

Web Academy

Fundamentos de Programação Front-end



Manoel Limeira de Lima Júnior



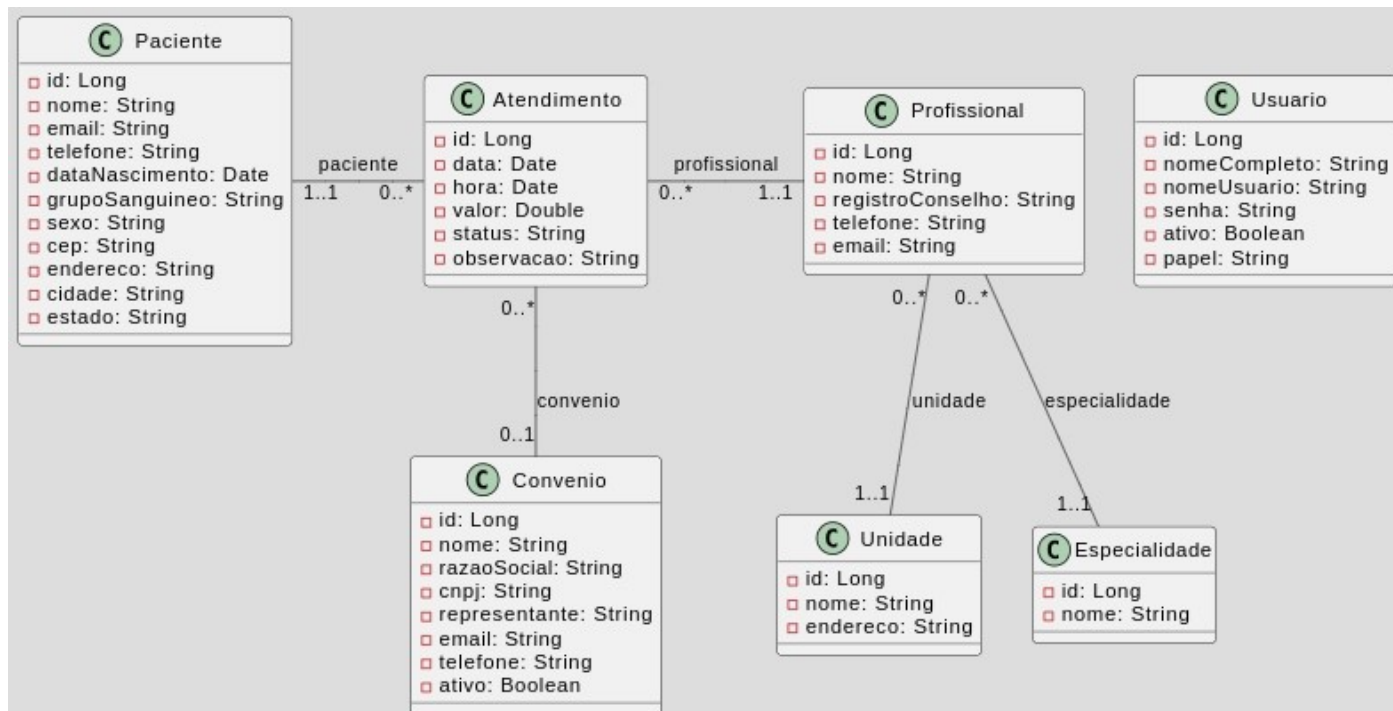
Web Academy



Apresentação

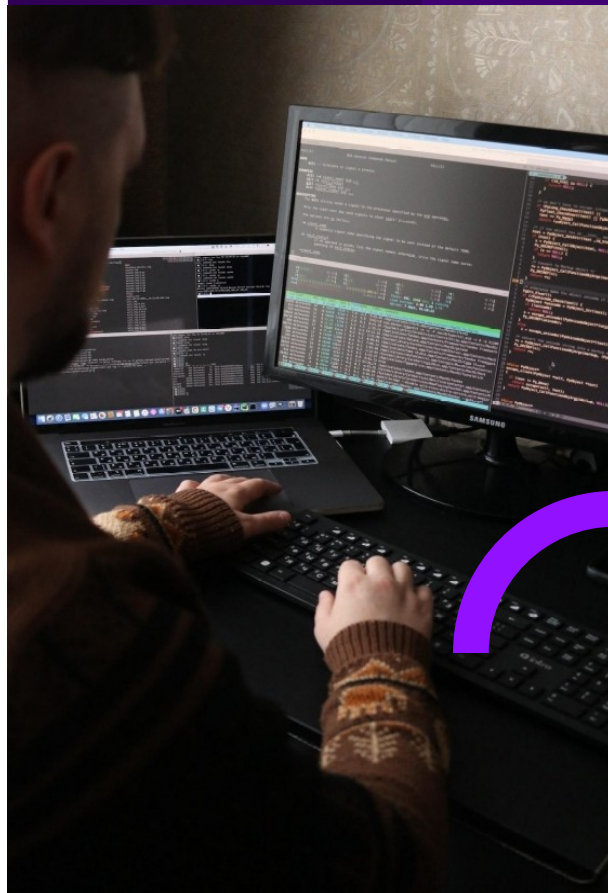
SGCM – Sistema de Gerenciamento de Consultas Médicas

- Documentação: <https://github.com/webacademyufac/sgcmdocs>
 - Diagrama de classes



Ementa

1. Fundamentos de **HMTL**.
2. Padrões e recomendações da **W3C**, semântica e acessibilidade.
3. Estilização de páginas HTML com **CSS**.
4. Técnicas de design responsivo.
5. Tipos de dados, funções, objetos, arrays e manipulação de eventos em **JavaScript**.
6. Manipulação de **DOM** (Document Object Model).
7. **JSON** (JavaScript Object Notation).
8. Requisições assíncronas.



Objetivos

- **Geral**

- Capacitar o aluno na utilização de **procedimentos e técnicas básicas** de desenvolvimento de aplicações para a WEB, com **ênfase nos fundamentos** de tecnologias voltadas ao desenvolvimento **front-end**.

- **Específicos:**

- Apresentar os principais conceitos de linguagens, protocolos e ferramentas que dão suporte ao funcionamento da Web;
- Compreender a importância dos padrões Web na produção de códigos válidos, semanticamente corretos e acessíveis;
- Capacitar o aluno no emprego correto dos recursos disponíveis nas tecnologias HTML, CSS e JavaScript, para construção de aplicações Web, separando conteúdo, apresentação e interatividade.

Conteúdo programático

Introdução

O lado cliente (front-end) e o lado servidor (back-end);
O protocolo HTTP, HTML e a Web;
Evolução do HTML;
Tecnologias de front-end; Padrões web, acessibilidade e design responsivo.

HTML

Introdução ao HTML;
Estrutura de um documento HTML;
Principais elementos (tags).

CSS

Introdução ao CSS;
Bordas e margens (box model); Sintaxe e seletores;
Herança; Aplicação de CSS: cores, medidas, textos e layout.

JavaScript

Introdução ao JavaScript; Sintaxe; Principais tipos de dados; Objetos e Arrays; Formas de utilização; Eventos; DOM; JSON; Requisições assíncronas (AJAX).

Bibliografia



Web

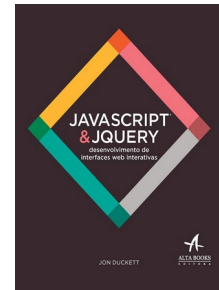


HTML e CSS: projete e construa websites.

Jon Duckett

1a Edição – 2016

Editora Alta Books



JavaScript e JQuery: desenvolvimento de interfaces web interativas.

Jon Duckett

1a Edição – 2016

Editora Alta Books

Academy

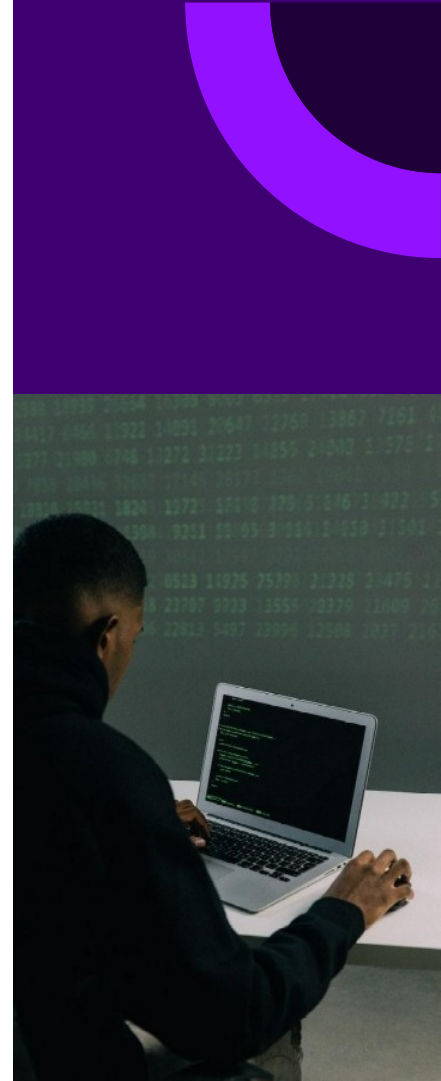
Sites de referência

- **MDN Web Docs: Aprendendo desenvolvimento web.**
 - <https://developer.mozilla.org/pt-BR/docs/Learn>
- **W3Schools Online Web Tutorials.**
 - <https://www.w3schools.com/>
- **W3C Standards.**
 - <https://www.w3.org/standards/>



Ferramentas

- **Visual Studio Code**
 - <https://code.visualstudio.com/Download>
- **Live Server (Extensão do VS Code)**
 - <https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer>
- **Git**
 - <https://git-scm.com/downloads>
- **Chrome Developer Tools (F12)**





Web Academy



Introdução

O protocolo HTTP, HTML e a Web (WWW)

- Nos anos **1980**, **Tim Berners-Lee**, Físico do CERN, trabalhava no projeto Enquire, que tinha como um dos objetivos criar o que ficou conhecido como **hipertexto**.
- O Hipertexto relaciona textos, imagens, sons, vídeos e qualquer tipo de conteúdo multimídia.
- Tim também criou o **HTTP** (*HyperText Transfer Protocol*), os **URLs** (*Uniform Resource Locators*), que são a base da web até hoje, o primeiro navegador **Web** (*WorldWideWeb*, mais tarde renomeado Nexus) e o primeiro servidor web.



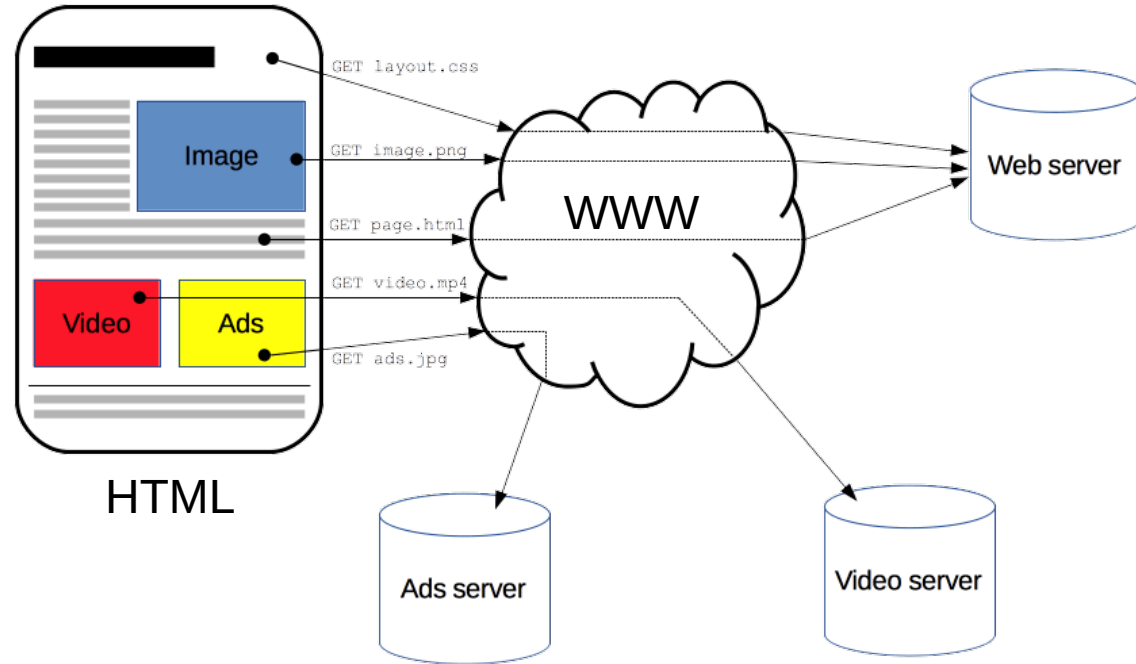


O protocolo HTTP, HTML e a Web (WWW)

- Com base no **TCP/IP** (*Transmission Control Protocol/Internet Protocol*) surgiu a ideia de transmitir o conteúdo hipertexto pela rede.
- Para isso foi criado o protocolo **HTTP** (*Hypertext Transfer Protocol*).
- Uma linguagem para criação de conteúdo hipertexto foi criada, o **HTML** (*HyperText Markup Language*).
- E além disso foi criado o conceito **WWW** (*World Wide Web*) que engloba todos os serviços de conteúdo multimídia baseados no protocolo HTTP.

O protocolo HTTP, HTML e a Web (WWW)

- HTTP é um **protocolo cliente-servidor** que permite a obtenção de recursos, como documentos HTML.
- Clientes e servidores se comunicam trocando mensagens enviadas pelo cliente, geralmente um navegador da Web, são chamadas de requisições ou (**requests**) e as mensagens enviadas pelo servidor são chamadas de respostas (**responses**).



O protocolo HTTP, HTML e a Web (WWW)

- Primeiro site criado com a linguagem HTML pra funcionar sob o protocolo HTTP:
 - <http://info.cern.ch/>
- O número exato de sites muda a cada segundo, estima-se 1,2 bilhão de sites na internet em fevereiro de 2025, cerca de 17% estão ativos
 - <https://siteefy.com/how-many-websites-are-there/>

World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#), [Policy](#), November's [W3 news](#), [Frequently Asked Questions](#).

[What's out there?](#)

Pointers to the world's online information, [subjects](#), [W3 servers](#), etc.

[Help](#)

on the browser you are using

[Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#), X11 [Viola](#), [NeXTStep](#), [Servers](#), [Tools](#), [Mail robot](#), [Library](#).)

[Technical](#)

Details of protocols, formats, program internals etc

[Bibliography](#)

Paper documentation on W3 and references.

[People](#)

A list of some people involved in the project.

[History](#)

A summary of the history of the project.

[How can I help ?](#)

If you would like to support the web..

[Getting code](#)

Getting the code by [anonymous FTP](#), etc.

Evolução da linguagem HTML

Ano	Versão	Evolução Histórica
1991	HTML 1.0	Primeira versão publicada no CERN por Tim Berners-Lee.
1995	HTML 2.0	Desenvolvimento (formulários, imagens embutidas e tabelas) se deu em colaboração com várias empresas e fabricantes de navegadores.
1997	HTML 3.2	Sob responsabilidade do W3C, trazendo padronização e incluindo suporte para CSS e applets de Java.
1999	HTML 4.01	A HTML 4.01 publicada pelo W3C foi uma revisão da HTML 3.2, incluindo novos recursos como suporte para frames, scripts do lado do cliente (JavaScript) e melhorias na acessibilidade.
2000	XHTML	O W3C criou a linguagem XHTML versão 1.0 (baseada na tecnologia XML) a partir da HTML versão 4 e propôs acabar com a linguagem HTML.
2004		Discutiu-se a evolução HTML 5, proposta apresentada pela Mozilla e Opera, e foi rejeitada pela W3C que havia optado por evoluir a linguagem XHTML.

Evolução da linguagem HTML (cont.)

Ano	Versão	Evolução Histórica
2007	WHATWG XHTML 2.0	Contrários a decisão do W3C, Firefox, Opera e Safari criaram o WHATWG. A W3C ainda lançou, em 2007, o XHTML 2.0, mas o WHATWG já tinha uma proposta de nova versão do HTML e o W3C aceitou a proposta.
2014	HTML 5	Inclusão de novos elementos e atributos de mídia e formulário, semântica aprimorada e APIs JavaScript para interação com a web moderna. Projetada para funcionar em vários dispositivos.
2016	HTML 5.1	Novos recursos e refinamentos, como o elemento <picture> para seleção de imagem responsiva e o elemento <dialog> para criar caixas de diálogo modais.
2017	HTML 5.2	Novos recursos como o elemento <main> para identificar o conteúdo principal de uma página e o elemento <details> para criar caixas de detalhes que podem ser expandidas ou recolhidas pelo usuário
2020	HTML 5.3	Novos elementos e recursos, como o elemento <slot> para ajudar na construção de componentes da web reutilizáveis

História e evolução da Web

- **Web 1.0 (1991 - 2004)**

- Sites estáticos, conteúdo fixo, HTML básico (*hiperlinks*), design semelhante ao de páginas impressas. Vídeos eram raros. Os internautas apenas consumiam o que estava disponível. Exceções: envios de e-mail e formulários de cadastro. Exemplos: sites de universidades, órgãos governamentais e empresas.

- **Web 2.0 (2004 - ...)**

- Sites dinâmicos, interação do usuário, a produção de conteúdo ganhou espaço possibilitando a inserção em páginas de blogs e redes sociais. Popularização da tecnologia AJAX – *Asynchronous Javascript and XML* (2005) e o uso dos estilos em CSS – *Cascading Style Sheet*. Exemplos: Wikipedia, YouTube, Facebook.

- **Web 3.0 (2014 - ...)**

- Automação, Web semântica, maior personalização e descentralização por meio de *Blockchain* e algoritmos para melhorar a eficiência de atividades com o uso de *Machine Learning*.

- **Web 4.0 (atualmente)**

- Cuidados com políticas de privacidade, Internet das coisas (IoT), forte tendência na integração de serviços com uso de Inteligência Artificial

Fonte: <https://br.hubspot.com/blog/marketing/evolucao-web>

Evolução da linguagem HTML

- O projeto do **HTML 5** com o apoio do W3C teve início em 2008.
- A nova versão trazia pela primeira vez a separação total entre **semântica**, **estilo** e **interatividade**.



Tecnologias relacionadas a sistemas web

- A estrutura de uma página web é baseada atualmente em 3 tecnologias principais. Além do **HTML**, são elas:
 - **CSS** (*Cascading Style Sheets*): linguagem que define o layout de documentos HTML;
 - **JavaScript**: linguagem de programação que roda no lado cliente (navegador).



Wayback Machine

- Banco de dados digital mantido pelo Internet Archive com bilhões de páginas de internet;
- Permite visualizar versões antigas de páginas web;
- <https://archive.org/web/>



Wayback Machine

- Globo.com em 2000



Wayback Machine

- UFAC em 2000

The screenshot shows the homepage of the Universidade Federal do Acre (UFAC) in 2000. The layout includes a header with the university's name and a banner image. A left sidebar contains links to various university services. The main content area is divided into sections for announcements, news, and downloads. A right sidebar features a 'Veja Mais' section with links to academic calendar, email, projects, photos, and national events. The footer contains contact information, logos for BNE and Fundação BIOMA, and a copyright notice.

UFAC
Universidade Federal do Acre

Links Úteis
Página Inicial dos Links
Órgãos do Governo
Universidades Brasileiras
Busca na Web

Downloads
Diretório do Grupo de Pesquisas no Brasil - versão 4.0 /CNPq

Veja Mais »»»
» Calendário Acadêmico
» Email's Úteis
» Projetos na UFAC
» Fotos da UFAC
» Eventos Nacionais

PRODUC
Produção de Docentes

POP-AC

BNE
Banco Nacional de Empresas
WWW.BNE.COM.BR

Fundação BIOMA

Universidade Federal do Acre - UFAC
BR 364 Km 04 Cep: 69915-900 Bairro Distrito Industrial
Caixa Postal 500 - Rio Branco - Acre
PABX: (0xx68) 229-2244
Design by webmaster@ufac.br
(c) 2000 - Todos os direitos reservados.

Melhor Visualizado com 800x600
Desde 23/06/2000
Atualizado em 25/09/2000.

O que são os padrões web?

- Os **padrões web** (*web standards*) são amplamente discutidos e empregados por desenvolvedores e pessoas envolvidas com o desenvolvimento de aplicações para web.
- São **recomendações** (e não normas!) destinadas a orientar os desenvolvedores para o uso de boas práticas de construção de páginas web que tornam o conteúdo acessível para todos.
- São essenciais para a construção de uma **web aberta, acessível e interoperável**, garantindo que todos possam desfrutar de uma experiência de qualidade.

O que são os padrões web?

- Apesar de existirem **órgãos normatizadores**, como o **ISO** (International Organization for Standardization) e **ECMA** (European Computer Manufacturers Association), normalmente quando discutimos padrões web nos referimos aos padrões do **W3C** (World Wide Web Consortium).
- Uma **recomendação** do W3C é uma especificação ou um conjunto de diretrizes que passou por discussão e foi estabelecido um consenso, passando a ser indicado seu amplo emprego.

Padrões Web

- O trabalho do W3C é abrangente e alcança diversas tecnologias.
- Essa abrangência pode ser agrupada em três segmentos:
 - Código válido;
 - Código semanticamente correto;
 - Separação entre **conteúdo** (HTML), **apresentação** (CSS) e **interatividade** (JavaScript).

Benefícios na adoção de padrões web

- Melhor indexação pelos mecanismos de busca;
- Renderização mais rápida;
- Garantia de funcionamento completo da página;
- Páginas com melhor aspecto de apresentação;
- Comportamento uniforme entre diferentes navegadores de internet.



Acessibilidade na Web

- **Acessibilidade** significa permitir que o maior número de pessoas possam usar a web, independente da sua limitação.
- **Restrições no acesso a web** é um problema que afeta muitas pessoas que possuem algum tipo de necessidade especial.
- Ainda existem muitas páginas com barreiras de acessibilidade que dificultam ou mesmo tornam impossível o acesso.



Acessibilidade na Web: exemplos de barreiras

- Imagens que não possuem texto alternativo.
- Formulários que não podem ser navegados em uma sequência lógica ou que não estão rotulados.
- Páginas com tamanhos de fontes absoluta, que não podem ser aumentadas ou reduzidas facilmente.
- Páginas que, devido ao layout inconsistente, são difíceis de navegar quando ampliadas por causa da perda do conteúdo adjacente.

Padrões web e acessibilidade

- Os padrões web representam o básico para uma página web acessível.
- É também importante acrescentar aos padrões web as técnicas de acessibilidade associadas ao **WCAG** (*Web Content Accessibility Guidelines*) e suas recomendações.
- As diretrizes WCAG abrangem um vasto conjunto de recomendações que têm como objetivo **tornar o conteúdo Web mais acessível**.

Design responsivo

- O conceito de Design Responsivo surgiu quando em 2010 um desenvolvedor chamado Ethan Marcotte, criou o artigo Responsive Web Design, para o blog A List Apart.
- Um conjunto de práticas que permite criar uma aplicação web com **conteúdo acessível**, otimizando a experiência do usuário, respeitando as limitações, **independente do dispositivo que está sendo utilizado**.
- Não se trata de criar uma versão para cada tipo de dispositivo.



Design responsivo

- Vantagens:
 - Rapidez nas manutenções, pois não precisa atualizar em versões diferentes do site;
 - Melhor posicionamento nos motores de busca em relação a não responsivos;
 - Consistência na experiência do usuário porque os visitantes recebem a mesma experiência mesmo acessando dispositivos.
- Desvantagens:
 - Otimização para celulares, a velocidade de carregamento é menor em comparação a outros dispositivos;
 - Dispositivos menores podem causar dificuldades para projetar navegações mais complexas;
 - O tempo gasto em um projeto usando Design Responsivo é maior do que a execução de um site para monitores 'desktop'.



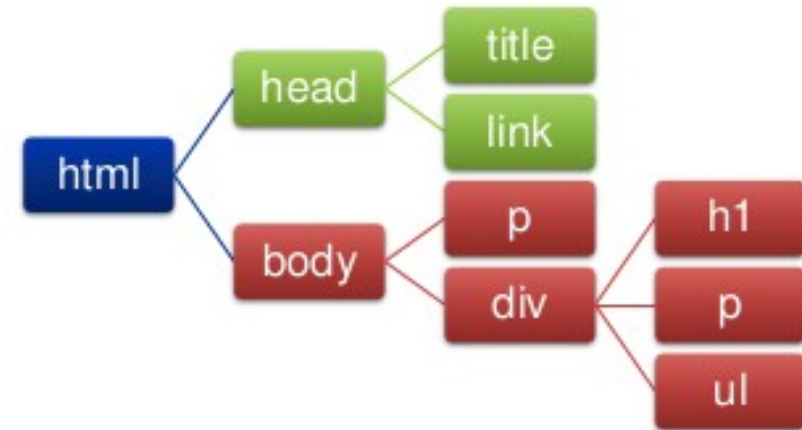
Web Academy



HTML (HyperText Markup Language)

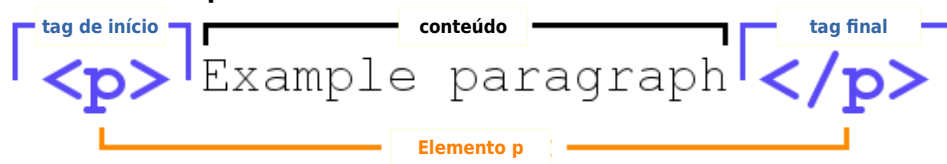
Introdução a HTML

- HTML é uma linguagem interpretada pelo navegador para exibir conteúdo.
- Nossa referência é o HTML 5.
- O documento HTML é composto por elementos hierarquicamente organizados.



Tags (elementos)

- Para inserir um elemento em um documento HTML, utilizamos **tags** correspondentes a esse elemento.



- As tags não diferenciam maiúsculas de minúsculas: `<BODY>` significa o mesmo que `<body>`
- O W3C recomenda letras minúsculas e exige letras minúsculas para tipos de documentos mais rígidos, como XHTML.

Exemplos de tags

```
<html> </html>
<head> </head>
<script> </script>
<title> </title>
<table> </table>
<body> </body>
<p> </p>
<h1> </h1>
<br>
```


Tags (elementos)

- Alguns elementos HTML são classificados como **normal elements**, são abertos com uma tag e fechados com uma barra seguida da mesma tag.
- Exemplo:
- Há também os chamados **void elements**, que não possuem conteúdo, sendo abertos e fechados com apenas uma tag.
- Opcionalmente podem conter uma barra no final da tag.
- Exemplo:

```
<h1>WEB ACADEMY</h1>
```

```

```

Estrutura básica de uma página HTML

- Um documento HTML válido precisa obrigatoriamente seguir uma estrutura básica.
- O primeiro elemento não é um tag, mas sim uma instrução que indica para o navegador a versão HTML.
- Para ver o código HTML da página digite **CTRL + U** ou com o botão direito e selecione “Exibir código-fonte da página”.
- Com o botão direito em um elemento (ou em uma área em branco) é possível **"Inspeccionar"** para ver como os elementos são compostos.
- O **Markup Validation Service** examina e fornece um relatório para informar o que há de errado com seu HTML (<https://validator.w3.org/>).

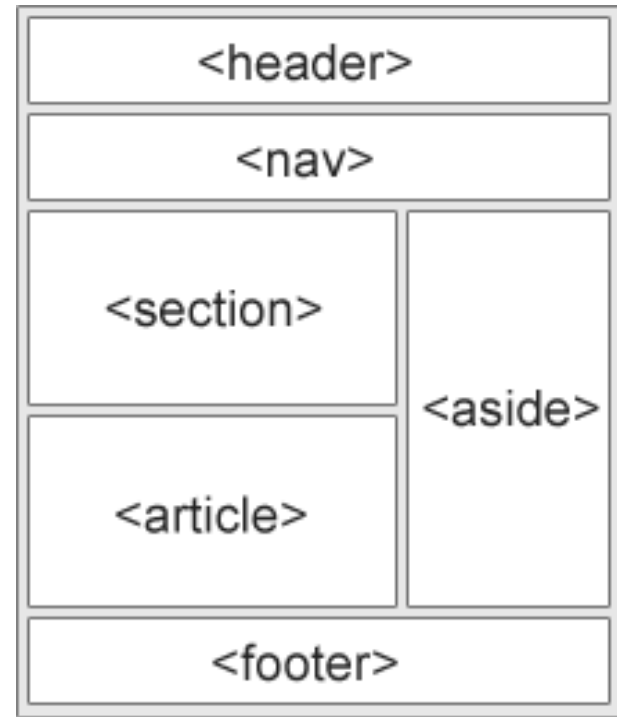
```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="utf-8"/>
  <title>Título</title>
</head>
<body>
  <p>Conteúdo</p>
</body>
</html>
```

Doctype

- Para cada tipo de documento existe uma instrução doctype específica.
- Exemplos:
 - HTML 5: `<!DOCTYPE html>`
 - HTML 4.01 Strict: `<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">`
 - XHTML 1.0 Strict: `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`

Estrutura e Layout

- **<header>** Define o cabeçalho da página ou seção.
- **<nav>** contém a principal funcionalidade de navegação da página.
- **<section>** define uma seção que agrupa um conteúdo.
- **<article>** conteúdo relacionado que faz sentido por si só.
- **<main>** Define a seção principal da página.
- **<aside>** Define o conteúdo lateral que não está diretamente relacionado ao conteúdo principal.
- **<footer>** Define rodapé da página ou seção.



Títulos do conteúdo

- Quando se quer indicar que um texto é um título deve-se utilizar as tags de título (heading).
- São tags de conteúdo que vão de **<h1>** até **<h6>**, sendo **<h1>** o título principal e mais importante, e **<h6>** o título de menor relevância.

```
<h1>Título</h1>
```

```
<h2>Título</h2>
```

```
<h3>Título</h3>
```

```
<h4>Título</h4>
```

```
<h5>Título</h5>
```

```
<h6>Título</h6>
```

Títulos do conteúdo

- A ordem de importância, além de influenciar no tamanho padrão de exibição do texto, tem **impacto nas ferramentas que processam HTML**, como as ferramentas de indexação de conteúdo para buscas (Google, Bing, etc).
- Além disso, os navegadores especiais para acessibilidade também interpretam o conteúdo dessas tags de maneira a **diferenciar seu conteúdo e facilitar a navegação do usuário** pelo documento.

Parágrafos

- Para exibir qualquer texto em uma página, é recomendado que ele esteja dentro de uma tag filha da tag **<body>**, sendo a marcação mais indicada para textos comuns a tag de parágrafo: **<p>**.

- Exemplo:

```
<p>Primeiro parágrafo.</p>
```

```
<p>Segundo parágrafo.</p>
```

- Os navegadores ajustam os textos dos parágrafos à largura do elemento pai, inserindo as quebras de linha necessárias automaticamente.

Caracteres especiais

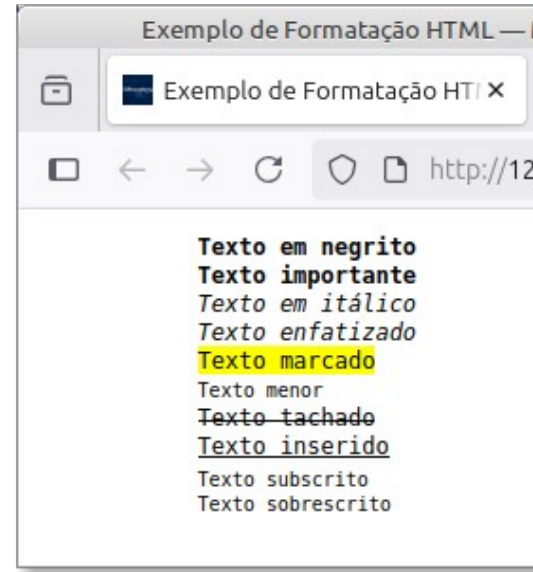
- Em HTML, os caracteres <, >, ", ' e & são especiais. Eles fazem parte da própria sintaxe HTML. Para incluir esses caracteres especiais deve-se inserir um E comercial (&) seguido da referência com um ponto e vírgula (;) no final.

Caractere	Literal
<	<
>	>
"	"
'	'
&	&

Formatação e Comentários

- Os elementos de formatação foram projetados para exibir tipos especiais de texto.
- Para escrever um comentário HTML, coloque-o nos marcadores especiais `<!--` e `-->`.
- Os comentários não são exibidos pelo navegador, mas ajudam a documentar o código HTML.

```
<pre>
<b>Texto em negrito</b>
<strong>Texto importante</strong>
<i>Texto em itálico</i>
<em>Texto enfatizado</em>
<mark>Texto marcado</mark>
<small>Texto menor</small>
<del>Texto tachado</del>
<ins>Texto inserido</ins>
<sub>Texto subscrito</sub>
<sup>Texto sobrescrito</sup>
<!-- Comentários -->
</pre>
```



Elementos genéricos (de agrupamento)

- **<div>** e **** são elementos genéricos que não representam um conteúdo específico, mas são úteis para agrupar conteúdos (ou elementos) que compartilham atributos de estilo.
- Devem ser utilizados apenas quando não existirem outros elementos para representar o conteúdo.
- Diferença: **<div>** é um elemento de nível de bloco (agrupar blocos) e **** de nível de linha (agrupar texto).

```
<!DOCTYPE html>
<html>
<body>
  <h1>Título</h1>
  <div>
    <h2>Exemplo</h2>
    <p>Dentro do elemento DIV</p>
  </div>
  <p>Parágrafo fora do elemento DIV
```

que contém um elemento

```
<span>SPAN</span>.
```

```
</p>
</body>
</html>
```

Listas

- Para criar listas em HTML são utilizadas as tags:
 - **** cria listas não ordenadas;
 - **** cria listas ordenadas;
 - **** cria itens nas listas.
- As listas podem ser aninhadas (lista dentro da lista)
- O atributo **type** das tags de criação define o tipo de marcador do item. Possíveis valores não ordenados: “disc”, “square”, “circle” e ordenados: “1”, “A”, “a”, “I”, “i”

```
<h4>Lista não  
ordenada:</h4>  
<ul>  
  <li>Item A</li>  
  <li>Item B</li>  
  <li>Item C</li>  
</ul>  
<h4>Lista ordenada:</h4>  
<ol>  
  <li>Item A</li>  
  <li>Item B</li>  
  <li>Item C</li>  
</ol>
```

Lista não ordenada:

- Item A
- Item B
- Item C

Lista ordenada:

1. Item A
2. Item B
3. Item C

Imagens

- A tag **** insere uma imagem e possui dois atributos obrigatórios:
 - **src**: indica o URL (*Uniform Resource Locator*), ou seja, o caminho do arquivo.
 - **alt**: define um texto alternativo caso a imagem não seja carregada.
- Em HTML 5:
 - **<figure>**: especifica conteúdo como ilustrações, diagramas, fotos, etc.
 - **<figcaption>**: define uma legenda.
- Atributos de tamanho:
 - **width** e **height** especificam a largura e a altura da imagem (em pixels). Cuidado com a proporcionalidade.

URL absoluto: imagem hospedada em outro site.

src="www.site.com/images/foto.jpg"

URL relativo: imagem hospedada no próprio site. Se o URL começar com uma barra, será relativo ao domínio.

src="/images/foto.jpg".

```
<figure>
```

```
  
```

```
  <figcaption>
```

```
    Legenda da foto.
```

```
  </figcaption>
```

```
</figure>
```


Links

- As ligações (âncoras) entre páginas (hiperlinks ou simplesmente link) são definidas pela tag **<a>**
- O atributo **href** (referência de hipertexto) especifica o URL da página de destino.
- Os links podem ser criados sobre conteúdo de texto simples ou vários outros tipos de elementos HTML, como imagens, títulos, etc.
- O atributo **target** especifica onde abrir a página:
 - **_self**: abre na mesma janela/aba
 - **_blank**: abre em uma nova janela/aba

```
<a
href="http://www.ufac.br">UFAC</a>
<a href="http://csi.ufac.br"
target="_blank">
  <h1>SI-UFAC</h1>
</a>
<a href="http://webacademy.ufac.br"
">
  <figure>
    
  </figure>
</a>
```

Tabelas

- Uma tabela é definida não apenas por uma tag, mas pode ter até 10 tags diferentes
- Três elementos básicos:
 - **<table>**, **<tr>** e **<td>**
- Objetivo: apresentar dados tabulares, comparativos, etc. (não para posicionar elementos na página)

```
<table>
  <tr> <td></td> <td></td> <td></td> </tr>
  <tr> <td></td> <td></td> <td></td> </tr>
  <tr> <td></td> <td></td> <td></td> </tr>
  <tr> <td></td> <td></td> <td></td> </tr>
</table>
```

Tabelas

Tag	Descrição
<table>	Define uma tabela
<tr>	Insere uma linha na tabela
<td>	Insere uma célula dentro de um elemento <tr>
<th>	Insere uma célula (cabeçalho) dentro de um elemento <tr>
<caption>	Atribui um título ou descrição para a tabela
<colgroup>	Especifica um grupo de colunas para formatação
<col>	Define propriedades da coluna para cada elemento dentro do <colgroup>
<thead>	Define o cabeçalho da tabela
<tbody>	Define o corpo (conteúdo principal) da tabela
<tfoot>	Define o rodapé da tabela

```

<table>
  <caption>Alunos</caption>
  <thead>
    <tr>
      <th>Nome</th>
      <th>Nota</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Aluno A</td>
      <td>9.0</td>
    </tr>
    <tr>
      <td>Aluno B</td>
      <td>4.5</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td colspan="2">Quantidade de alunos:
2</td>
    </tr>
  </tfoot>

```

Fundamentos de Programação Front-end

Tabelas (Exemplo)

Alunos	
Nome	Nota
Aluno A	9.0
Aluno B	4.5
Quantidade de alunos: 2	

Formulários

- Um formulário serve para enviar informações
- A tag **<form>** define, dentre outras coisas, que a página irá processar as informações
- Os tipos de campos são definidos pela tag **<input>**, e suas identificações pela tag **<label>**
- O atributo **name** identifica o campo no formulário, o **type** define o tipo do campo
- Lista de tipos de input:
https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input#input_types

O atributo **action** define a ação a ser executada quando o formulário for enviado.

```
<form action="/action_page.jsp">
  <label for="nome">Nome:</label>
  <input type="text" name="nome" id="nome"
placeholder="Digite seu primeiro nome">
  <br>
  <label for="sobrenome">Sobrenome:</label>
  <input type="text" name="sobrenome"
id="sobrenome" placeholder="Digite seu
sobrenome">
  <br>
  <input type="submit" value="Enviar">
</form>
```

Nome:

Sobrenome: