

Tutorial

1. O que é HTML?

HTML é uma linguagem para descrever páginas web. HTML vem do ingles **Hyper Text Markup Language** que significa Linguagem de Marcação de Hipertexto.

Para entender este conceito vamos a separar os termos que o conformam:

- **Linguagem:** é um método/maneira padronizada para comunicar instruções para um computador.
- **Marcação:** representa a anotação de algum elemento dentro do documento. Em HTML, a marcação é feita com **tags** para indicar a estrutura do documento.
Tags: são marcas especiais definidas por uma etiqueta que descreve o papel de cada elemento dentro do texto. As tags em HTML permitem ao browser interpretar e exibir a página conforme ela foi concebida.
- **Hipertexto:** é um modelo de documento que contem blocos de textos, palavras, imagens entre outros e que têm referências específicas denominadas hiperlinks (ou simplesmente links) que ligam o documento com outros documentos de hipertexto.
Basicamente, é um documento composto por páginas e ligações entre elas.

Resumindo:

HTML (linguagem de marcação de hipertexto) é uma linguagem que permite marcar hipertextos com o intuito de adicionar informações particulares sobre esse texto. Basicamente, HTML indica a estrutura de um documento mediante o uso de tags; pode-se dizer então que os comandos usados para passar instruções ao computador com a linguagem HTML são as tags.

Um arquivo HTML deve ter uma extensão de arquivo **htm** ou **html** e pode ser criado usando um simples editor de textos. Os documentos ou arquivos html também são chamados páginas web.

2. Breve Historia

A linguagem de HTML foi originalmente concebida por Tim Berners-Lee, e popularizada pelo browser Mosaic. No decorrer dos anos 90 HTML floresceu com o crescimento explosivo da Web. Durante este tempo, o código HTML expandiu-se de variadas maneiras. A Web depende do fato em que autores das páginas web e as empresas compartilhem as mesmas convenções HTML. Isso levou a que se realizasse um trabalho conjunto na criação das especificações HTML [1].

Foi assim, que a partir do meio da década dos 90, foi criado o W3C, entidade com objetivo específico de desenvolver padrões para serem usados na Web. Dessa maneira surgiram rapidamente diversas versões que mostraram o processo de evolução da linguagem.

Version	Year	Recommendations (recomendações do W3C)
HTML	1991	HTML inicial não estandarizado
HTML+	1993	Não chegou a ser recomendação
HTML 2.0	1995	W3C Recommendation
HTML 3.2	1997	W3C Recommendation
HTML 4.01	1999	W3C Recommendation

XHTML 1.0	2000	W3C Recommendation
HTML5	2014	W3C Recommendation

Cada versão do HTML tentou refletir um maior consenso entre os industriais, de modo que o investimento feito pelos fornecedores satisfeitos não se desperdice e ainda que os seus documentos não se tornem ilegíveis ou inválidos num período de tempo curto [1].

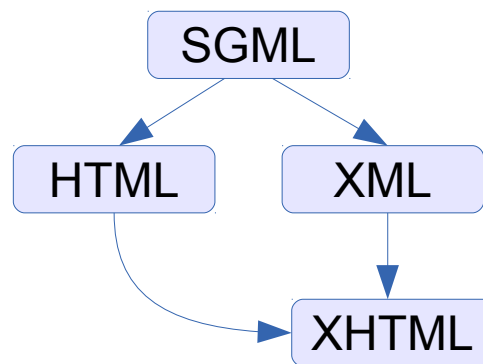
3. HTML e XHTML

Quando o **HTML** foi criado, ele baseou-se na metalinguagem¹ **SGML** (Standard Generalized Markup Language). A SGML permite definir linguagens de marcação para documentos, HTML é uma linguagem definida com SGML (exceto HTML 5).

XML (*eXtensible Markup Language*) é outra metalinguagem baseada em SGML. XML é uma recomendação da W3C para gerar linguagens de marcação para necessidades especiais. Pode-se considerar então que XML é um subtipo/subconjunto da SGML projetada para ser mais simples de se analisar gramaticalmente e de se processar do que SGML (tentativa de simplificar SGML).

XHTML (Extensible HyperText Mark-up Language) 1.0 nasceu como uma reestruturação do HTML 4 existente nessa época. XHTML é uma maneira mais bem estruturada de escrever código para páginas web. Ele se baseia em XML no lugar de se basear em SGML. XHTML faz uso de uma sintaxe mais rigorosa e menos ambígua para tornar o HTML mais simples de ser processado e estendido.

O World Wide Web Consortium (W3C) continuam mantendo a recomendação HTML 4.01, e a especificação para HTML5 e XHTML5 está sendo ativamente desenvolvida.



4. Sintaxe²

HTML usa tags para descrever um documento.

Todas as tags começam com um sinal de menor "<" e acabam com um sinal de maior ">" como por exemplo <html>.

Usualmente encontramos dois tipos de tags:

- tags de abertura: <comando>

¹ Metalinguagem: linguagem existente para definir outras linguagens.

² Sintaxe é a forma como as instruções de uma linguagem são escritas, mas sem atender ao seu significado

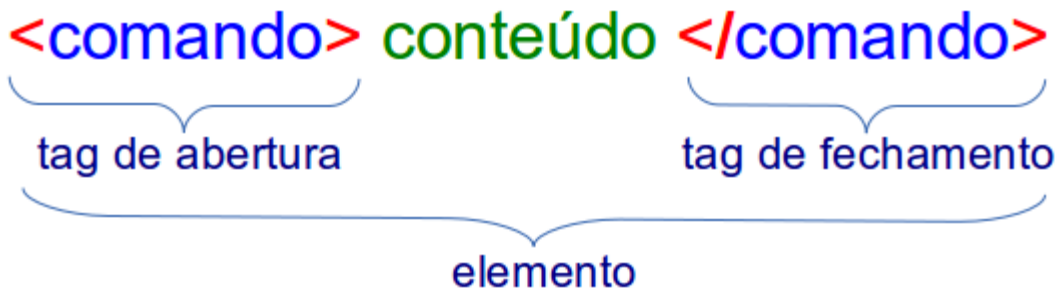
- tags de fechamento: `</comando>`. A diferença entre elas é que na tag de fechamento existe um barra `"/`.

O conteúdo que estiver contido entre uma tag de abertura e uma tag de fechamento será processado segundo o comando (nome-tag) contido na tag como mostra a figura a seguir.

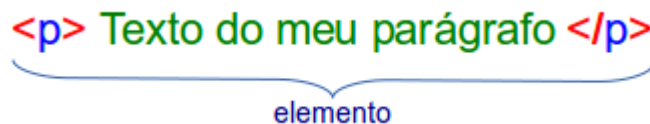


5. Elementos

Os elementos HTML constituem um documento html. Os elementos HTML são definidos usando tags (etiquetas) HTML. As tags servem para definir elementos HTML. Um elemento HTML é tudo o que estiver contido entre a tag de abertura e a tag de fechamento, incluindo as tags.



Exemplo: O elemento parágrafo é indicado pela tag `p` que informa ao navegador que todo o texto colocado entre `<p>` e `</p>` é um parágrafo



6. Atributos

As tags podem ter atributos. Os atributos podem prover informação adicional sobre os elementos HTML na sua página.

Os atributos vem em pares nome/valor como este: `nome="valor"`.

Os atributos são sempre adicionados à tag de abertura de um elemento HTML.

7. Diferenças entre HTML e XHTML

HTML	XHTML
Permite maiúsculas e minúsculas. Exemplos: <code><LINK REL="stylesheet" HREF="styles.css"></code> <code><link REL="stylesheet" href="styles.css"></code> <code><link rel="stylesheet" href="styles.css"></code>	Todas as tags e os atributos devem ser escritos em minúsculas Exemplo: <code><link rel="stylesheet" href="styles.css"></code>
Não precisamente deverá se fechar todas as tags. Exemplo: elementos vazios	Tags de fechamento são obrigatórias.

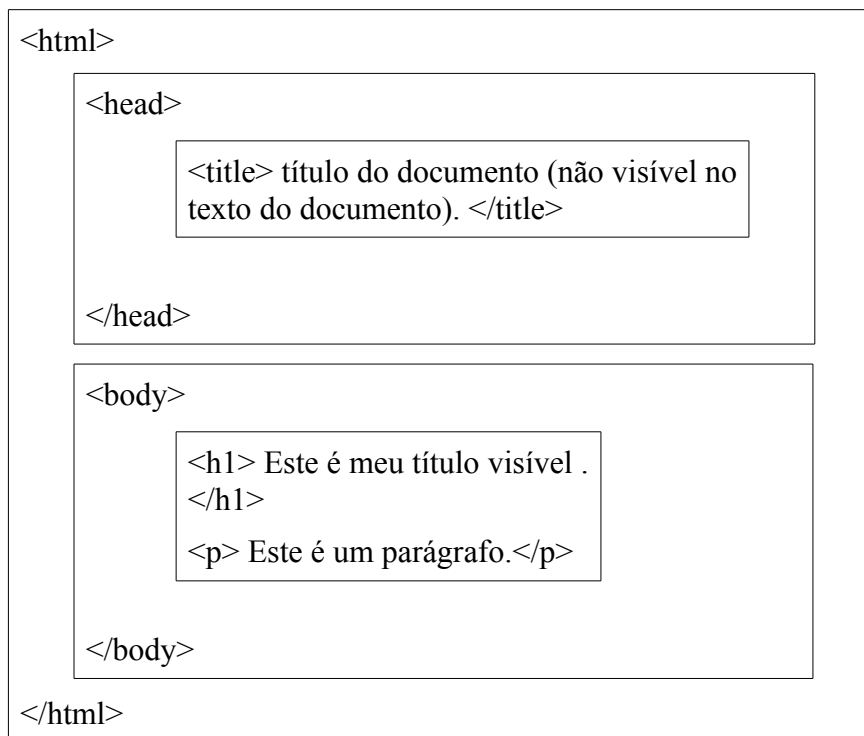
Elementos vazios não precisam de fechamento na mesma tag. Exemplos: <link rel="stylesheet" href="styles.css"> 	Elementos vazios* (aqueles que não possuem tag de fechamento) devem ser fechados com: espaço + / Exemplos: <link rel="stylesheet" href="styles.css" />
Em html os valores podem ou não ir entre aspas Exemplo: <link rel="stylesheet" href=styles.css>	Todos os valores dos atributos devem ir entre aspas: Exemplo: <link rel="stylesheet" href="styles.css">
Alguns atributos não possuem igualdade Exemplo: <input checked> O exemplo é correto em html e errado em xhtml	Todo atributo deve ter um valor associado a ele. Se o atributo não possuir igualdade em html, no xhtml deve ser igual a ele mesmo entre aspas. Exemplo: <input checked= "checked">
Tem o atributo name	Inutilização do atributo "name" em troca do atributo "id".

***Elementos vazios** possuem apenas uma tag de abertura que possui os seus respectivos atributos. Como não possuem tag de fechamento, no XHTML deve-se adicionar **espaço** + / no final do tag de abertura. **Exemplo:**

Recomenda-se a escrita de um código mais estrito seguindo as especificações do XHTML.

8. Estrutura

A estrutura básica de uma página html é ilustrada na Figura a seguir.



Como pode-se observar na Figura anterior, um documento html está formado por vários elementos básicos (adicionaram-se caixas ao redor de alguns elementos por fines ilustrativos). Os elementos identificam-se mediante seus tags respectivos. Dessa maneira podemos definir os elementos mais importantes:

- **HTML:** O texto colocado entre <html> e </html> descreve o site web. Dentro dessas tags temos duas seções : head e body que indicam o cabeçalho do documento e o corpo, respectivamente.
Em XHTML, o elemento HTML deve conter o atributo **xmlns** para associa-lo com um namespace XHTML. O namespace para XHTML é <http://www.w3.org/1999/xhtml>.
Exemplo: <html xmlns="http://www.w3.org/1999/xhtml">
O atributo xmlns é requerido em XHTML, mas é inválido em HTML.
- **Cabeçalho:** O texto entre <head> e </head> é a informação de cabeçalho. As informações de cabeçalho são configurações da página. O texto dentro de head não é mostrado na janela do navegador.
- **Corpo:** O texto entre as tags <body> e </body> define o início e o fim do corpo do documento HTML e é o texto que será mostrado no seu navegador (conteúdo visível).
- **Título:** O texto entre <title> e </title> é o título do seu documento. O título é mostrado na barra de título do navegador (a barra mais superior do navegador). Deve ser utilizado sempre dentro do cabeçalho.
- **Cabeçalho:** O texto entre as tags <h1> e </h1> representa um título.
- **Parágrafo:** O texto entre as tags <p> e </p> representa um parágrafo.

A seguir estudaremos cada elemento com maior detalhe:

9. Cabeçalhos ou Títulos

Títulos são definidos com as tags <h1> até <h6>.

<h1> define o título maior. A <h6> define o título menor.

```
<h1>Este é um título</h1>  
<h2>Este é um título</h2>  
<h3>Este é um título</h3>  
<h4>Este é um título</h4>  
<h5>Este é um título</h5>  
<h6>Este é um título</h6>
```

A HTML automaticamente adiciona uma linha em branco extra antes e após um título.

Lembre-se de usar cabeçalhos para por títulos ao seu documento e não para fazer q o texto se veja maior ou em negrito. Os motores de busca usam os cabeçalhos para indexar a estrutura e o conteúdo das suas páginas web.

Lembre-se que um cabeçalhos indica um título visível no documento como visto no exemplo. O elemento que usa as tags <title> e </title> no cabeçalho representa o título do documento de uma maneira abstrata e não fica visível dentro do documento (geralmente fica na aba ou na barra superior da janela do navegador).

Como os usuários costumam ver rapidamente os títulos de uma página web antes que o conteúdo, é

importante que os títulos mostrem a estrutura de um documento.

H1 deve ser usado para os cabeçalhos mais importantes, seguido de h2 e assim sucessivamente.

10. Parágrafos

Os parágrafos são definidos com a tag <p>.

```
<p>Este é um parágrafo</p>
<p>Este é outro parágrafo</p>
```

A HTML adiciona automaticamente uma linha em branco extra antes e após um parágrafo.

11. Links/hiperlink (vínculo/hipervínculo)

HTML usa um link para vincular-se a qualquer recurso na Web.

Um link pode ser usado para direcionar o usuário para uma outra página, para um ponto da mesma página, abrir uma imagem, fazer download de um arquivo, enviar um e-mail, etc.

HTML usa a tag <a> e (âncora) para criar um link.

Um link é composto por duas âncoras:

- A âncora-fonte: no documento de partida usando a tag <a> para criar uma âncora desde onde vincular. Neste documento de partida, as palavras entre as tags de abertura e fechamento da âncora serão exibidas como um link fonte (de partida).
- A âncora-alvo: representaria o documento a ser aberto, o arquivo a ser baixado, etc. Para indicar o recurso alvo do link, adiciona-se o atributo **href** (do inglês Hyperlink Reference) dentro do tag de abertura <a>.

Assim, a sintaxe de um link seria:

```
<a href="url">Texto a ser exibido</a>
```

Exemplo

Esta âncora define um vínculo para o google:

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

Atributo Alvo (Target)

Pode-se adicionar o atributo target para definir **onde** o documento vinculado será aberto. Se o valor do atributo for `_blank`, então o documento será aberto numa nova janela do navegador.

Exemplo:

```
<a href="http://www.google.com/" target="_blank">Visite as W3Schools!</a>
```

Links internos a uma página

O atributo nome é usado para criar uma âncora nomeada. Quando usamos âncoras nomeadas podemos criar vínculos que saltam diretamente para uma seção específica em uma página, em vez de deixar o usuário rolar [a janela (scroll)] por toda parte para encontrar o que ele/ela está procurando.

Abaixo está a sintaxe de uma âncora nomeada:

```
<a name="tips">Seção de Dicas Úteis</a>
```

O atributo nome é usado para criar uma âncora nomeada. O nome da âncora pode ser qualquer texto que você quiser usar.

Você deve observar que a âncora nomeada não é exibida de maneira especial.

Um hipervínculo para a Seção de Dicas Úteis de DENTRO do arquivo "html_links.asp" seria assim:

```
<a href="#tips">Saltar para a Seção de Dicas Úteis</a>
```

Âncoras nomeadas são geralmente usadas para criar "sumários" ("table of contents") no início de um documento grande. A cada capítulo dentro do documento é dado um nome de âncora, e vínculos para cada uma destas âncoras são colocados no topo do documento.

Se um navegador não puder encontrar uma âncora nomeada que foi especificada, ele vai para o topo do documento. Nenhum erro ocorre.

12. Imagens

As imagens são definidas com a tag . Img é um elemento vazio, portanto em XHTML é preciso adicionar **espaço** e / no final da tag. Assim, a sintaxe para definir uma imagem em XHTML é:

```

```

Para exibir uma imagem, você necessita usar o atributo src. Src significa "source" [fonte]. O valor do atributo src é a URL da imagem que você quer exibir na sua página.

O URL aponta para o local onde a imagem está armazenada. Se a imagem estiver na mesma pasta na qual esta o documento web que você está editando, basta escrever o nome da imagem.

```

```

Se o recurso está na web, pode se escrever o endereço web completo da imagem.

```

```

Atributo Alt

O atributo alt é usado para definir um "texto alternativo" para uma imagem. O valor do atributo alt é um texto definido pelo autor:

```

```

O atributo "alt" provee informação alternativa para uma imagem em caso o navegador não possa

carregar a imagem.

Atributos width e height

São usados para indicar uma altura e largura.

```

```

É uma boa prática especificar estes atributos para uma imagem pois assim o navegador reserva espaço para a imagem enquanto a página carrega, de outro modo, a disposição de elementos poderia variar quando terminar de carregar a imagem.

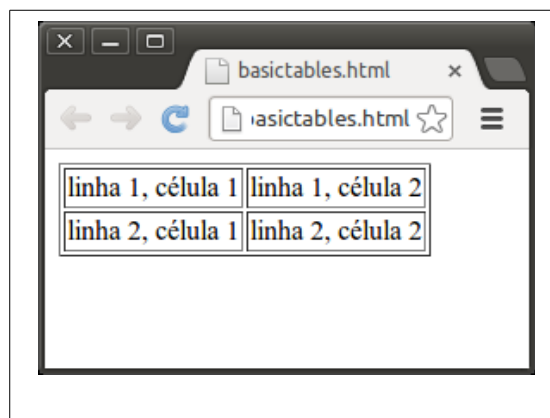
13. Tabelas

As tabelas são definidas com a tag <table>. Uma tabela é dividida em linhas (com a tag <tr> de “table row”), e cada linha é dividida em células de dados (com a tag <td> de “table data”). Uma célula de dados pode conter texto, imagens, listas, parágrafos, formulários, outras tabelas, etc.

Exemplo

```
<table border="1">
  <tr>
    <td>linha 1, célula 1</td>
    <td>linha 1, célula 2</td>
  </tr>
  <tr>
    <td>linha 2, célula 1</td>
    <td>linha 2, célula 2</td>
  </tr>
</table>
```

Como aparece no navegador:



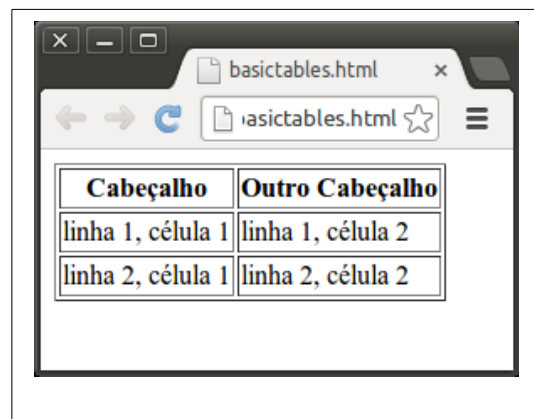
O atributo border serve para especificar a grossura da borda. Sem o atributo border, a tabela será mostrada sem bordas. O valor do atributo representa a grossura da borda começando com 1. A maior número a grossura vai se incrementando.

Cabeçalhos em uma Tabela: Os cabeçalhos em uma tabela são definidos com a tag <th> para representar uma célula no lugar de <td>.

Exemplo

```
<table border="1">
  <tr>
    <th>Cabeçalho</th>
    <th>Outro Cabeçalho</th>
  </tr>
  <tr>
    <td>linha 1, célula 1</td>
    <td>linha 1, célula 2</td>
  </tr>
  <tr>
    <td>linha 2, célula 1</td>
    <td>linha 2, célula 2</td>
  </tr>
```

Como aparece no navegador:




```
</table>
```

14. Listas

Servem para representar um conjunto de itens.

A HTML suporta listas ordenadas, não ordenadas e de definições.

Dentro de um item de uma lista você pode colocar parágrafos, quebras de linha, imagens, vínculos, outras listas, etc.

14.1 Listas Não Ordenadas

Representa uma lista de itens onde os elementos não precisam de uma ordem específica.

As listas de itens são marcadas com bullets (tipicamente pequenos círculos pretos).

Uma lista não ordenada começa com a tag `` (do inglês **u**nordered list). Cada item da lista começa com a tag `` (do inglês list **i**tem)

Exemplo:

```
<ul>
  <li>Café</li>
  <li>Leite</li>
</ul>
```

Como aparece no navegador:

- Café
- Leite

Dentro de um item de uma lista você pode colocar parágrafos, quebras de linha, imagens, vínculos, outras listas, etc.

14.2 Listas Ordenadas

Uma lista ordenada é também uma lista de itens. As listas de itens são marcadas com números.

Uma lista ordenada começa com a tag `` (do inglês **o**rdered list). Cada item da lista começa com a tag `` (do inglês list **i**tem).

Exemplo:

```
<ol>
  <li>Café</li>
  <li>Leite</li>
</ol>
```

Como aparece no navegador:

1. Café
2. Leite

14.3 Listas de Descrição

Uma lista de descrição tem um conjunto de termos e uma descrição associada a cada termo.

A tag `<dl>` (do inglês **d**escription list) define uma nova lista de descrição.

A tag `<dt>` (do inglês **d**escription term) define um termo, e `<dd>` (do inglês description **d**ata) define uma descrição.

Exemplo:

```
<dl>
  <dt>Café</dt>
```

Como aparece no navegador:

Café
- bebida escura e quente
Milk
- bebida branca

```
<dd>- bebida escura e quente</dd>
<dt>Leite</dt>
<dd>- bebida branca</dd>
</dl>
```

15. Outros elementos

Comentários em HTML

A tag de comentário é usada para inserir um comentário no código fonte HTML. Um comentário será ignorado pelo navegador.

Exemplo

```
<!-- Este é um comentário -->
```

Observe que você necessita de um ponto de exclamação após a chave de abertura, mas não antes da chave de fechamento.

Quebras de linha

A tag **br** é usada quando você quer terminar uma linha, mas não quer começar um novo parágrafo. A tag **br** força a quebra de linha onde quer que você a coloque.

```
<p>Este <br /> é um parágrafo com quebras de linha</p>
```

A tag **br** é uma tag vazia. Portanto em html pode-se usar a tag **
**. Porém em xhtml deve-se usar a tag **
**.

16. Formatação de texto

HTML define uma série de elementos para formatar a saída, como texto em negrito ou itálico.

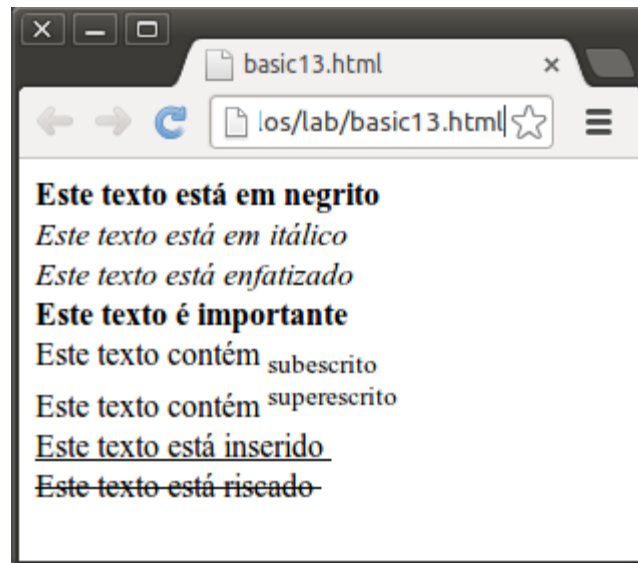
A seguir mostram-se alguns tags de formatação :

Tag	Descrição vem do inglês:	Descrição em português
	bold	negrito
<i>	<i>italic</i>	itálico
	emphasized	ênfático
	strong	Texto importante
<sub>	subscript	subscrito
<sup>	superscript	superescrito
<ins>	inserted	inserido
	deleted	cancelado (riscado)

*Lembre-se que geralmente o nome dos tags vêm do nome em inglês daquilo que representam (2º col).

Exemplo:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
  <body>
    <b>Este texto está em negrito</b>          <br />
    <i>Este texto está em itálico</i>         <br />
    <em>Este texto está enfatizado </em>      <br />
    <strong> Este texto é importante </strong><br />
    Este texto contém <sub>subescrito</sub>  <br />
    Este texto contém <sup>superescrito</sup> <br />
    <ins>Este texto está inserido </ins>      <br />
    <del>Este texto está riscado </del>       <br />
  </body>
</html>
```



17. Estilos - CSS

Junto com a versão 4.0 de HTML, o CSS foi introduzido. O intuito do CSS é fornecer uma forma melhor para dar estilo aos elementos HTML. CSS vem do inglês: “Cascading Style Sheets” que significa Folha de Estilos em Cascata.

CSS é uma linguagem para estilos que define o layout de documentos HTML. Por exemplo, CSS controla fontes, cores, margens, linhas, alturas, larguras, imagens de fundo, posicionamentos e muito mais.

HTML pode ser (in)devidamente usado para definir o layout de websites. Contudo CSS proporciona mais opções e é mais preciso e sofisticado. CSS é suportado por todos os navegadores atuais.

HTML é usado para estruturar conteúdos. CSS é usado para formatar conteúdos estruturados.

Sintaxe do CSS

Para definir a formatação precisamos definir regras.

Uma regra CSS tem dois partes principais: um seletor, e um conjunto de declarações como mostra a Figura a seguir.



O seletor indica o elemento(s) HTML ao qual se deseja dar estilo. Cada declaração consiste de uma propriedade (atributo visual que se deseja mudar/definir) e um valor atribuído a essa propriedade.

Para definir um conjunto de propriedades visuais para o mesmo seletor, usaremos varias declarações como mostra a Figura a seguir.



Como Usar Estilos

Quando um navegador lê uma folha de estilo, ele formata o documento html de acordo com ela. Há três maneiras de inserir uma folha de estilo:

- ***Inline (local)***

Um estilo local (inline) deve ser usado quando um estilo único deve ser aplicado a uma única ocorrência de um elemento.

Para usar estilos locais (inline) você usa o atributo de estilo na tag relevante. O atributo de estilo pode conter qualquer propriedade CSS. O exemplo mostra como mudar a cor e a margem esquerda de um parágrafo:

```
<p style="color: red; margin-left: 20px;">
  Este é um parágrafo
</p>
```

- ***Internal (Incorporada no head)***

Uma folha de estilo interna é ideal quando um documento específico tem um estilo único. Você define estilos internos dentro do elemento `<head>` com a tag `<style>`.

```
<head>
  <style type="text/css">
    body {background-color:red;}
    p {color:blue;}
  </style>
</head>
```

- **External (Folha de Estilo Externa)**

Uma folha de estilo externa é ideal quando o estilo é aplicado em muitas páginas. Com uma folha de estilo externa, você pode mudar a aparência de um sítio Web inteiro pela mudança de um arquivo único CSS. Cada página deve vincular-se à folha de estilo usando a tag <link>. A tag <link> vai dentro do elemento <head>.

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

Contudo, a forma mais aconselhável para adicionar CSS a um documento HTML é pondo a sintaxe CSS em um arquivo separado (external).

18. DOCTYPE

Um DOCTYPE deve ir posicionado na primeira declaração de um documento HTML.

Um DOCTYPE (do inglês Document Type) indica ao navegador qual é o tipo e versão do documento que deve analisar após essa declaração. Como existem muitos tipos de documentos diferentes na web, um navegador exibirá adequadamente uma página HTML se conhece o seu tipo e versão.

É importante notar que a declaração DOCTYPE não é uma tag HTML, senão uma instrução para o navegador antes de começar com a sintaxe HTML.

O DOCTYPE da maior parte dos documentos HTML e XHTML requerem uma referência para uma DTD (**D**ocument **T**ype **D**efinition) que significa Definição do Tipo de Documento. Uma DTD especifica as regras de uma linguagem de marcação. Já o HTML5 não requer um DTD. Mas ainda não é uma versão oficial.

Exemplos:

HTML5 (não precisa DTD)

```
<!DOCTYPE html>
```

HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

Tipos de DOCTYPE para XHTML

Existem três tipos de DOCTYPE para XHTML:

- **Strict:**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

É a mais rígida das declarações. Documentos neste modo não admitem qualquer item de formatação dentro dos elementos e nem elementos em desuso (“deprecated”) segundo as especificações do W3C. São indicados para o uso com folhas de estilo em cascata, com marcação totalmente independente da apresentação

- **Transitional:**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

Possui maior flexibilidade e é indicada para documentos que ainda utilizam formatação embutida em tags e elementos em desuso (“deprecated”). Não admite marcação para frames

- **Frameset**

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">

Admite tudo o que admite a declaração Transitional e acrescenta os elementos específicos para frames.

Bibliografia

[1] <http://www.w3.org/TR/html401/intro/intro.html>.

[2] <http://www.w3schools.com/>

[3] <http://html.net/>