

TECNOLOGIAS WEB

INTRODUÇÃO AO HTML

Olá!

Nesta aula, iremos aprender como construir páginas Web utilizando as principais tags da linguagem mais popular para a Internet.

Veremos, ainda, como aperfeiçoar nosso código para garantir maior integração entre navegadores.

Ao final desta aula, você será capaz de:

1. Apresentar as principais tags utilizadas na construção de páginas Web.
2. Compreender as principais funcionalidades de cada tag utilizada.
3. Listar as principais tags para tratamento de imagens, textos e conteúdos.

1 HTML

A grande reviravolta na Internet aconteceu quando, em 1989, Tim Berbers-Lee desenvolveu uma linguagem capaz de publicar e organizar conteúdos na Internet. Nascia, aí, a linguagem HTML (Hypertext Markup Language) que permitiria aos pesquisadores do laboratório CERN publicarem suas informações de maneira rápida, padronizada e fácil. Esta facilidade ganhou atenção mundial e tornou-se o maior hit da Internet até hoje. Já são milhões de sites e aplicações baseadas nesta linguagem que vem explorando, a cada dia, mais e mais recursos, trazendo inovação e informação para qualquer pessoa em qualquer parte do mundo onde haja uma conexão com a Internet.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>

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

A linguagem HTML é a linguagem padrão do seu navegador. Não importa se você desenvolveu uma aplicação para Web em Java, pois o servidor sempre converterá em HTML qualquer código para que o seu navegador entenda.

As 3 principais facilidades da linguagem HTML são:

- Princípio da não linearidade da informação, permitindo ao usuário navegar pelas informações utilizando o princípio do hipertexto.
- Permite a formatação do texto, utilizando marcações para estilizá-lo, podendo ser interpretável por todo e qualquer navegador Web desde que sejam respeitadas as marcações padrão do W3C.
- Flexibilidade de uso. Esta linguagem permite fácil adequação aos mais diferentes propósitos.

Saiba mais



Acompanhe os detalhes da história da linguagem HTML: www.w3.org/People/Raggett/book4/ch02.html.

2 Estrutura

Todos os documentos HTML são chamados de páginas e contém elementos dentro de padrões estruturais.

Cada elemento é chamado de tag. Sua sintaxe permite, ainda, a definição de parâmetros o que foi fundamental para o avanço da linguagem.

Cada elemento, quando utilizado, pode trabalhar em par com a abertura e o fechamento desta tag. Neste caso, fecha-se a tag utilizando o caracter \ dentro dos sinais de <> da mesma.

Acompanhe, na descrição sintática, estas opções.

Sintaxe:

<tag> </tag> - abertura e fechamento ou para tags sem elementos adicionais **<tag>**

Uma tag é formada por:

• Comandos	
• Atributos	modificam os resultados da mesma.
• Valores	caracterizam essa mudança.

Toda as tags devem ser descritas dentro da estrutura maior de uma página HTML, que nada mais é que um arquivo texto, cujos elementos estruturais são também descritos por tags como, por exemplo:

```
<HTML><BODY></BODY></HTML>
<HR color="red">
```

No qual:

HR = Este comando desenhará uma barra horizontal na tela do seu navegador

color = atributo que especifica a cor da barra

red = Especifica que a cor do elemento é vermelha

Cada comando tem seus atributos possíveis e seus valores.

Um exemplo é o atributo size que pode ser usado com o comando FONT, com o HR, mas que não pode ser usado com o comando BODY.

Isso quer dizer que devemos saber exatamente quais os atributos e valores possíveis para cada comando.

De uma maneira geral, o HTML é um poderoso recurso, sendo uma linguagem de marcação muito simples e acessível, voltada para a produção e compartilhamento de documentos e imagens.

3 Documento HTML

Os documentos em HTML são arquivos de texto simples que podem ser criados e editados em qualquer editor de textos comum.

Você poderá salvá-lo com a extensão html ou htm.

As tags básicas de HTML que você normalmente encontrará nas páginas da Internet são:

<HTML> Esta tag irá definir o início de um documento HTML ao navegador web.

<HEAD> Nesta tag estão as definições de cabeçalho do documento. Podem ser postos aqui: chamadas Javascript, CSS, tags META entre outras.

<BODY> Em um documento HTML faz-se necessário indicar o corpo do documento. A partir desta tag até o fechamento da mesma, todos os elementos aqui descritos serão apresentados como conteúdos na página em seu navegador.

A estrutura de um documento HTML apresenta os seguintes componentes:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<HTML lang="pt-br">
  <HEAD>
    <TITLE>Título do Documento</TITLE>
  </HEAD>
  <BODY>
    Elementos
    ...
  </BODY>
</HTML>
```

4 Comandos em HTML

Dentro do cabeçalho de um documento em HTML podemos encontrar os seguintes comandos:

- <TITLE>

O conteúdo definido dentro desta tag será exibido na barra de título do seu navegador.

É compatível com os principais navegadores Web.

EXEMPLO:

```
<head>
```

```
<title>Aula de Tecnologias Web</title>
```

```
</head>
```

- <STYLE>

Quando trabalharmos com Folhas de Estilos, definimos aqui os estilos carregados.

É compatível com os principais navegadores Web.

EXEMPLO:

```
<HEAD>
```

```
<STYLE type="text/css">
```

```
p {color:red}</STYLE>
```

```
</head>
```

- **<SCRIPT>**

Utilizada para definir a carga de um script do tipo Javascript, por exemplo, do lado do cliente.

É compatível com os principais navegadores Web. No HTML 4.1 tipo de conteúdo é declarado como CDATA, que significa que as entidades não serão parseadas. Já no XHTML o tipo de conteúdo muda para (#PCDATA).

EXEMPLO1:

```
<SCRIPT type="text/javascript">
```

```
document.write("Olá Alunos do Curso!")
```

```
</SCRIPT>
```

EXEMPLO 2:

```
<SCRIPT type="text/javascript"> // <![CDATA[document.write("Olá Alunos do Curso!");]]></SCRIPT>
```

- **<LINK>**

Utilizada para definir ligações entre a página e arquivos externos como, por exemplo, arquivo de folhas de estilos.

Em XHTML esta tag deve ser obrigatoriamente fechada.

É compatível com os principais navegadores Web. Repare que esta tag é composta, também, por atributos type e href onde o primeiro indica o tipo de referência do link e o segundo a localização do arquivo em si.

Outros atributos também são importantes, a saber:

- charset - onde definiremos o conjunto de caracteres utilizado

- media – onde definiremos o tipo de mídia utilizada. Isto é importante para a acessibilidade. Acompanhe na aula sobre acessibilidade mais à frente.

EXEMPLO:

```
<head>
```

```
<link rel="estilo" type="text/css" href="meuestilo.css" />
```

</head>

- <META>

Esta tag é uma das mais utilizadas para dar aos robôs de busca informações a respeito de sua página e site. Ela define as propriedades da página. Desta forma, ferramentas de indexação de conteúdos podem, ao ler as informações contidas nestas tags, catalogar sites e suas informações, tais como descrição da página, autor, palavras chave, etc.

O atributo **content** é obrigatório, pois especifica o conteúdo da meta informação. Os demais atributos são opcionais.

Para definir o nome do autor da página devemos utilizar o atributo AUTHOR. Exemplo: <META NAME="AUTHOR" CONTENT="Flavio Brito">

Com o intuito de especificar as informações do cabeçalho da página poderemos utilizar o atributo HTTP-EQUIV. Exemplo: <META HTTP-EQUIV="CACHE-CONTROL" CONTENT="NO-CACHE">

CONTENT-TYPE – Define o tipo de conteúdo.

EXEMPLO: <META HTTP-EQUIV="CONTENT-TYPE" CONTENT="text/html; charset=UTF-8">

EXPIRES - Define o prazo de expiração, indicando o tempo de obsolescência do documento. Caso coloquemos o número 0, estaremos indicando que a página expira agora e deve ser atualizada.

EXEMPLO: <META HTTP-EQUIV="EXPIRES" CONTENT="Sex, 13 Nov 2009 11:12:01 GMT">

REFRESH - Define o tempo, em segundos, de atualização da página e, ainda, pode redirecioná-la após o prazo.

EXEMPLO: http://www.estacio.br/index2.html">

CACHE-CONTROL - Define como o cachê será controlado. Exemplo: <META HTTP-EQUIV="CACHE-CONTROL" CONTENT="NO-CACHE">

PRAGMA NO-CACHE – Indica para não armazenar a página no cache, devendo esta ser sempre solicitada ao servidor.

KEYWORDS – Define as palavras chave para a indexação da página. Todas as palavras devem ser separadas por vírgula.

EXEMPLO: <META NAME="KEYWORDS" CONTENT="Universidade, graduação, ensino a distância ">

ROBOTS - Indica como os robôs de busca e indexação deverão trabalhar em sua página.

EXEMPLO: <META NAME="ROBOTS" CONTENT="INDEX,NOFOLLOW"> . Neste caso, estamos indicando aos robôs de busca que indexem a página e não sigam os links contidos nela.

Existem outros atributos dentro de meta, porém estes são os mais utilizados.

5 Tags para web

Algumas tags são importantes para o desenvolvimento da página Web como, por exemplo, as tags de corpo, quebra de linhas e centralizações de elementos.

5.1 Tags de Corpo

- **Comentários**

Tag: <!-- -->

Esta tag é utilizada para aplicar comentários ao código HTML. Não se esqueça que ela deve ser fechada por dois sinais de menos e um sinal de maior para limitar o tamanho do comentário. Caso contrário, todo o conteúdo ficará escondido do usuário.

Exemplo:

<!-- Este é o meu comentário. Quando o navegador passar neste ponto, ele não apresentará este texto na tela -->

- **Âncora**

Tag: <a>

Esta tag define uma âncora dentro ou fora da página.

Através dela, é possível fazer uma ligação com outro documento ou ainda saltar para um ponto específico do documento sem a necessidade da ação de paginação feita pelo usuário.

Código html

```
<html>
```

```
<body>
```

```
<a href="http://www.google.com.br">Acesse o Google!</a>
```

```
</body>
```

```
</html>
```

Navegador

Acesse o Google!

5.2 Caracteres e seus Controles

- **Negrito Tag: **

Através desta tag, podemos indicar ao navegador que negrite o texto.

Código html

```
<html>
<body>
<b>Texto em negrito</b> <br />
</body>
</html>
```

Navegador

Texto em negrito

- **Itálico Tag: <i>**

Para aplicar o efeito itálico no caracter, basta utilizar a tag <i>, não se esquecendo de fechá-la, pois, caso isto não ocorra, todos os elementos textuais ganharão este estilo.

Código html

```
<html>
<body>
<i>Estou em itálico</i>
</body>
</html>
```

Navegador

Estou em itálico

- **Tag: <big>**

Esta tag enfatiza o conjunto de caracteres ligados a ela não só aumentando seu tamanho em letra como também a espessura do corpo de letra.

Código html

```
<html>
<body>
<big>Este texto está em Big</i>
</body>
</html>
```

Navegador

Estou em *itálico*

- **Tag: **

Outra forma de enfatizar o texto. A diferença é que, neste caso, o texto ficará em *itálico*.

- **Tag: **

Esta tag é semelhante a tag .

- **Tag: <small>**

Publica o texto de forma diminuta.

Em HTML, esta tag faz com que o texto seja apresentado em uma fonte pequena e estreita.

- **Tag: <strike>**

Esta tag é utilizada quando queremos apresentar um texto todo ele riscado na horizontal.

É como se tudo aquilo que estivesse escrito fosse eliminado, porém desejamos que fosse visto.

A tag <strike> foi descontinuado no HTML 4.0.1 e não é suportada pelo DTD Strict do XHTML 1.0

- **Tag: **

O HTML é rico em recursos e nos permite definir as características das fontes a serem apresentadas no navegador Web do usuário. Existem muitas fontes a serem utilizadas. Tenha cuidado, pois nem todas estão disponíveis nas máquinas dos usuários na Internet. Prefira fontes mais conhecidas, tais como Arial, Helvetica, Times New Roman, Verdana e Courier New. Elas estão disponíveis na maioria dos ambientes gráficos dos sistemas operacionais Gnu/Linux, MacOS e Windows. O elemento font foi descontinuado no HTML 4.0.1 e não é suportado pelo DTD do Strict do XHTML 1.0.

Código html

```
<html>
```

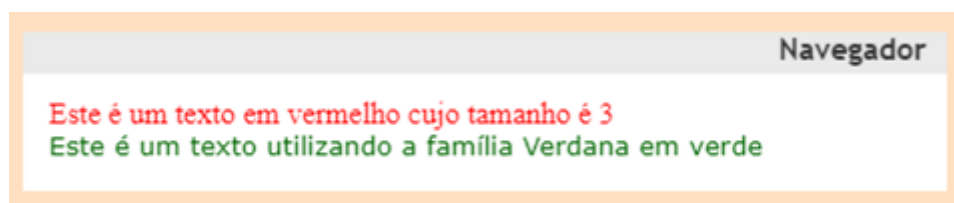
```
<body>
```

```
<font size="3" color="red">Este é um texto em vermelho cujo tamanho é 3</font><p>
```

```
<font face="verdana" color="green">Este é um texto utilizando a família Verdana em verde</font>
```

```
</body>
```

```
</html>
```



6 Controle de Caracteres

- Quebra de Linha

- Tag: **
**

Quando desejarmos quebrar uma linha, tal como fazemos em nosso editor de texto, basta utilizar a tag **
**.

- Tag: **<p>**

Já esta tag representa um salto maior. Ela simboliza a quebra de parágrafo feita no editor de textos.

- Centralização de Elementos

Tag: **<center>**

Para centralizarmos um texto ou elemento na tela do seu navegador, basta utilizarmos a tag **<center>**.

Ela permite a centralização de forma independente da resolução gráfica apresentada em seu monitor.

7 Tags para web - listas

Ao utilizarmos uma lista, precisamos definir seu começo e fim e seus itens. Para isso, deveremos trabalhar com algumas tags em bloco, tais como **<dd>**, **<dl>** e **<dt>**.

Primeiramente, devemos definir a lista através do bloco de tags **<dl>**

<dl>

<!-- Definir itens -->

</dl>

Agora você sabe que para definir uma lista é necessário usar a tag **<dl>**. Em uma tradução livre, poderemos dizer que esta tag significa definição da lista. Vamos definir os itens em uma lista **<dt>** e os seus respectivos conteúdos **<dd>**.

Para facilitar o entendimento, **<dt>** indica a definição do item e **<dd>** definição dos dados do item.

- Tags: **<dl>**, **<dd>**, **<dt>**

Quando necessitamos elencar itens em uma lista.

Esta tag deve ser utilizada juntamente com a tag **<dl>** e **<dt>**.

Exemplo:

<dl>

<dt>Graduação em Análise de Sistemas</dt>

<dd>Confira</dd>

<dt>Graduação em Desenvolvimento Web</dt>

<dd>Confira</dd>

</dl>

Listas Ordenadas

- **Tag: **

Para a apresentação de itens ordenados utilizando marcadores numéricos, você deve utilizar a tag para apresentar uma lista ordenada. Cada item da lista deve ser indicado pela tag , traduzido livremente para lista de itens.

<p>Lista Ordenada:</p>

Gás

Luz

Telefone

Listas Não Ordenadas

- **Tag: **

A lista não ordenada somente apresenta marcadores junto ao item.

Não há como verificar ordenação dos dados.

<p>Lista Não Ordenada:</p>

Gás

Luz

Telefone

- **Tag: <option>**

Existe uma maneira interessante de se exibir uma lista não ordenada economizando bastante espaço na tela. Para isso, devemos utilizar a tag <option>, comumente empregada em formulários cujo objetivo é armazenar grandes listas e poupar espaço na apresentação no navegador do usuário.

8 Embarcando objeto em sua página (Tag: <embed>)

A tag <embed> permite que você acople objetos externos a sua página como um plug-in. Hoje temos vários sites que oferecem seus objetos codificados nesta tag. Temos o Youtube, Google Vídeo entre outros.


- **Atributos**
- **height** – altura em pixels do objeto
- **src** – origem do objeto
- **type** – tipo do objeto
- **width** – largura em pixels

```
<embed autoplay="false" src="universidade.wma" width="50" height="50"></embed>
```

9 Cabeçalhos (Tag: <h1>, <h2>, <h3>, <h4>, <h5>, <h6>)

As tags de h1-h6 foram muito utilizadas no passado para dar ênfase ao tamanho dos textos utilizados em cabeçalhos. O valor 1 indica o maior e o 6 o menor tamanho das letras utilizadas nos cabeçalhos. Para um melhor entendimento, pense que o cabeçalho se assemelha ao Estilo utilizado pelos editores de texto. Eles não só alteram o tamanho como a aparência da letra, grifando-a.

Veja um exemplo:

**EXEMPLO**

```
<html>
<body>

<h1>Cabeçalho com h1</h1>
<h2>Cabeçalho com h2</h2>
<h3>Cabeçalho com h3</h3>
<h4>Cabeçalho com h4</h4>
<h5>Cabeçalho com h5</h5>
<h6>Cabeçalho com h6</h6>

</body>
</html>
```

Cabeçalho com h1
Cabeçalho com h2
Cabeçalho com h3
Cabeçalho com h4
Cabeçalho com h5
Cabeçalho com h6

Os recursos de cabeçalhos foram descontinuados no HTML 4.0.1 e não são suportados pelo DTD do XHTML 1.0 Strict.

10 Outros recursos

- **<hr>**

LINHA HORIZONTAL

Em HTML é fácil criarmos uma linha horizontal que ocupe toda a extensão da página sem nos preocuparmos com a definição do tamanho.

Basta que usemos a tag `<hr>`, que em uma tradução livre significa linha horizontal.

Este recurso facilita a divisão visual de elementos no corpo da página.

Exemplo: `<hr>`

``

Em HTML é possível associarmos imagens ao texto na página. Este tipo de recurso trouxe novos horizontes, pois muitas páginas começaram a ganhar vida com fotos e imagens associadas, permitindo a criação de sites muito interessantes. Esse recurso se expandiu com o uso de CSS pela facilidade do posicionamento dos elementos dentro da página. Percebam que, diferentemente dos editores de texto, as imagens em HTML são vinculadas e não inseridas. Existem apenas links para a fonte da imagem.

Esta tag possui os atributos `src`, onde indicaremos a fonte da imagem, o atributo `alt`, que facilita a identificação com um texto alternativo no caso da não visualização da mesma.

Em XHTML a tag `` deve ser fechada. Os elementos `align`, `border`, `hspace` e `vspace` foram descontinuados na versão HTML 4.1 e no DTD do XHTML 1.0 Strict

- **<map>**

Esta tag é utilizada para criarmos regiões clicáveis em uma imagem do lado do cliente. Através dela, podemos mapear pontos sensíveis de uma imagem que redirecionam o usuário de acordo com o contexto clicado dentro da imagem.

Atributos:

- **name** - nome do mapa
- **area** - define uma área clicável
- **shape** – define o formato da região clicável
- **coords** – define as coordenadas `x_min, y_min, x_max, y_max`
- **alt** – informação alternativa

Exemplo:

`<body>`

``

`<map name="logo">`

`http://www.universidade.br/ajuda.html<`

<http://www.universidade.br/cursos.html>">

<http://www.universidade.br/vestibular.html>">

</map>

</body>

- <frameset> e <frame>

Frames

A tecnologia do frame foi criada com o intuito de dividir a página apresentada ao usuário em segmentos distintos, como se fossem janelas, possibilitando a cada um o carregamento independente do conteúdo a ser apresentado. Cada frame está contido em um frameset (conjunto de frames) e cada um pode ter atributos diferentes, tais como tamanho, paginação entre outros. Em uma tradução livre, frame que dizer moldura.

Desta forma, temos para um conjunto de frames um frameset para organizá-los. Vide o exemplo:

```
<frameset cols="25%,50%,25%">
```

```
<frame src="frame_a.html" />
```

```
<frame src="frame_b.html" />
```

```
<frame src="frame_c.html" />
```

```
</frameset>
```

No exemplo acima, temos 3 frames com diferentes tamanhos. Os tamanhos foram definidos com base no porcentual da tela do navegador, isto é, eles se redimensionam a cada resolução apresentada.

No caso da existência de navegadores que não interpretam Frames poderemos utilizar a tag <noframes> Confira no exemplo abaixo:

Exemplo:

```
<frameset rows="18%,82%">
```

```
<frame src="frame_menu.html">
```

```
<frame src="frame_conteudo.html">
```

```
<noframes>Para visualizar frames é necessário um navegador mais recente como o Mozilla Firefox</noframes>
```

```
<iframe>
```

Este recurso teve se apogeu há 4 anos, quando muitos sites criavam frames internos para esconder de seus usuários a carga de determinadas informações. Até hoje, encontramos soluções interessantes que fazem uso desta tag.

Alguns navegadores podem entender seus resultados de forma diferente. Principalmente ao utilizarmos controles de sessão.

A tag <iframe> não é suportada pelo DTD Strict do HTML 4.1 e pelo DTD Strict do XHTML 1.0

- **Atributos:**

- **align** - alinhamento – left, right, top, bottom, middle. Descontinuado
- **frameborder** – tamanho da borda – 1 e 0
- **height** – estabelece a altura do iframe em pixels
- **width** – estabelece a largura do iframe em pixels
- **longdesc** – aponta para um arquivo que contém informações adicionais a respeito daquele iframe
- **marginheight** – especifica as margens superior e inferior em pixels
- **marginwidth** – especifica a margem a direita e a esquerda do iframe
- **name** – nome do iframe
- **scrolling** – habilita ou não a barra de scroll
- **src** – indica a URL com o caminho do documento a ser carregado pelo iframe

Exemplo:

```
<iframe src="conteudo.jsp" width="100%" height="200">
```

```
<p>Seu navegador não suporta iframes</p>
```

```
</iframe>
```

O que vem na próxima aula

Na próxima aula, você vai estudar:

- como funciona a linguagem mais popular da Internet - HTML.

CONCLUSÃO

Nesta aula, você:

- Aprendeu quais são as principais tags utilizadas no desenvolvimento de páginas Web.
- Aprendeu como desenvolver sites com aspectos contemporâneos em seus códigos.