



## Manual Básico - HTML

Izaquiel Queiroz <sup>1</sup>

<sup>1</sup> Graduando SI na UFRPE-UAST.

E-mail: [izaquielqueiroz@live.com](mailto:izaquielqueiroz@live.com)

**Resumo:** Este manual tem por fim apresentar de forma simples os conceitos básicos para a criação de sites com HTML, manipulando seus códigos de forma organizada, mantendo um padrão organizado e contextualizado de como manter as boas práticas determinadas pela W3C.

**Abstract:** This manual aims to present simply the basics for creating websites with HTML, manipulating its organized form of codes, keeping an organized pattern and contextualized keeping best practices determined by the W3C.

### Introdução

HTML (HyperText Markup Language) é uma linguagem de marcação usada para transformar documentos de textos em páginas da web. O objetivo final do HTML como uma linguagem de marcação é proporcionar uma descrição semântica (o significado) do conteúdo e estabelecer uma estrutura (uma hierarquia de elementos) ao documento.

### HTML x XHTML

XHTML (eXtensible Hypertext Markup language) é uma reformulação do HTML 4.01 conforme regras mais rigorosa de sintaxe XML (eXtensible Markup language). Os elementos são os mesmos do HTML, mas há mais restrições para a marcação de documentos.

O XHTML foi criado em um tempo em que as pessoas foram sentindo necessidade de mais sofisticação. Cada dia que passava, mais adereços eram inseridos no código, e o que era pra ser simples acabou virando uma sopa de letrinhas. Já não era mais apenas conteúdo, havia também a forma. Algumas regras foram adicionadas, alguns elementos excluídos. Usamos o XHTML para

definir a estrutura do site, e o CSS, para dar forma a ele. A parte editorial é sustentada pelo primeiro e o design, pelo segundo.

### Três versões de (X)HTML

As recomendações HTML 4.01 e XHTML 1.0 abrangem três levemente diferentes documentos de especificação, chamados Document Type Definitions (ou DTDs). Estes elementos definem todos os elementos, atributos e entidades juntamente com as regras para o seu uso.

Strict – diz ao navegador que o documento possui apenas elementos recomendados para o XHTML e respeita todas as regras.

```
<!DOCTYPE html
```

```
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

Transitional – diz ao navegador que o documento possui elementos XHTML, mas pode conter alguns elementos de HTML 4.0. É o modelo utilizado na fase de transição, quando o desenvolvedor ainda não tem o domínio completo das regras de XHTML.

```
<!DOCTYPE html
```

```
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Frameset – diz ao navegador que o documento possui elementos XHTML, mas usa a tecnologia de frames ou iframes. Esse modelo aceita as regras de HTML 4.0.

```
<!DOCTYPE html
```

```
PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-Frameset.dtd">
```



## UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO – UFRPE UNIDADE ACADÊMICA DE SERRA TALHADA - UAST

Obs.: A declaração frameset está em desuso, pois o uso dessa tecnologia foi descontinuado pelo W3C.

### Estrutura Mínima de Documento

Esta amostra de marcação expõe a estrutura mínima de um documento HTML 4.01. Este exemplo usa o DTD Estrito do HTML:

```
<DOCTYPE HTML PUBLIC "-//W3// DTD HTML 4.01//EN"
```

```
    "http://www.w3.org/TR/html4/strict.dtd">
```

```
<html>
```

```
    <head>
```

```
        <title>Título do documento</title>
```

```
    </head>
```

```
    <body>
```

```
        Conteúdo do documento...
```

```
    </body>
```

```
</html>
```

### Elemento raiz único

Todo documento deve possuir um elemento raiz, onde todo o código deve ser inserido. Essa função é exercida pelo elemento html.

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

### Referenciar um namespace de XML



## UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO – UFRPE UNIDADE ACADÊMICA DE SERRA TALHADA - UAST

Essa exigência é em função do XHTML ser parte do vocabulário do XML. Para nós, é necessário apenas saber que essa exigência deve ser cumprida. A forma como ela deve ser escrita é a seguinte:

```
<html xmlns:http://www.w3.org/1999/xhtml>
```

### Elementos básicos

Todo documento XHTML deve conter obrigatoriamente os elementos html, head, title e body.

### Nome dos elementos em letras minúsculas

Todo o código deve ser escrito em letras minúsculas. Tanto o nome dos elementos quanto os nomes dos atributos. O valor do atributo não precisa seguir a regra.

Errado: <P Class="Destaque">

Certo: <p class="Destaque">

Obs.: Um atributo é uma regra ou característica que altera um elemento (tag). Alguns atributos específicos de determinadas tags e outros são genéricos, podendo ser utilizados em várias tags.

### Os elementos deve possuir abertura e fechamento

Todo o código é descrito por tags, que são os elementos da página. Todo elemento é composto por uma tag de abertura e uma de fechamento, que é o elemento precedido de um sinal de barra normal (/).

Errado: <p>Parágrafo de um conteúdo da página

Certo: <p> Parágrafo de um conteúdo da página</p>

### Elementos vazios devem ser fechados

Na linguagem HTML há elementos vazios. São elementos que não possuem uma tag de fechamento. A solução para que essa regra fosse cumprida



## UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO – UFRPE UNIDADE ACADÊMICA DE SERRA TALHADA - UAST

também nesses casos foi o acréscimo dos sinais de espaço e barra ( /) antes do final do final da tag, que é marcado pelo sinal de maior (>).

Errado: <hr>

Certo : <hr />

### Sequência certa de abertura e fechamento

Existe uma sequência correta para o fechamento das tags. A ordem correta é fechar as tags na ordem inversa de sua abertura.

Errado: <p><strong>Parágrafo escrito em negrito</p></strong>

Certo: <p><strong>Parágrafo escrito em negrito</strong></p>

### Valores dos atributos entre aspas

Vimos anteriormente que o código deve ser escrito em letras minúsculas e que isso vale para os atributos. Outra regra para esses dados é que o valor do atributo deve estar sempre entre aspas(“”).

Errado: <p title=Descrição do parágrafo>Este é um parágrafo.</p>

Certo: <p title="Descrição do parágrafo">Este é um parágrafo.</p>

### Atributos vazios não podem ser simplificados

Na linguagem HTML alguns atributos não possuem valor, são os chamados atributos vazios. Para que essa regra fosse cumprida, a solução foi duplicar o nome do atributo como seu valor.

Errado: checked

<input type="checkbox" name="assinatura value="sim" checked />

Certo: checked="checked" />

<input type="checkbox" name="assinatura value="sim" checked="checked" />

## **Criando uma página**

Conforme descrito nas “Regras de XHTML”, além das declarações obrigatórias e do elemento raiz único html, uma página é composta por cabeçalho e corpo (head e body, respectivamente). Os elementos entre as tags <head> e </head> compõem o cabeçalho do documento e não são exibidos na janela do navegador. Os elementos que estão nas tags <body> e </body> compõem o corpo do documento e são exibidos na janela do navegador.

Dentro do cabeçalho há o elemento title, de uso obrigatório. Esse elemento define o título da página e não o título do conteúdo. O título de um conteúdo aparece na janela de navegação, enquanto o título definido entre as tags <title> e </title> aparece em outros lugares como as abas de navegação do navegador.

## **Salvando um arquivo HTML**

Para criar um documento HTML é preciso de um programa editor de texto, como o bloco de notas, word, entre outro. Para salvar um arquivo no formato HTML após apertar o botão SALVAR do editor de texto usado, na opção file name(nome do arquivo) digite o nome completo do arquivo com a extensão .html no fim do nome. Na opção Save as type (salvar como tipo), escolha a opção all files (todos os arquivos). Na opção encoding (codificação), escolha a opção UTF-8 e aperte no botão salvar. Um arquivo HTML foi criado e já pode ser aberto em qualquer navegador.

## **Tags meta**

As tags meta são os metadados da sua página, ou seja, são informações sobre o documento. A maior parte do conteúdo descrito nas tags meta é utilizado pelo próprio navegador ou por motores de busca como google, Bing, Yahoo. As tags meta são digitadas dentro das tags <head> </head>.

<meta name="keywords" content="palavras-chaves, textos utilizados para sua página ser encontrada por mecanismos de pesquisa" />



## UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO – UFRPE UNIDADE ACADÊMICA DE SERRA TALHADA - UAST

`<meta name="description" content="Descrição da página reconhecida pelos mecanismos de busca" />`

`<meta name="author" content="Nome do autor da página, nesse caso, você." />`

### **Parágrafos e quebras de linhas**

Colocar a tag `<p>` antes do texto e `</p>` após cria um parágrafo determinado pelo conteúdo contido dentro dessas tags. Lembrando que essas tags de parágrafos devem ser colocadas dentro do corpo do documento, entre `<body>` e `</body>`.

`<body>`

`<p>parágrafo no conteúdo do documento</p>`

`<p>Outro parágrafo</p>`

`</body>`

Os parágrafos têm como característica principal a adaptação de seu conteúdo à janela do navegador. Se a janela estiver expandida, haverá mais espaço para o texto acomodar-se. Se a janela do navegador estiver menor, serão necessárias mais linhas para acomodar a mesma quantidade de texto.

Pular linhas no editor de texto não significa criar novos parágrafos ou espaços entre os conteúdos na página exibida pelo navegador. Deve-se indicar o elemento estrutural desejado. Para as quebras de linhas existe a tag `<br />`.

Exemplo:

`<body>`

`<p>Conteúdo`

`<br />`

`Conteúdo após uma quebra de linha</p>`

`</body>`

As quebras de linha forçam o texto a ir para a linha de baixo, mas o parágrafo ainda é o mesmo. Note que o espaçamento entre parágrafos é maior que o espaço entre quebras de linhas.

### **h1, h2, h3, h4, h5, h6 – textos pré-estruturados para títulos**

São tags que especificam um cabeçalho descrevendo a seção que apresenta. Há seis níveis de cabeçalhos, desde o h1 (mais importante) até h6 (menos importante). A sintaxe de HTML exige que os cabeçalhos apareçam na ordem (por exemplo, um h2 não devem preceder um h1) para uma estrutura apropriada do documento.

### **Régua e parágrafo em branco**

Digitando `<p>&nbsp;</p>` cria-se um parágrafo em branco. O código “&nbsp;”, representa um espaço em branco que não pode ser quebrado em duas linhas. Usa-se esse código para dar um espaço entre palavras que não devem aparecer separadas no caso de uma quebra de linha natural em um parágrafo, por exemplo: 10 cm, que ficaria 10&nbsp;cm.

O elemento `<hr />` cria uma régua no documento, ou seja, uma linha que acomoda-se conforme a largura da janela do navegador. Assim como a quebra de linha `<br />`, a régua também é um elemento vazio e é considerado um elemento de parágrafo.

### **Lista ordenada**

Para criar uma lista, são necessários no mínimo dois elementos: um para indicar qual o tipo de lista que está sendo utilizado (ordenada ou não ordenada) e outro para indicar os itens pertencentes a essa lista. O item lista é sempre marcado com o elemento “`<li>`” (list item – item de lista). O elemento que define o tipo de lista ordenada é “`<ol>`” (ordered list – lista ordenada). Exemplo:

```
<ol>
```

```
<li>item 1</li>
```

```
<li>item 2</li>
```



<li>item 3</li>

</ol>

### Lista não ordenada

Nesse caso o elemento “<ol>” usado na lista ordenada é substituído por “<ul>”, que é a marcação de lista não ordenada (unordered list). O item lista continua sendo indicado pelo elemento <li>.

<ul>

<li>item 1</li>

<li>item 2</li>

<li>item 3</li>

</ul>

### Imagens

O elemento usado para posicionar imagem é “<img />”, que não deve aparecer flutuando na página, ela deve estar contida em algum elemento de bloco, como, por exemplo, um parágrafo.

<p><img src=”imagem.jpg” alt=”descrição da imagem” /></p>

A linguagem (X)HTML não é capaz de criar imagens, ela as manda buscar. Isso significa que é preciso indicar no código o endereço correto dessa imagem para o navegador encontre-a e posicione-a conforme indicado. Essa função é exercida pelo atributo “src” (search – procurar).

Outro atributo essencial é o atributo “alt”, que exibe um texto alternativo à imagem. É regra que todas as imagens devem possuir o texto “alt”, pois possibilita a acessibilidade.

Existem outros atributos para a tag “img”, mas a maior parte deles deve ser resolvida através de CSS, por exemplo, os atributos “align” (alinhamento), “border” (borda), “hspace” e “vspace” (espaçamento horizontal e vertical

respectivamente). Já os atributos “width” e “height” (altura e largura) podem ser usados na própria tag <img />.

## Links

Essa é a parte mais importante do conteúdo. Os links são a alma da web. É através dele que navegamos nas marés da rede e, não raro, nos perdemos. Por isso é importante saber trabalhar com os links e deixar bem claro ao visitante de onde ele vem e para onde ele vai. A tag utilizada para acrescentar links a elementos é “<a href=”local para onde o link aponta”> </a>”.

É importante indicar o local onde o endereço será aberto. Essa tarefa é realizada pelo atributo “target”, que pode ser “\_blank” – que abre o link em uma nova janela do navegador; ou “\_self” – que abre o link na mesma janela do navegador.

## Link tipo âncora

O link tipo âncora, ou vertical, é usado para navegar em uma mesma página de cima para baixo e vice-versa. Para criar esse tipo de link é preciso criar uma âncora, que é o ponto para onde deseja ir, numa página propriamente dito.

```
<a name=”nome_da_ancora”></a>
```

Após adicionar a âncora em um local da página, é preciso agora acrescentar o link em algum elemento do conteúdo.

```
<p><a href=”nome_da_ancora”>Link</a></p>
```

## Tabelas

Todo o conteúdo de uma tabela deve ser escrito dentro dos elementos <table> e </table>. O elemento “caption” é utilizado para dar título à tabela.

```
<table>
```

```
<caption>Título da tabela</caption>
```

```
</table>
```

Elementos fundamentais - <tr>, <th> e <td>

O elemento <tr>(table row – linha de tabela) é a marcação de linha.

O elemento <th> (table heading – título de tabela) é a marcação de célula de cabeçalho.

O elemento <td> (table data – dado de tabela) é a marcação de célula (ou coluna).

Cada conjunto <tr> </tr> representa uma linha da tabela. Cada conjunto <td> </td> (ou <th> </th> para o caso de células de cabeçalho) representa uma célula da tabela.

Exemplo:

```
<table border "1">  
  
<caption>Tabela</caption>  
  
<tr>  
  
<th>Célula 1</th>  
  
<th>Célula 2</th>  
  
</tr>  
  
</table>
```

## História do HTML

Primeiro vieram os computadores interligados, no período vai do final da Segunda Guerra Mundial até a década de 1960. No começo, os motivos eram bélicos. A ideia era manter comunicação mesmo que uma parte da linha fosse cortada pelo inimigo. Interrompido o cabo de transmissão de rádio, toda a comunicação era perdida. Se houvesse um jeito de, apesar de um corte qualquer, ser mantida a comunicação... Assim mesmo com o corte, os



**UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO – UFRPE**  
**UNIDADE ACADÊMICA DE SERRA TALHADA - UAST**

computadores interligados podiam restabelecer o caminho por outras vias, porque havia outros percursos já estabelecidos. Uma teia de fios ligados uns aos outros, para que sempre houvesse comunicação.

Comunicação. Pessoas falando com pessoas, mas a base eram as máquinas. Máquinas ligadas, máquinas conectadas, cada uma falando uma coisa diferente, em idiomas que precisavam ser traduzidos o tempo todo. Em meados das décadas 1970 e 1980, as conexões haviam extrapolado as bases militares e alcançavam as universidades e centros de pesquisas. Até os cientistas, os que mais usavam toda essa tecnologia (para não dizer os únicos), estavam cansados de tanta confusão. Um, em especial: Tim Berners-Lee. Ele queria arrumar a própria bagunça. Queria poder falar com todo mundo, e de forma um pouco mais fácil. Tanto estudou que acabou unindo e criando as chamadas três tecnologias elegantes da Word Wide Web: a URL (ou URI), que é o endereço para identificar os recursos e saber onde encontrá-los; o HTTP, que é o protocolo para mover dados na web e em toda a internet; e o HTML, que é a linguagem para representar o conteúdo em termos de páginas na web e promover sua união através de links.

Comunicação. Agora era possível, e de uma forma muito mais fácil. Pessoa conversando com pessoas de forma natural. E o mais genial de tudo: o então Conseil European pour la Recherche Nucléaire (Cern) (hoje conjencido por European Organization for Nuclear Research), laboratório na Suíça em que trabalhava Tim Berners-Lee, permitiu que sua obra ficasse aberta. Ele então a publicou para que todos pudessem usar e abusar. Distribuiu. Compartilhou. E com isso a rede cresceu e hoje é o que conhecemos.

**Referências bibliográficas:**

Gomes, Ana Laura; XHTML/CSS: Criação de páginas web - São Paulo: Editora Senac São Paulo, 2010. - (Nova Série Informática)

HTML & XHTML - Guia de bolso - Jennifer Niederst Robbins