

SENAC
Campus Santo Amaro

TADS - Análise Desenvolvimento de Sistemas

PW - Programação Web



Aula#3 HTML Básico

Professor: Veríssimo - carlos.hypereira@sp.senac.br

28/02/2024

Sobre este documento

Este documento objetiva deixar registrado o conteúdo abordado em sala de aula pelo professor. Importante destacar que a Nota de Aula serve como guia ao professor, bem como serve aos alunos como um norte, quanto ao conteúdo desenvolvido em sala de aula.

Este documento não tem a pretensão de ser uma única fonte para estudo. Para tal, o aluno deverá assistir às aulas e fazer uso (consulta) à bibliografia recomendada na ementa da disciplina, e à bibliografia complementar, apontada pelo professor.

Preâmbulo da Aula

Nesta aula abordaremos os conceitos fundamentais do Hyper-Text Markup Language (*HTML*), onde iremos explorar os fundamentos fundamentais e elementos básicos. O HTML é a base sobre a qual a grande maioria dos sites é construída, e compreender seus conceitos é essencial para os interessados em desenvolvimento web.

Nesta aula, você irá aprender os princípios básicos do HTML, desde sua estrutura fundamental até a criação de elementos e organização de conteúdo em suas páginas. Vamos explorar como usar tags para definir diferentes tipos de conteúdo, como texto, imagens, links e listas. Além disso, vamos discutir a importância de atributos e como eles podem enriquecer seus elementos HTML.

Importante destacar que esta aula possui uma abordagem prática, na qual os elementos conceituais servem de guia para a parte prática da aula.

Contents

| | | |
|----------|--|-----------|
| 1 | Introdução | 1 |
| 1.1 | Tags ou marcações HTML | 2 |
| 1.2 | Estrutura de um Documento HTML | 3 |
| 1.3 | Manipulação de um Documento HTML | 4 |
| 2 | Precedentes Históricos | 5 |
| 2.1 | A evolução do HTML | 5 |
| 2.2 | XML | 6 |
| 2.2.1 | Representação estruturada dos dados . . | 7 |
| 3 | Fundamentos do HTML | 11 |
| 3.1 | Características gerais de HTML | 11 |
| 3.2 | Criando e salvando um arquivo HTML | 12 |
| 3.3 | A estrutura de um documento HTML | 13 |
| 3.3.1 | Elemento do HEAD | 15 |
| 3.3.2 | Elemento de BODY | 17 |
| 3.4 | Elementos HTML | 18 |
| 3.4.1 | Elementos HTML aninhados | 18 |
| 3.5 | Cabeçalhos HTML | 19 |
| 3.6 | Parágrafos | 19 |

| | | |
|----------|--|-----------|
| 3.7 | Formatar texto | 20 |
| 3.7.1 | Elementos de Formatação HTML | 20 |
| 3.8 | Atributos HTML | 21 |
| 3.8.1 | O atributo href | 22 |
| 3.8.2 | O atributo src | 22 |
| 3.8.3 | O atributo alt | 22 |
| 3.8.4 | O atributo style | 23 |
| 3.8.5 | O atributo lang | 23 |
| 3.8.6 | O atributo title | 24 |
| 3.9 | A tag de Comentário | 24 |
| 4 | Elementos mais avançados | 25 |
| 4.1 | Tabelas | 25 |
| 4.1.1 | Células da Tabela | 27 |
| 4.1.2 | Linhas Tabelas | 28 |
| 4.2 | Listas | 31 |
| 4.2.1 | Lista HTML não ordenada | 31 |
| 4.2.2 | Lista HTML ordenada | 32 |
| 4.3 | Classes | 34 |
| 4.3.1 | Usando o atributo de classe | 34 |
| 5 | Formulários | 39 |
| 6 | JavaScript | 41 |
| 6.1 | A tag HTML <script> | 42 |

Chapter 1

Introdução

A linguagem HTML (*Hypertext Markup Language*) tem o objetivo de formatar textos através de marcações especiais denominadas **tags**, para que possam ser exibidos de forma conveniente pelos *browsers*. O HTML também possibilita a interligação entre páginas da Web, criando assim documentos com o conceito de **hipertexto**

Características básicas do HTML:

- São arquivos de texto escritos em ASCII ¹
- Não diferencia entre letras maiúsculas e minúsculas em suas marcações
- Arquivos HTML podem possuir as extensões *html* ou *htm*.
- Documentos html são feitos para prover **estrutura lógica da informação** destinada à apresentação de páginas da rede mundial de computadores.

¹ASCII (American Standard Code for Information Interchange)

- A linguagem html contém um conjunto de **tags** com um número fixo para definir a estrutura do documento, e cada tag tem a sua *semântica* já definida.

CSS (Cascading Style Sheets)

O CSS (Cascading Style Sheets) permite a separação da estrutura lógica da aparência da página. Mas, embora o layout possa ser separadamente definido no CSS, o html é destinado especificamente para hipertexto, e não para informação em geral!



1.1 Tags ou marcações HTML

A linguagem HTML pertence a uma classe de linguagens de programação conhecida como *Tag Languages*, ou simplesmente **Linguagem de Marcação**, onde os comandos são escritos em forma de marcações denominadas **tags**. As tags HTML são como palavras-chave que definem como o navegador Web irá formatar e exibir o conteúdo. São representadas com o símbolo de **chevron** `< >`.

No interior de uma *tag* tem propriedades diferentes, os atributos, que darão os comandos de criação do documento HTML, descrevendo as características dos elementos.

```
1 <html>
2   <head>   </head>
3   <body>   </body>
4 </html>
```

Listing 1.1: Exemplos de *tag* HTML

1.2 Estrutura de um Documento HTML

A estrutura principal de um documento HTML possui duas partes básicas:

- Cabeçalho
- Corpo do programa.

Atenção aos seguintes pontos:

- Todo documento deve ser iniciado com a marcação **<HTML>** e finalizado com **</HTML>**.
- A área do cabeçalho, **<HEAD>** e **</HEAD>**, é opcional
 - Dentro temos o título da página **<TITLE>** e **</TITLE>** (exibido na barra de título do Browser).
- Corpo do programa e inserido nas tags **<BODY>** e **</BODY>**.

Analisar à figura 1.1, uma típica estrutura de página html.

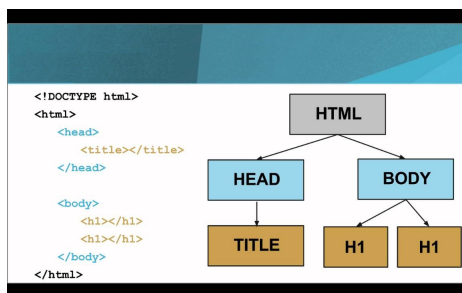


Figure 1.1: Estrutura de página HTML

1.3 Manipulação de um Documento HTML

Um documento HTML podem ser criados em qualquer editor de texto visual (ou até mesmo em um bloco de notas). Em nosso curso utilizaremos o *Visual Studio Code (VS Code)*, que é um editor de código de código aberto desenvolvido pela **Microsoft**. (<https://code.visualstudio.com/>)

Chapter 2

Precedentes Históricos

2.1 A evolução do HTML

Essa linguagem foi desenvolvida em 1992 por Tim Berners Lee e Robert Caillau no *CERN*, que é o **European Concil for Nuclear Research**. O html é um exemplo do SGML (Standard Generalized Markup Language). Originalmente o html definia estritamente a estrutura lógica de um documento, e não a sua aparência física. Mas, com a pressão dos usuários (principalmente da indústria), as versões posteriores do html foram forçadas a prover cada vez mais e mais controle da aparência do documento.

Alguns marcos importantes:

- 1969: *ARPANET*-Surge a internet nos USA
- 1991: Surgimento do - *CERN* (European Concil for Nuclear Research) html foi concebido
- 1992: Primeira Publicação do HTML
- 1993: algumas definições físicas da aparência, tabelas, formulários e equações matemáticas (HTML+)

- -1994: HTML 2.0 (padrão para as características principais) e 3.0 (uma extensão do HTML+, entendido como um rascunho de padrão).
- 1995 e 1996: Netscape e Internet Explorer definem seus próprios padrões e surge o HTML 3.2 baseado nas implementações correntes.
- 1997: O HTML 4.0 é desenvolvido separando a apresentação da estrutura com style sheets: Browser **Explorer 4.0** - CSS, JavaScript e DOM.
- 1998: CSS2 - Propriedades do CSS1 + 70 novas propriedades
- 1999: Definição do HTML 4.01 (suaves modificações da versão anterior).
- 2000: O XHTML 1.0 é criado, o qual consiste de uma versão XML do HTML 4.01.
- 2008: É lançado pela *W3C* o HTML5

2.2 XML

Extensible Markup Language (XML) é linguagem de marcação de dados (meta-markup language) que provê um formato para descrever dados estruturados. Isso facilita declarações mais precisas do conteúdo e resultados mais significativos de busca através de múltiplas plataformas. O XML também vai permitir o surgimento de uma nova geração de aplicações de manipulação e visualização de dados via internet. O XML permite a definição de um número infinito de tags. Enquanto no HTML, se as tags podem ser usadas para definir a formatação de caracteres e parágrafos, o XML provê um sistema para criar tags

para dados estruturados. Um elemento XML pode ter dados declarados como sendo preços de venda, taxas de preço, um título de livro, a quantidade de chuva, ou qualquer outro tipo de elemento de dado. Como as tags XML são adotadas por intranets de organizações, e também via Internet, haverá uma correspondente habilidade em manipular e procurar por dados independentemente das aplicações onde os quais são encontrados. Uma vez que o dado foi encontrado, ele pode ser distribuído pela rede e apresentado em um browser como o Internet Explorer 5 de várias formas possíveis, ou então esse dado pode ser transferido para outras aplicações para processamento futuro e visualização.

Veja à figura 2.1 Exemplo de uma estrutura de mensagem XML :

```
<?xml version="1.0"?>
<DOC xmlns="http://www.bcb.gov.br/XXX/YYYYYYY.xsd">
  <BCMSG>
    . . . Segmento de Controle
  </BCMSG>
  <SISMSG>
    . . . Segmento do Sistema
  </SISMSG>
  <USERMSG>
    . . . Segmento do Usuário
  </USERMSG>
</DOC>
```

Figure 2.1: Estrutura de Mensagens do Banco Central do Brasil

2.2.1 Representação estruturada dos dados

O XML provê uma representação estruturada dos dados que mostrou ser amplamente implementável e fácil de ser desenvolvida. Implementações industriais na linguagem SGML (Standard Generalized Markup Language) mostraram a qualidade

intrínseca e a força industrial do formato estruturado em árvore dos documentos XML. O XML é um subconjunto do SGML, o qual é otimizado para distribuição através da web, e é definido pelo Word Wide Web Consortium(W3C), assegurando que os dados estruturados serão uniformes e independentes de aplicações e fornecedores. O XML provê um padrão que pode codificar o conteúdo, as semânticas e as esquematizações para uma grande variedade de aplicações desde simples até as mais complexas, dentre elas:

- Um simples documento.
- Um registro estruturado tal como uma ordem de compra de produtos.
- Um objeto com métodos e dados como objetos Java ou controles ActiveX.
- Um registro de dados. Um exemplo seria o resultado de uma consulta a bancos de dados.
- Apresentação gráfica, como interface de aplicações de usuário.
- Entidades e tipos de esquema padrões.
- Todos os links entre informações e pessoas na web.

Uma característica importante é que uma vez tendo sido recebido o dado pelo cliente, tal dado pode ser manipulado, editado e visualizado sem a necessidade de “invocar” o servidor. Dessa forma, os servidores tem menor sobrecarga, reduzindo a necessidade de computação e reduzindo também a requisição de banda passante para as comunicações entre cliente e servidor.

O XML é considerado de grande importância na Internet e em grandes intranets porque provê a capacidade de interoperação dos computadores por ter um padrão flexível e aberto

e independente de dispositivo. As aplicações podem ser construídas e atualizadas mais rapidamente e também permitem múltiplas formas de visualização dos dados estruturados.

Chapter 3

Fundamentos do HTML

Trataremos nesta aula alguns elementos básicos do HTML. Fica como sugestão um aprofundamento do assunto, pois o objetivo é fazer uma abordagem ampla: <https://www.w3schools.com/html/> abordagem ampla: <https://www.w3schools.com/html/>

3.1 Características gerais de HTML

Uma página HTML é um documento composto de **textos e códigos especiais** denominados **Tags** que possibilitam a exibição de um documento na web. Mais do que informações textuais, estes documentos podem conter imagens, sons animações e até vídeos.

Uma página WEB pode se ligar a outra por meio de links endereçáveis para locais dentro do seu micro ou em qualquer parte do mundo. O que chamamos de HTML é basicamente uma linguagem simples que tem como princípio fundamental formatar textos criando uma aparência agradável, e promover ligações ou links com outras páginas. Embora simples, não foge às regras clássicas que as linguagens de programação de

computador exigem, ou seja:

HTML requer digitação de código fonte baseado em regras de sintaxe e formatos pré-definidos

O código fonte uma vez pronto é interpretado pelo **navegador** que se encarregará de executar os comandos colocados dentro de seu programa para acessar os recursos da WEB. Para escrever um arquivo HTML, o mais simples dos editores disponível pode ser usado. Podemos utilizar o Bloco de Notas (Notepad) por estar disponível em todas as versões do Windows. Uma vez que seu HTML for digitado, ele deverá ser visualizado ou conferido em um browser.

3.2 Criando e salvando um arquivo HTML

Para se programar em HTML, você deverá inicialmente seguir a lista de procedimentos abaixo:

- Abra sua pasta de trabalho.
- Abra o programa Bloco de Notas.
- Escreva as linhas que farão parte de seu código fonte HTML.
- Na primeira vez que estiver salvando o arquivo, use a opção Arquivo + Salvar Como.
- De um nome e não se esqueça de colocar a extensão **.HTML** (ponto HTML)

Após gravar seu arquivo, abra ou alterne até sua pasta e veja com ele ficou. Observe os ícones à figura 3.1. Caso seu arquivos

esteja com o ícone do IExplorer (pode estar sendo substituído pelo ícone de outro navegador que você tenha instalado), você gravou um HTML de maneira correta. Clicando sobre seu arquivo para abri-lo, será carregado o navegador padrão. Se o ícone parece um bloco de notas, você pode ter esquecido um dos passos acima. Verifique. Clicando no caderninho para abri-lo você verá que apenas o bloco de nota será carregado.

figura Exemplo de uma estrutura de mensagem XML :

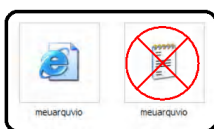


Figure 3.1: Visão dos Arquivos html gravados

3.3 A estrutura de um documento HTML

Um arquivo HTML é construído a partir de um código fonte, composto de texto puro em formato *ASCII*¹. Um programa HTML é dividido em três partes básicas :

- A estrutura principal
- O cabeçalho
- O corpo do programa.

¹ASCII: (American Standard Code for Information Interchange) Conjunto de normas para a representação números binários. Este sistema é amplamente utilizado em informática, telecomunicações, programação, etc.

Todo programa HTML deve começar com o comando `<HTML>` e ser encerrado com o comando `</HTML>`. A área de cabeçalho é opcional e é delimitada pelo par de comandos `<HEAD>` `</HEAD>`. A maioria dos comandos será colocada na área corpo do documento formado pelos comandos `<BODY>` `</BODY>`.

Esta estrutura é demonstrada à figura 3.2

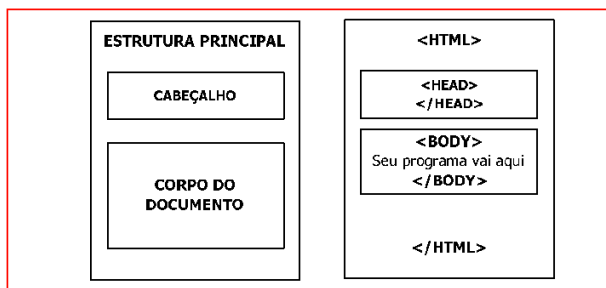


Figure 3.2: Estrutura de uma página HTML

A maioria dos elementos HTML segue uma estrutura hierárquica. Há uma estrutura básica (Mínima) para uma página HTML, construída de acordo com a especificação padrão, que deve ser respeitada para garantir uma compatibilidade com o maior número de browsers possível. Conforme demonstrado no código HTML 3.7

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE
6         =edge">
7     <meta name="viewport" content="width=device-
8         width, initial-scale=1.0">
9     <title>Document</title>
10 </head>
11 <body>
```

```
10  
11 </body>  
12 </html>
```

Listing 3.1: Estrutura Básica HTML

A linha 1 é um descritor no início informa que estamos usando o **HTML5**, e o restante do código HTML está contido dentro das tags `<html>`, `<head>` e `<body>`, seguindo as convenções padrão do HTML.

O trecho `<!DOCTYPE html>` é conhecido como um declarador de tipo de documento (Document Type Declaration) em HTML. Ele é usado para definir o tipo de documento e a versão do HTML que está sendo usada na página web. O declarador de tipo de documento não é tecnicamente uma tag HTML, mas é uma parte fundamental de um documento HTML.

A função principal do `<!DOCTYPE html>` é informar ao navegador ou ao interpretador de HTML como a página deve ser renderizada e tratada. Ele define o padrão para o navegador entender qual versão do HTML está sendo utilizada e quais regras de renderização e comportamento devem ser aplicadas.

No HTML5, o `<!DOCTYPE html>` é a declaração padrão e é recomendado para todos os documentos HTML. Ele é simples e genérico, permitindo que os navegadores saibam que o documento está seguindo as especificações do HTML5.

3.3.1 Elemento do HEAD

O bloco do cabeçalho, marcado por `<HEAD>` e `</HEAD>` pode conter informações sobre o conteúdo do documento utilizada para fins de indexação e organização. Não contém informação que será exibida na página.

TITLE

O elemento `<title>` é usado para definir o título da página, que é exibido na barra de título da janela do navegador ou na guia da página. Além disso, o conteúdo do `<title>` é usado pelos mecanismos de busca para exibir o título da página nos **resultados da pesquisa**. É uma parte essencial de qualquer documento HTML.

META

O elemento `<meta>` é usado para fornecer metadados sobre a página, como a codificação de caracteres, descrições, palavras-chave e outras informações relevantes para os mecanismos de busca e outros processos de indexação. Um exemplo comum é a definição da codificação de caracteres.

LINK

O elemento `<link>` é usado para vincular recursos externos à página, como folhas de estilo CSS, ícones e fontes. Ele não exibe conteúdo diretamente, mas estabelece conexões com recursos que afetam a aparência e o comportamento da página. Um exemplo comum é a inclusão de uma folha de estilo.

BASE

O elemento `<base>` é usado para definir uma URL base para todos os URLs relativos dentro do documento. Isso significa que os URLs relativos usados na página serão resolvidos a partir desta URL base. Ele é útil quando você deseja que todos os links dentro de uma página sejam tratados como relativos a uma determinada localização.

3.3.2 Elemento de BODY

O bloco maracado por `<BODY>` e `</BODY>` contém a parte do documento onde está colocada a informação que efetivamente será mostrada e formatada na tela pelo browser. Todos os elementos que serão apresentados daqui em diante tratam da formatação do texto (ou imagem) da página.

ElementosHTML podem ser de vários tipos e têm regras específicas sobre o que podem conter e onde podem ser usados. Quanto à estrutura que assumem na pagina, podem ser classificados da seguinte forma:

- Elementos de bloco (contém texto ou agrupam outros elementos de bloco)
 - `<div>` (Divisão)
 - `<section>` (Seção)
 - `<article>` (Artigo)
 - `<blockquote>` (Citação em bloco)
 - `<figure>` e `<figcaption>` (figura e legenda)
- Elementos *inline* (em linha - Usados dentro do texto)
 - `` - Usado para aplicar estilos ou manipular parte do texto.
 - `` Indica um texto importante que deve ser enfatizado em negrito.
 - `` - Indica ênfase em um texto, normalmente exibido em itálico.
 - `<a>` - Cria links para outras páginas ou recursos
 - `
` - Insere uma quebra de linha
 - `<code>` - Exibe código inline
- Elementos *lista* (organizar itens em uma sequência)

- `` - Lista não ordenada (marcadores).
- `` - Lista ordenada (números ou letras).
- Elementos *tabela* (construir tabelas)
 - `<tr>` - para linhas
 - `<td>` - para células de dados.
- Elementos *formulário* (construir formulários)

3.4 Elementos HTML

O elemento HTML é tudo, desde a tag inicial até a tag final:

`<nome da tag>CONTEÚDO</nome da tag>`

3.4.1 Elementos HTML aninhados

Os elementos HTML podem ser aninhados (isso significa que os elementos podem conter outros elementos).

Todos os documentos HTML consistem em elementos HTML aninhados.

O exemplo listagem 3.2 contém quatro elementos HTML (`<html>`, `<body>`, `<h1>` e `<p>`):

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4
5     <h1>Cabe alho </h1>
6     <p>My este      um par grafo </p>
7
```



```
8     </body>  
9 </html>
```

Listing 3.2: Aninhamento de elementos HTML

3.5 Cabeçalhos HTML

Os cabeçalhos HTML são definidos com as tags `<h1>` a `<h6>`.

importante destacar que `<h1>` define o cabeçalho mais importante, e `<h6>` define o cabeçalho menos importante.

Veja na listagem 3.3 um exemplo com a definição dos seis cabeçalhos.

```
1 <!DOCTYPE html>  
2 <html>  
3     <body>  
4         <h1>Cabecalho nivel 1</h1>  
5         <h2>Cabecalho nivel 2</h2>  
6         <h3>Cabecalho nivel 3</h3>  
7         <h4>Cabecalho nivel 4</h4>  
8         <h5>Cabecalho nivel 5</h5>  
9         <h6>Cabecalho nivel 6</h6>  
10    </body>  
11 </html>
```

Listing 3.3: Aninhamento de elementos HTML

3.6 Parágrafos

Um parágrafo sempre começa em uma nova linha e geralmente é um bloco de texto.

O elemento HTML `<p>` define um parágrafo.

Um parágrafo sempre começa em uma nova linha e os navegadores adicionam automaticamente algum espaço em branco (uma margem) antes e depois de um parágrafo.

Veja na listagem 3.4 um exemplo com a definição dos seis cabeçalhos.

```
1 <!DOCTYPE html>
2   <html>
3     <body>
4       <p>Este      um paragrafo.</p>
5       <p>Este      um paragrafo novo.</p>
6       <p>Este      um outro paragrafo novo.</p>
7     </body>
8 </html>
```

Listing 3.4: parágrafo s HTML

3.7 Formatar texto

O HTML contém vários elementos para definir o texto com um significado especial.

3.7.1 Elementos de Formatação HTML

Os elementos de formatação foram projetados para exibir tipos especiais de texto:

- `` - teste negrito
- `` -texto importante

- `<i>` - Italico
- `` - ênfase
- `<mark>` - texto marcado
- `<small>` - letra pequena
- `` - texto deletado
- `<ins>` - texto inserido
- `<sub>` - texto subscrito
- `<sup>` - texto sobrescrito

Veja na listagem 3.5 um exemplo formatacao.

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <p>Este texto esta nomal.</p>
5     <p><b>Esta tag sabe escrever n<sup>2</sup>
      >.</b></p>
6   </body>
7 </html>
```

Listing 3.5: formatandos HTML

3.8 Atributos HTML

Os atributos HTML fornecem informações adicionais sobre os elementos HTML.

- Todos os elementos HTML podem ter atributos
- Os atributos fornecem informações adicionais sobre os elementos

- Os atributos são sempre especificados na *tag* de início
- Os atributos geralmente vêm em pares de nome/valor como: `name="value"`

3.8.1 O atributo href

A tag `<a>` define um *hiperlink*. O atributo **href** especifica a URL da página para a qual o link vai:

```
<a href="https://www.sp.senac.br/">Senac<a>
```

3.8.2 O atributo src

A tag `` é usada para incorporar uma imagem em uma página HTML. O atributo **src** especifica o caminho para a imagem a ser exibida::

```

```

Os atributos **width** e **height** A tag `` também deve conter os atributos **width** e **height**, que especificam a **largura** e a **altura** da imagem (em pixels):

```
< img src="logo_senac.jpg" width="500"  
height="600">
```

3.8.3 O atributo alt

O atributo **alt** necessário para a *tag* `` especifica um texto alternativo para uma imagem, se a imagem por algum motivo não puder ser exibida. Isso pode ocorrer devido a uma conexão lenta ou a um erro no atributo **src** ou se o usuário usar um leitor de tela.

```

```

3.8.4 O atributo style

O atributo **style** é usado para adicionar estilos a um elemento, como cor, fonte, tamanho e muito mais.

```
<p style="color:red;">Este parágrafo é  
vermelho.</p>
```

3.8.5 O atributo lang

Devemos sempre incluir o atributo **lang** dentro da *tag* `<html>`, para declarar o idioma da página da Web. Isso se destina a ajudar os motores de busca e navegadores.

O exemplo a seguir especifica o inglês como o idioma:

```
1 <!DOCTYPE html>  
2 <html lang="pt-BR">  
3 <body>  
4 ...  
5 </body>  
6 </html>
```

Listing 3.6: Utilizando atributo *tag*

Note que neste código os dois primeiros caracteres definem o idioma da página HTML e os dois últimos caracteres definem o país.

3.8.6 O atributo title

O atributo **title** define algumas informações extras sobre um elemento.

O valor do atributo **title** será exibido como uma dica de ferramenta quando você passar o mouse sobre o elemento:

**<p title="Esta é uma dica">
Este é um parágrafo.</p>**

Exemplo:

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4
5   <h2 title="Cabe alho">T tulo </h2>
6
7   <p title="Isto      uma dica">Passe o mouse
    sobre este par grafo para exibir o
    atributo de t tulo como uma dica de
    ferramenta.</p>
8
9   </body>
10 </html>
```

Listing 3.7: Utilizando atributo *tag*

3.9 A tag de Comentário

Podemos adicionar comentários à sua fonte HTML usando a seguinte sintaxe:

<!-- Escreva Aqui o Cmentário -->

Chapter 4

Elementos mais avançados

Neste capítulo demonstraremos elementos mais avançados HTML.

4.1 Tabelas

As tabelas HTML permitem que os desenvolvedores da Web organizem os dados em **linhas** e **colunas**.

Uma tabela em HTML consiste em **células** de tabela dentro de linhas e colunas. Podemos ver no código 4.1, um exemplo de tabela implementada em HTML.

```
1 <!DOCTYPE html>
2 <html>
3 <style>
4 table, th, td {
5     border:1px solid black;
6 }
7 </style>
8 <body>
9
10 <h2>Tabela b sica HTML</h2>
11
12 <table style="width:100%">
13     <tr>
14         <th>Empresa</th>
15         <th>Contato</th>
16         <th>Pa s </th>
17     </tr>
18     <tr>
19         <td>Castanha Verde</td>
20         <td>Assimi</td>
21         <td>Mali</td>
22     </tr>
23     <tr>
24         <td>Centro comercial Moctezuma</td>
25         <td>Hakainde</td>
26         <td>Z mbia </td>
27     </tr>
28 </table>
29
30 <p>Para entender melhor o exemplo, foram
    adicionadas bordas tabela.</p>
31
32 </body>
33 </html>
```

Listing 4.1: Exemplo de tabela

Veja na figura 4.1 o resultado do código:

Tabela básica HTML

| Empresa | Contato | País |
|----------------------------|---------|--------|
| Castanha Verde | Assani | Mali |
| Centro comercial Moctezuma | Hakande | Zâmbia |

Para entender melhor o exemplo, foram adicionadas bordas à tabela.

Figure 4.1: Estrutura de página HTML

4.1.1 Células da Tabela

Cada célula da tabela é definida por uma tag `<td>` e `</td>`. **td** significa *table data*

Tudo entre `<td>` e `</td>` são o **conteúdo da célula** da tabela.

Podemos ver no código 4.2, um exemplo de célula da tabela implementada em HTML.

```

1 <!DOCTYPE html>
2 <html>
3 <style>
4 table, th, td {
5     border:1px solid black;
6 }
7 </style>
8 <body>
9
10 <h2>Os elementos TD definem as células da
    tabela</h2>
11
12 <table style="width:100%">
13     <tr>
14         <td>celula#1</td>
15         <td>celula#2</td>
16         <td>celula#3</td>
17     </tr>
18 </table>
19

```

```
20 <p>Para entender melhor o exemplo, Foram  
    adicionadas bordas   tabela.</p>  
21  
22 </body>  
23 </html>
```

Listing 4.2: Exemplo Célula

Veja na figura 4.2 o resultado do código:

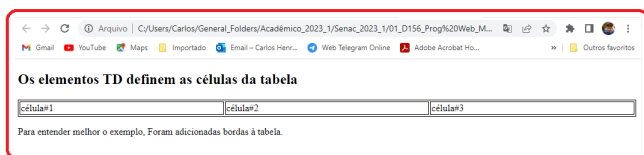


Figure 4.2: Estrutura de página HTML

Nota:

Uma célula de tabela pode conter todos os tipos de elementos HTML: texto, imagens, listas, links, outras tabelas, etc.

4.1.2 Linhas Tabelas

Cada linha (*row*) da tabela começa com um `<tr>` e termina com uma tag `</tr>`. **tr** significa *table row*

Podemos ver no código 4.3, uma estrutura simples de página HTML.

```
1  <!DOCTYPE html>
2  <html>
3  <style>
4  table, th, td {
5      border:1px solid black;
6  }
7  </style>
8  <body>
9
10 <h2>Os elementos <b><i>TR</i></b> definem as
    linhas da tabela</h2>
11
12 <table style="width:100%">
13     <tr>
14         <td>Linha1-1</td>
15         <td>Linha1-1</td>
16         <td>Linha1-1</td>
17     </tr>
18     <tr>
19         <td>Linha2-1</td>
20         <td>Linha2-1</td>
21         <td>Linha2-1</td>
22     </tr>
23 </table>
24
25 <p>Para entender melhor o exemplo, foram
    adicionadas bordas     tabela.</p>
26
27 </body>
28 </html>
```

Listing 4.3: Exemplo declaração de linha da tabela

Veja na figura 4.3 o resultado do código:

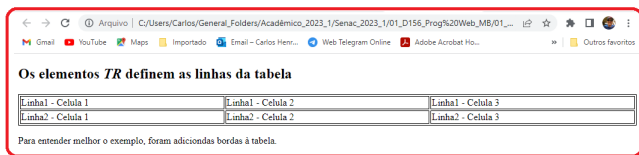


Figure 4.3: Estrutura de página HTML

4.2 Listas

As listas HTML permitem que os desenvolvedores da Web agrupem um conjunto de itens relacionados em listas.

4.2.1 Lista HTML não ordenada

Uma lista não ordenada começa com a tag ``. Cada item da lista começa com a tag ``.

Para representar a cada item da lista é renderizado um marcador (pequenos círculos pretos) por padrão, conforme demonstrado no exemplo abaixo:

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h2>Uma lista HTML não ordenada</h2>
6
7 <ul>
8   <li>Contas a pagar</li>
9   <li>Gastos gerais</li>
10  <li>Manutenção</li>
11  <li>Recursos Humanos</li>
12 </ul>
13
14 </body>
15 </html>
```

Listing 4.4: Exemplo Lista não ordenada

Veja na figura 4.4 o resultado do código:



Figure 4.4: Estrutura de página HTML

4.2.2 Lista HTML ordenada

Uma lista ordenada começa com a tag ``. Cada item da lista começa com a tag ``.

Os itens da lista serão marcados com números por padrão:

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h2>Uma lista HTML ordenada</h2>
6
7 <ol>
8     <li>Contas a pagar</li>
9     <li>Gastos gerais</li>
10    <li>Manuten ok </li>
11    <li>Recursos Humanos</li>
12 </ol>
13
14 </body>
15 </html>
```

Listing 4.5: Exemplo Lista nao ordenada

Veja na figura 4.4 o resultado do código:

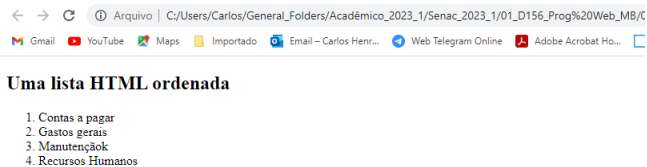


Figure 4.5: Lista Ordenada

4.3 Classes

O atributo de classe HTML é usado para especificar uma classe para um elemento HTML. Vários elementos HTML podem compartilhar a mesma classe.

4.3.1 Usando o atributo de classe

O atributo **class** geralmente é usado para apontar para um nome de classe em uma folha de estilo **CSS**. Também pode ser usado por um **JavaScript** para acessar e manipular elementos com o nome de classe específico.

No exemplo a seguir, temos três elementos `<div>` com um atributo de classe com o valor "cidade". Todos os três elementos `<div>` serão estilizados igualmente de acordo com a definição de estilo **.cidade** na seção `head`, conforme demonstrado na listagem:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <style>
5  .cidade {
6      background-color: tomato;
7      color: white;
8      border: 2px solid black;
9      margin: 20px;
10     padding: 20px;
11 }
12 </style>
13 </head>
14 <body>
15
16 <div class="cidade">
17 <h2>Muzambinho</h2>
18 <p>    um munic pio brasileiro localizado no
        estado de Minas Gerais. Sua popula o
```



```
    estimada em 2017 era de 21.026 habitantes
    .</p>
19 </div>
20
21 <div class="cidade">
22 <h2>Paudalho</h2>
23 <p>    um munic pio brasileiro do estado de
    Pernambuco.Popula o estimada [2021]
24    57.346 habitantes </p>
25 </div>
26
27 <div class="cidade">
28 <h2>Parna ba</h2>
29 <p>    um munic pio brasileiro do Estado do
    Piau , o segundo mais populoso do estado,
    possuindo uma popula o estimada em 183
    482 habitantes em julho de 2020</p>
30 </div>
31
32 </body>
33 </html>
```

Listing 4.6: Exemplo Lista nao ordenada

Veja na figura 4.6 o resultado do código:

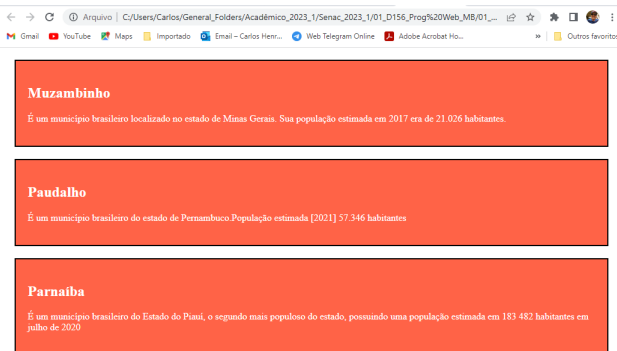


Figure 4.6: Exemplo de classe

A sintaxe da classe

Para criar uma classe; escreva um caractere de ponto (.), seguido de um nome de classe. Em seguida, defina as propriedades CSS entre chaves .

Uso do atributo class em JavaScript

O nome da classe também pode ser usado pelo **JavaScript** para executar determinadas tarefas para elementos específicos.

O **JavaScript** pode acessar elementos com um nome de classe específico com o método **getElementsByClassName()**, conforme demonstrado no código abaixo:

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h2>Uso do atributo class em JavaScript</h2>
6 <p>Clique no bot o para ocultar todos os
   elementos com o nome de classe "cidade":</p>
7
8 <button onclick="myFunction()">Ocultar
   elementos</button>
9
10 <h2 class="cidade">Muzambinho</h2>
11 <p> um munic pio brasileiro localizado no
   estado de Minas Gerais. Sua popula o
   estimada em 2017 era de 21.026 habitantes
   .</p>
12
13 <h2 class="cidade">Paudalho </h2>
14 <p> um munic pio brasileiro do estado de
   Pernambuco. Popula o estimada [2021]
   57.346 habitantes </p>
15
16
17 <h2 class="cidade">Parna ba </h2>
```

```
18      <p>    um munic pio brasileiro do Estado
          do Piau , o segundo mais populoso do
          estado, possuindo uma popula o
          estimada em 183 482 habitantes em
          julho de 2020</p>
19
20 <script>
21 function myFunction() {
22     var x = document.getElementsByClassName("
          cidade");
23     for (var i = 0; i < x.length; i++) {
24         x[i].style.display = "none";
25     }
26 }
27 </script>
28
29 </body>
30 </html>
```

Listing 4.7: Exemplo uso de classe no JavaScript

Veja na figura 4.7 o resultado do código:

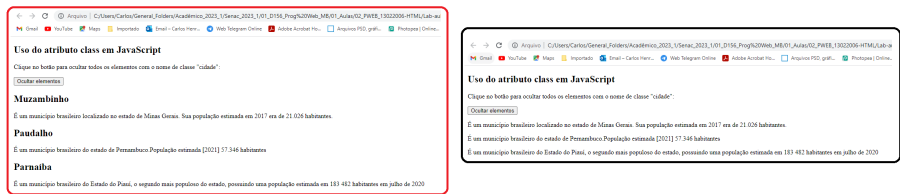


Figure 4.7: Exemplo de classe com JavaScript

Chapter 5

Formulários

Um formulário HTML é usado para coletar a entrada do usuário. A entrada do usuário geralmente é enviada a um servidor para processamento. Veja o exemplo abaixo:

```
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <h2>HTML Forms</h2>
6
7  <form action="/acao_da_pagina.php">
8    <label for="fname">Nome:</label><br>
9    <input type="text" id="fname" name="fname"
      value="Joaqui Jose"><br>
10   <label for="lname">Sobrenome:</label><br>
11   <input type="text" id="lname" name="lname"
      value="Da Silva Xavier"><br><br>
12   <input type="submit" value="Envia">
13 </form>
14
15 <p>Se voc  clicar no bot o "Enviar", os
      dados do formu rio ser o enviados para
      uma p gina chamada "/acao_da_pagina.php"
      .</p>
```

16

```
17 </body>  
18 </html>
```

Listing 5.1: Exemplo Formulário

Veja na figura 5.1 o resultado do código:

← → ⓘ Arquivo | C:/Users/Carlos/General_Folders/Acadêmico_2023_1/Senac_2023_1/01_D156_Prog%20Web_M8/01_A

Gmail YouTube Maps Importado Email - Carlos Henr... Web Telegram Online Adobe Acrobat Ho... An

HTML Forms

Nome:
Joaqui Jose

Sobrenome:
Da Silva Xavier

Envia

Se você clicar no botão "Enviar", os dados do formulário serão enviados para uma página chamada "/acao_da-pagina.php".

Figure 5.1: Exemplo de formulário

Chapter 6

JavaScript

O JavaScript torna as páginas HTML mais dinâmicas e interativas. Veja o exemplo abaixo:

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h1>Ola Mundo:JavaScript</h1>
6
7 <button type="button"
8 onclick="document.getElementById('demo').
   innerHTML = Date()">
9 Clique em mim para exibir Data e Hora.</
   button>
10
11 <p id="demo"></p>
12
13 </body>
14 </html>
```

Listing 6.1: Exemplo HTML/JavaScript

Veja na figura 6.1 o resultado do código:

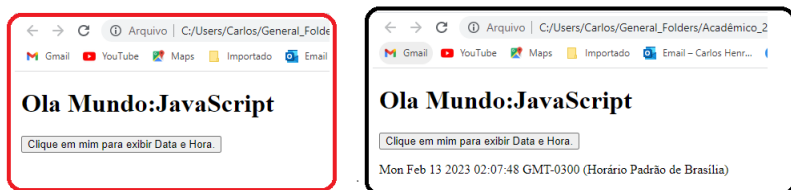


Figure 6.1: Exemplo HTML/JavaScript

6.1 A tag HTML `<script>`

A tag HTML `<script>` é usada para definir um script do lado do **cliente (JavaScript)**.

O elemento `<script>` contém instruções de script ou aponta para um arquivo de script externo por meio do **atributo src**.

```

1  <!--
2      SENAC - TADS - Programa o Web
3      Aula #02 - Introdu o ao JavaScript
4      Objetivos deste código: Demonstrar JS em
5          P ginas html
6          "Intera o com Usu rio - Fazendo
7              C lculos"
8  -->
9
10 <!DOCTYPE html>
11 <html lang="pt-br">
12 <head>
13     <meta charset="UTF-8">
14     <meta name="viewport" content="width=
        device-width, initial-scale=1.0">
        <title>N meros com JS</title>

```



```
15     <style>
16         body { font: 12pt Arial; }
17         button { font-size: 12pt; padding: 30
18             px; }
19     </style>
20 </head>
21 <body>
22     <h1>Senac - TADS - PW - 2     Semestre </
23     h1>
24     <h2>Aula #02 - Introdu o ao JS</h2>
25     <h3>Fazendo C lculos:Soma de N meros</h3>
26     <button onclick="somarNumeros()">Iniciar
27     a soma</button>
28     <section id="res">
29         <p>0 resultado aparecer aqui...</p>
30     </section>
31     <script src="arquivo.js"></script>
32 </body>
</html>
```

Listing 6.2: Exemplo HTML/javascript

Abaixo temos o conteúdo do arquivo apontado na *SRC* :

arquivo.js, mencionado no código acima.

```
1 function somarNumeros() {
2     let n1 = Number(window.prompt('Por favor,
3         Digite um n mero: '))
4     let n2 = Number(window.prompt('Por favor,
5         Digite outro n mero:'))
6     soma = n1 + n2
7     let res = document.querySelector('section
8         #res')
9     res.innerHTML = '<p>A soma entre <mark>${
10         n1}</mark> e <mark>${n2}</mark>
11         igual a <strong>${soma}</strong>!</p>'
```

Listing 6.3: Conteúdo do **arquivo.js**