da página (o mais importante).
- Siga a hierarquia de forma lógica (`<h1>` deve ser seguido por `<h2>`, depois `<h3>`, e assim por diante), não pulando níveis (ex: não vá de `<h1>` para `<h3>` diretamente).

Diferença entre Tag e Elemento HTML

- Tag: É tudo que está entre os sinais de menor que e maior que (`<` e `>`). Exemplos são a tag de abertura (`<h1>`) e a tag de fechamento (`</h1>`).
- Elemento HTML: O elemento é a combinação da tag de abertura, do conteúdo (o texto ou o que estiver dentro) e da tag de fechamento. Exemplo:
`<h1>Conteúdo</h1>`.

Elemento Paragraph (`<p>`)

- Definição: O elemento *Paragraph* serve para criar parágrafos em uma página HTML.
- Comportamento: Um parágrafo sempre começa em uma nova linha e é tipicamente um bloco de texto.
- Necessidade: Sem a tag `<p>`, não seria possível determinar onde um parágrafo termina e outro começa na estrutura do documento.

Formatação de Texto

O HTML contém vários elementos para definir o texto com um significado especial (semântica):

- ``: Exibe o texto em negrito (foco visual, sem importância semântica).
- ``: Marca o texto como importante (ênfase semântica forte).
- `<i>`: Exibe o texto em itálico (ex: termo técnico, frase em outro idioma).
- ``: Marca o texto como enfatizado.
- `<small>`: Exibe o texto em tamanho menor (ex: para textos laterais ou direitos autorais).
- `<sub>`: Exibe o texto subscrito (abaixo da linha-base, ex: H₂O).
- `<sup>`: Exibe o texto sobrescrito (acima da linha-base, ex: x²).

Tags de Fechamento Automático

- Definição: São tags que não possuem tag de fechamento.
- Conteúdo: Elas não possuem conteúdo entre tags, ou seja, são elementos vazios.
- Sintaxe (Opções): As duas representações funcionam da mesma forma:
 - `</hr />` ou `</hr>`

- `
` ou `
`

Elemento Horizontal Rule (`<hr />`)

- Função: Cria uma régua horizontal (linha).
- Uso: É usada para separar conteúdos que não são relacionados na página.

Elemento Break (`
`)

- Função: É usado para quebrar linhas em texto.
- Uso: Serve para dar um sentido apropriado ao texto, garantindo que o conteúdo seja exibido em linhas separadas onde for necessário (ex: quebrar linhas em poemas ou endereços).

Listas

- Definição: Listas são estruturas frequentemente encontradas ao navegar na internet (Ex: Top 10 linguagens de programação, Melhores cidades para morar).
- Tipos Abordados: Listas Não Ordenadas e Listas Ordenadas.

Listas Não Ordenadas

- Tag Principal: *Unordered List* (``).
- Propósito: São listas onde a ordem dos itens não precisa ser definida.
- Estrutura do Item: Cada item dentro da lista deve ser colocado entre a tag *List Item* (``).
- Exibição Padrão: Por padrão, as listas não ordenadas são exibidas com "bullet points" (marcadores).

Exemplo Básico de Estrutura:

```
<ul>
  <li>Primeiro Item</li>

  <li>Segundo Item</li>

  <li>Terceiro Item</li>
</ul>
```

Listas Ordenadas

- Tag Principal: *Ordered List* (``).
- Propósito: São listas onde a ordem dos itens precisa ser definida (a ordem é relevante).
- Estrutura do Item: Os itens de lista também precisam ser colocados entre a tag *List Item* (``), dentro da tag ``.
- Exibição Padrão: Por padrão, as listas ordenadas são exibidas com números, iniciando em '1'.

Exemplo Básico de Estrutura:

```
<ol>
  <li>Primeiro Passo/Item</li>

  <li>Segundo Passo/Item</li>

  <li>Terceiro Passo/Item</li>
</ol>
```

Elemento Anchor (Âncora) (`<a>`)

- Função: Adiciona hyperlinks (links) ao documento.
- Link Ativo: A tag `<a>` por si só não torna o link ativo; para isso, é necessário um atributo.
- Atributo Específico: Para tornar o link ativo, o atributo `href` deve ser adicionado à tag de abertura.
- Valor do Atributo: O valor do atributo `href` é a URL para onde o hyperlink será redirecionado (Ex: para a página inicial do Google).

Atributos HTML

- Definição: São características ou propriedades de um elemento HTML que fornecem informações adicionais sobre esse elemento.
- Uso: São usados para definir ou modificar o comportamento e a aparência dos elementos.
- Localização: São sempre especificados na tag de abertura do elemento.
- Estrutura: Consistem em um nome e um valor (o valor é opcional em alguns casos).
 - Sintaxe:

`<elemento nome-do-atributo="valor-do-atributo">...</elemento>`.

Outros Atributos de Lista ()

- **start**: Em uma lista ordenada (), o atributo **start** é usado para especificar o número inicial do contador de itens da lista.
- Comportamento Padrão: Normalmente, as listas ordenadas são numeradas sequencialmente a partir de 1, mas o atributo **start** permite alterar esse comportamento.
 - Exemplo: `<ol start="5">` iniciaria a lista a partir do item 5 (ou número 5).

Elemento Image ()

- Função: Adiciona imagens ao documento HTML.
- Tag de Fechamento: É uma tag de fechamento automático (não possui tag de fechamento).
- Atributo **src** (Source): É um atributo obrigatório que indica o caminho (URL ou arquivo local) da imagem a ser exibida.
 - O valor de **src** pode ser uma URL de uma imagem da web.
 - O valor de **src** também pode ser o caminho para uma imagem dentro do próprio projeto (Ex: `"img/cachorro pretofofinho.jpg"`).
- Atributo **alt** (Alternative Text Description): É um texto alternativo que descreve a imagem.
 - É importante para a acessibilidade de deficientes visuais que utilizam leitores de tela.
 - O leitor de tela lê a descrição fornecida como valor do atributo **alt**.

Elemento Iframe (<iframe>)

- Função: O elemento `<iframe>` é usado para incorporar outro HTML à página HTML.
- Atributo **src** (Source): Precisa do atributo **src** para indicar o caminho (URL) do arquivo HTML a ser incorporado.
 - É comumente usado para incorporar vídeos do YouTube, onde o código de incorporação contém a tag `<iframe>`.

File Path (Caminho do Arquivo)

- Conceito: É o caminho usado para encontrar a localização exata de um arquivo ou mesmo de um diretório (pasta).
- Contexto: É similar a informar o caminho para chegar a uma localização física no mundo.
- Ponto de Partida: O ponto de partida do sistema de arquivos é o Root (raiz) do Disco Rígido (Ex: **C:** em muitos sistemas).

- Tipos de Caminhos: Existem dois tipos principais de caminhos: Absoluto e Relativo.

Caminho Absoluto

- Definição: É o caminho que especifica o local exato de um arquivo ou diretório a partir do diretório raiz do sistema (Root).
- Exemplo: Para acessar o arquivo `gatinho.png`, o caminho absoluto seria `C:/Projeto/img/gatinho.png`.

Caminho Relativo

- Definição: É o caminho que especifica o local de um arquivo ou diretório a partir do local de trabalho atual.
- Vantagem: Para desenvolvimento web, é considerado muito mais prático trabalhar com caminhos relativos.
- Símbolos Usados:
 - Ponto (`./`): O ponto antes da barra (`./`) indica que o arquivo ou pasta está sendo acessado a partir do diretório atual.
 - *Exemplo (mesma pasta)*: Para acessar `cachorrinho.jpg` na mesma pasta, o caminho é `./cachorrinho.jpg`.
 - *Exemplo (subpasta)*: Para acessar `gatinho.jpg` dentro da pasta `img` (que está no mesmo nível), o caminho é `./img/gatinho.jpg`.
 - Dois Pontos (`../`): Os dois pontos antes da barra (`../`) indicam que é necessário subir um nível na estrutura de diretórios para acessar o arquivo desejado.
 - *Exemplo (subir um nível)*: Para acessar `trabalho.docx` a partir de uma subpasta, o caminho seria `../trabalho.docx`.

Estrutura para Múltiplas Páginas

- Necessidade: Para criar um website com múltiplas páginas, é preciso ter outros arquivos HTML além do principal (Ex: `index.html`, `sobre.html`, `contato.html`).
- Pasta Raiz: É a pasta principal do seu projeto web, onde todos os arquivos e pastas necessários para o site são colocados.

Organização de Arquivos e Pastas

- `index.html`:
 - Todo website precisa ter uma página `index.html`, que é o arquivo principal do site.
 - Geralmente, é a página inicial que os visitantes veem quando acessam o site.

- Pode conter links para outras páginas.
- Pasta **assets**:
 - Usada para armazenar recursos estáticos que o site precisa, como imagens, estilos (CSS), scripts (JavaScript), fontes, etc..
- Pasta **img**:
 - Uma pasta específica usada para armazenar imagens.
- Pasta **public**:
 - Geralmente contém outras páginas HTML que são acessíveis publicamente a partir do seu site.
- Benefício: Essa organização ajuda a manter os arquivos do projeto estruturados e facilita a manutenção e expansão do site no futuro.

Navegação entre Páginas

- Links: Para navegar entre as páginas, é necessário utilizar o elemento links (`<a>`).
- Sintaxe de Exemplo: Para ir da `index.html` para a `sobre.html` (dentro da pasta `public`), usa-se o caminho relativo:

```
<a href = "../public/sobre.html">Sobre mim</a>.
```

- Atenção: É crucial ficar atento ao caminho dos arquivos (file paths) ao criar os links.

Elemento `<div>` (Division)

- Função: É usado como um container para outros elementos.
- Uso Comum: Normalmente utilizado para agrupar seções em uma página da web.
- Propósito: Ajuda a organizar o conteúdo em blocos lógicos.

Elemento ``

- Função: É usado como um container para uma parte de um texto ou documento.
- Uso: Diferentemente do `div`, que é um elemento de bloco, o `span` é um elemento *inline* (em linha). Ele permite aplicar estilos a uma pequena porção de texto sem afetar os elementos vizinhos.

Elemento Pai (Container)

- Definição: Também conhecido como elemento container, é um termo usado para se referir a um elemento que contém outros elementos dentro dele (elementos filhos).
- Importância: Desempenha um papel fundamental na estruturação e organização do conteúdo em uma página web.

- Benefício: Permite que os elementos filhos sejam agrupados de forma lógica, facilitando a leitura e a compreensão do conteúdo.

Exemplos de Relação Pai-Filho:

- O elemento `<body>` é o pai do `<h1>` e das duas tags `<div>` que o seguem.
- Uma tag `<div>` é o pai do `<h2>` e da tag `` que estão contidos nela.
- Uma tag `` é o pai das tags `` (itens de lista) que estão dentro dela.

Importância do Elemento Pai

O elemento pai é importante para:

- Agrupamento de elementos.
- Aplicação de estilos (geralmente através de seletores CSS).
- Manipulação de elementos filhos (geralmente via JavaScript).