

## :: Tutorial XHTML ::

:: Tutorial XHTML :: .....	1
Objetivo.....	4
1. De onde veio XHTML?.....	4
2. Qual a finalidade do XHTML?.....	4
3. Vantagens de se usar XHTML .....	4
4. As diferenças entre XHTML e HTML .....	5
5. Todas as tags devem ser escritas em letras minúsculas.....	5
6. Os elementos (tags) devem estar convenientemente aninhados.....	6
7. Os documentos devem ser bem formados .....	6
8. O uso de tags de fechamento é obrigatório.....	6
9. Elementos vazios devem ser fechados.....	7
10. Diferenças para os atributos .....	7
10.1. Nomes de atributos .....	7
10.2. Valores de atributos .....	8
11. Sintaxe dos atributos.....	8
11.1. Os atributos id e name.....	8
11.2. O atributo lang .....	9
12. Pontos de âncoras .....	9
13. O atributo alt para imagens.....	9
14. Separadores de blocos de códigos .....	10
15. Códigos gerados por editores .....	10
16. Caracter & (ampersand).....	10
17. Elementos obrigatórios em um documento XHTML.....	11
18. Para que serve o DOCTYPE ?.....	11
18.1. Os tipos de DOCTYPE .....	11
19. Validação do documento XHTML.....	12
20. Tags de Estilo Físico.....	12
20.1. Meta .....	12
20.2. Formatando parágrafo .....	13
20.3. Headings .....	14
20.4. Superscripts e Subscripts .....	15
20.5. Elementos em linha: Marcação de Caracteres .....	15
21. Tags de Estilo Lógico .....	16
22. Ligações internas .....	18
22.1. Nome das secções .....	18
22.2. Ligações para partes de outros documentos.....	18
23. Como formatar texto <font>.....	19
23.1. Diferenças entre o HTML e o XHTML.....	19
23.2. Exemplo .....	19
23.3. Atributos Opcionais .....	20
24. Como definir cores para web.....	20
24.1. Formas de exprimir cores.....	20
24.2. Cores Oficiais.....	21
24.3. Cores com Nomes Atribuídos Oficialmente .....	21
24.4. Mais de 16 milhões de cores diferentes .....	21

25.	Como fazer alinhamento no XHTML .....	22
25.1.	O Alinhamento.....	22
25.2.	Exemplo .....	22
26.	Criação de listas ordenadas <ol>, <li> e <ul> .....	22
26.1.	Elemento <ol>.....	22
26.1.1.	Diferenças entre o HTML e o XHTML .....	22
26.1.2.	Exemplo .....	23
26.1.3.	Atributos Opcionais.....	23
26.2.	Elemento <li> .....	23
26.2.1.	Diferenças entre o HTML e o XHTML .....	23
26.2.2.	Exemplos.....	23
26.2.3.	Atributos Opcionais.....	24
26.3.	Elemento <ul>.....	24
26.3.1.	Diferenças entre o HTML e o XHTML .....	24
26.3.2.	Exemplo .....	24
26.3.3.	Atributos Opcionais.....	24
27.	Criação de listas de definição <dl> - <dt> - <dd>.....	25
27.1.	Elemento <dl>.....	25
27.1.1.	Exemplo .....	25
27.2.	Elemento <dt>.....	25
27.2.1.	Exemplo .....	25
27.3.	Elemento <dd>.....	26
27.3.1.	Exemplo .....	26
28.	Como incluir imagens <img> .....	26
28.1.	Elemento <img> .....	26
28.1.1.	Diferenças entre o HTML e o XHTML .....	26
28.1.2.	Exemplo .....	26
28.1.3.	Atributos Obrigatórios.....	26
28.1.4.	Atributos Opcionais.....	27
MÓDULO 4.....		28
29.	Criando Tabela .....	28
29.1.	Elementos <table>, <tr>, <td>.....	28
29.1.1.	Diferenças entre o HTML e o XHTML .....	28
29.1.2.	Exemplo .....	28
29.1.3.	Atributos Opcionais.....	29
29.2.	Elementos <thead>, <tbody>, <tfoot> .....	29
29.2.1.	Exemplo .....	30
29.2.2.	Atributos Opcionais.....	30
29.3.	Elemento <caption> .....	31
29.3.1.	Diferenças entre o HTML e o XHTML .....	31
29.3.2.	Exemplo .....	31
29.3.3.	Atributos Opcionais.....	31
29.4.	Elemento <th>.....	31
29.4.1.	Diferenças entre o HTML e o XHTML .....	31
29.4.2.	Exemplo .....	32
29.4.3.	Atributos Opcionais.....	34
MODULO 5.....		35
30.	Elemento <form>.....	35
30.1.	Exemplo .....	35

30.2.	Atributos Obrigatórios .....	35
30.3.	Atributos Opcionais .....	36
31.	<frame> .....	37
31.1.	Diferenças entre o HTML e o XHTML .....	37
31.2.	Exemplo .....	37
31.3.	Atributos Opcionais .....	37
32.	<iframe> .....	38
32.1.	Diferenças entre o HTML e o XHTML .....	38
32.2.	Atributos Opcionais .....	38

## Objetivo

Este tutorial tem por objetivo descrever o que é, para que serve e porque adotar a linguagem de marcação XHTML, apresentando seus aspectos estruturais e semânticos, bem como ressaltando as vantagens do seu uso.

### 1. De onde veio XHTML?

Todas as linguagens de marcação da web são baseadas em SGML, uma metalinguagem complexa, projetada para máquinas com a finalidade de servir de base para criação de outras linguagens.

O SGML foi usado para criar XML (EXtensible Markup Language), também uma metalinguagem, porém bem mais simples.

Com XML você cria suas próprias tags e atributos para escrever seu documento web. Isto significa que é você quem cria sua linguagem de marcação. XHTML foi criado dentro deste conceito e por isso é uma aplicação XML. As tags e atributos do XHTML foram criados ("inventadas") aproveitando-se as nossas conhecidas tags e atributos do HTML 4.01 e suas regras.

Conclui-se que ao usar XHTML, estamos escrevendo um código XML, onde as tags e atributos já estão definidos e isto proporciona todos os benefícios de XML sem as complicações e complexidade do SGML.

XHTML é uma linguagem de marcação bastante familiar para quem conhece HTML e a transformação de um documento existente de HTML para XHTML é uma tarefa bem simples, como veremos adiante.

### 2. Qual a finalidade do XHTML?

XHTML é a sigla em inglês para EXtensible HyperText Markup Language que em português resulta em Linguagem Extensível para Marcação de Hipertexto, uma aplicação XML, escrita para substituir o HTML e nada mais é do que um HTML "puro, claro e limpo".

### 3. Vantagens de se usar XHTML

Enumeram-se várias e efetivas vantagens de se usar XHTML nos novos documentos web produzidos ou migrar os documentos atuais escritos em HTML. E, dentre elas destaca-se em primeiro plano a compatibilidade da linguagem XHTML com as futuras aplicações de usuários, garantindo desde já que as criações XHTML conservar-se-ão estáveis por longos anos. A tendência é a de que futuras versões de browsers e agentes de usuários em geral, deixem de suportar elementos e atributos já em desuso ("deprecated") segundo as recomendações do W3C, bem assim como antigos e ultrapassados esquemas e esboços do HTML.

XHTML é a linguagem da web do futuro desde já a disposição de projetistas e desenvolvedores web.

XHTML é um código consistente que dispensa uso de "truques" e "hacks" para contornar "bugs".

Editar um código XHTML existente é uma tarefa bem simples por se tratar de uma escrita limpa e evidente.

O tempo de carregamento de uma página XHTML é mais rápido pois os browsers têm a interpretar uma página limpa sem ter que interpretar e decidir sobre renderização de erros de código.

Uma página XHTML é mais acessível aos browsers e aplicações de usuário padrão incrementando a interoperabilidade e a portabilidade dos documentos web.

Uma página XHTML é totalmente compatível com todas as aplicações de usuários para HTML, antigas e já ultrapassadas.

XHTML é uma "Web Standard"

XHTML 1.0 é uma recomendação do W3C e sua versão atual data de 26 de janeiro de 2000. Isto significa que se trata de uma linguagem estável, oficialmente especificada pelo W3C, tendo sido projetada e revisada pelos seus membros e é uma "Web Standard".

Tradução para o português da Recomendação W3C de 26 de janeiro de 2000, revisada em 1 de agosto de 2002 - XHTML™ 1.0 The Extensible HyperText Markup Language (Segunda Edição) - Uma reformulação do HTML 4 em XML 1.0

#### **4. As diferenças entre XHTML e HTML**

As principais diferenças são:

- Todas as tags devem ser escritas em letras minúsculas;
- Elementos (tags) devem estar convenientemente aninhados;
- Documentos devem ser bem formados;
- Uso de tags de fechamento é obrigatório;
- Elementos vazios devem ser fechados;
- Diferenças para os atributos.

#### **5. Todas as tags devem ser escritas em letras minúsculas**

A metalinguagem XML é case-sensitive (sensível ao tamanho da letra). De vez que XHTML é uma aplicação XML, também é case-sensitive.

Errado:

```
<DIV><P>Aqui um texto</P></DIV>
```

Certo:

```
<div><p>Aqui um texto</p></div>
```

## 6. Os elementos (tags) devem estar convenientemente aninhados

Errado:

```
<div><em><p>Aqui um texto negrito</em></p></div>
```

Certo:

```
<div><em><p>Aqui um texto negrito</p></em></div>
```

## 7. Os documentos devem ser bem formados

Um documento diz-se bem formado quando está estruturado de acordo com as regras definidas nas Recomendações para XML 1.0 [ XML ].

Todos os elementos XHTML devem estar corretamente aninhados dentro do elemento raiz <html>.

A estrutura básica do documento deve ser conforme abaixo:

```
<html>  
<head>  
...  
</head>  
<body>  
...  
</body>  
</html>
```

## 8. O uso de tags de fechamento é obrigatório

Em HTML é permitido para determinados elementos, omitir-se a tag de fechamento. XML não permite omissão de qualquer tag de fechamento. Todos os elementos, excetuados aqueles vazios declarados nas DTD, têm que ter tag de fechamento. Os elementos vazios declarados nas DTD podem ter tag de fechamento ou usar uma abreviação para elemento vazio conforme veremos adiante.

Errado::

`<p>Um parágrafo.<p>Outro parágrafo.`

Certo:

`<p>Um parágrafo.</p><p>Outro parágrafo.</p>`

## 9. Elementos vazios devem ser fechados

Elementos vazios devem ter uma tag de fechamento ou a tag de abertura deve terminar com `/>`. Como exemplo, `<br />` ou `<hr></hr>`.

Errado: Elementos vazios sem terminação

`<br>`  
`<hr>`  
``

Certo: Elementos vazios com terminação

`<br />`  
`<hr />`  
``

## 10. Diferenças para os atributos

### 10.1. Nomes de atributos

Assim como as tags, os atributos também são case-sensitive então se deve escrever nomes de atributos em minúsculas;

Errado:

`<td ROWSPAN="3">`

Certo:

`<td rowspan="3">`

## 10.2. Valores de atributos

Os valores de atributos devem estar entre "aspas";

Errado:

```
<td rowspan=3>
```

Certo:

```
<td rowspan="3">
```

## 11. Sintaxe dos atributos

Sintaxe para atributos deve ser escrita por completo;

Errado:

```
<input checked />
```

Certo:

```
<input checked="checked" />
```

Abaixo uma relação dos atributos que se enquadram nesta recomendação

compact	compact="compact"
checked	checked="checked"
declare	declare="declare"
readonly	readonly="readonly"
disabled	disabled="disabled"
selected	selected="selected"
defer	defer="defer"
ismap	ismap="ismap"
nohref	nohref="nohref"
noshade	noshade="noshade"
nowrap	nowrap="nowrap"
multiple	multiple="multiple"
noresize	noresize="noresize"

### 11.1. Os atributos id e name

HTML 4 define o atributo *name* para os elementos a, applet, form, frame, iframe, img, e map. HTML 4.0 também introduziu o atributo *id*. Ambos estes atributos foram projetados para serem usados como identificadores.



Em XML, os identificadores são do tipo ID, e poderá existir somente um atributo do tipo ID por elemento. Assim, em XHTML 1.0 o atributo id é definido como do tipo ID. Documentos XHTML 1.0 compatíveis com XML e bem estruturados, DEVEM usar o atributo id ao definir identificadores para os elementos listados acima.

Notar que em XHTML 1.0, o atributo name destes elementos está formalmente em desuso e será excluído nas versões futuras de XHTML.

Errado:

```

```

Certo:

```

```

Nota: Por razões de compatibilidade com browsers antigos você pode usar ambos os atributos como abaixo:

```

```

### 11.2. O atributo lang

Use o atributo lang para definir a língua em que foi escrito o documento e o atributo xml:lang para definir a língua em que foi escrita a versão do XML

## 12. Pontos de âncoras

Em HTML para criar um ponto de âncora, associamos um nome ao elemento <a>:

```
<p><a name="topo" >Início</a > do parágrafo..bla...</p>
```

Em XHTML adicione o atributo id:

```
<p><a id="topo" name="topo" >Início</a> do parágrafo..bla...</p>
```

## 13. O atributo alt para imagens

Em XHTML o uso do atributo alt para imagens é obrigatório

```

```

Se tratar-se de uma imagem decorativa pode-se usar o atributo alt vazio:

```

```

## 14. Separadores de blocos de códigos

É comum o uso de uma seqüência de caracteres dentro da marcação de comentários ( `<!-- -->` ) para visualmente separar trechos do código com a finalidade de facilitar sua posterior leitura e manutenção.

Não use a clássica seqüência de caracteres `-----`, para conseguir este efeito.

Alguns agentes de usuário mais antigos podem ter dificuldades na interpretação desta seqüência. Use por exemplo a seqüência `====` . ou `xxxxxx`

Errado:

```
<!-- Aqui começa o menu -->
<!-- ----- -->
código XHTML do menu
<!-- ----- -->
```

Certo:

```
<!-- Aqui começa o menu -->
<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->
código XHTML do menu
<!-- xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx -->
```

## 15. Códigos gerados por editores

Cuidado com os códigos gerados por editores!

Este é um código gerado por editor: `onMouseOver=function()` não válido em XHTML

Errado:

```
onMouseOver=function()
```

Certo:

```
onmouseover=function()
```

## 16. Caracter & (ampersand)

Codifique o & ( e comercial)

Errado:

```
Comércio & Exportação
```

Certo:

```
Comércio & Exportação
```

## 17. Elementos obrigatórios em um documento XHTML

É obrigatória a declaração do DOCTYPE assim como a existências dos elementos <html> <head> <title> e <body>

Um modelo mínimo de documento XHTML é conforme abaixo:

```
<!DOCTYPE bla..bla..bla>  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<title>Título do documento</title>  
</head>  
<body>  
Conteúdo do documento  
</body>  
</html>
```

A declaração DOCTYPE não faz parte da marcação XHTML e como tal não é também um elemento, razão pela qual não há necessidade de tag de fechamento.

## 18. Para que serve o DOCTYPE ?

A Definição do tipo de documento (Document Type Definitions DTD) especifica qual é a sintaxe SGML usada no documento. A DTD é usada pelas aplicações SGML ( tais como HTML ) para identificar as regras que se aplicam a linguagem de marcação usada no documento bem como o conjunto de elementos e entidades válidas naquela linguagem. Assim uma DTD para um documento XHTML descreve com precisão a sintaxe e a gramática da linguagem de marcação XHTML.

O DOCTYPE deve ser sempre a primeira declaração em um documento web.

### 18.1. Os tipos de DOCTYPE

São três os tipos de DOCTYPE para XHTML:

- STRICT
- TRANSITIONAL
- FRAMESET

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

Esta é a mais rígida das declarações. Os documentos XHTML no modo Strict não admitem qualquer item de formatação dentro dos elementos e nem elementos em desuso

"deprecated" segundo as recomendações do W3C. São indicados para uso com folhas de estilo em cascata, com marcação totalmente independente da apresentação

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

Esta declaração permite uma maior flexibilidade e é indicada para documentos que ainda utilizam elementos em desuso (“deprecated”), regra de apresentação embutida em tags e também para documentos destinados a exibição em browsers sem suporte para CSS. Não admite qualquer tipo de marcação para frames.

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

Esta declaração permite tudo da declaração transacional e mais os elementos específicos para frames.

## 19. Validação do documento XHTML

W3C disponibiliza um validador gratuito para documentos XHTML. Ali, você digita a URL ou o caminho para o arquivo no seu HD e um robô analisa o documento fornecendo um relatório completo e detalhado das não conformidades porventura existentes. É uma ferramenta excelente para você usar durante a elaboração ou migração do seu documento para XHTML. Serve como um verdadeiro revisor do código que você cria. Abaixo o link para o validador:

Validador XHTML do W3C : <http://validator.w3.org/>

## 20. Tags de Estilo Físico

### 20.1. Meta

A tag <meta> é utilizada para dar informações adicionais a sua página, tal como descrição, palavras chaves para busca, taxas de releitura, linguagem, mapa de caracteres. Deve ser colocada na tag <head>. Exemplos:

- Define palavras-chaves para buscas: <meta name="keywords" content="HTML, DHTML, CSS, XML, XHTML, JavaScript, VBScript" />
- Define a descrição de sua página web: <meta name="description" content="Free Web tutorials on HTML, CSS, XML, and XHTML" />
- Re-leitura da página a cada 5 s: <meta http-equiv="refresh" content="5" />
- Aceita caracteres acentuados: <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
- Descreve a linguagem da página <meta name="language" content="pt-br" />

## 20.2. Formatando parágrafo

A tag parágrafo (<p>) é utilizada para delimitar inteiramente o parágrafo de um texto. Por exemplo, o seguinte código HTML resulta em dois parágrafos na página:

```
<p>O castelo fica em Bruxelas. O castelo é do século XII. O castelo possui duas torres. O castelo possui uma ponte suspensa.</p>  
<p>Este é outro parágrafo.</p>
```

Como várias tags, você pode definir vários elementos de formatação. Em substituição ao atributo (align) a formatação é feita através da folha de estilo (style sheets). Segue um exemplo do uso, veja a Figura 1:

```
<html>  
<head>  
  <style type="text/css">  
    p.center {text-align:center}  
  </style>  
  <title>Texto Centralizado</title>  
</head>  
<body>  
  <p class="center">  
    O castelo fica em Bruxelas. O castelo é do século XII. O castelo possui duas torres.  
  </p>  
</body>  
</html>
```

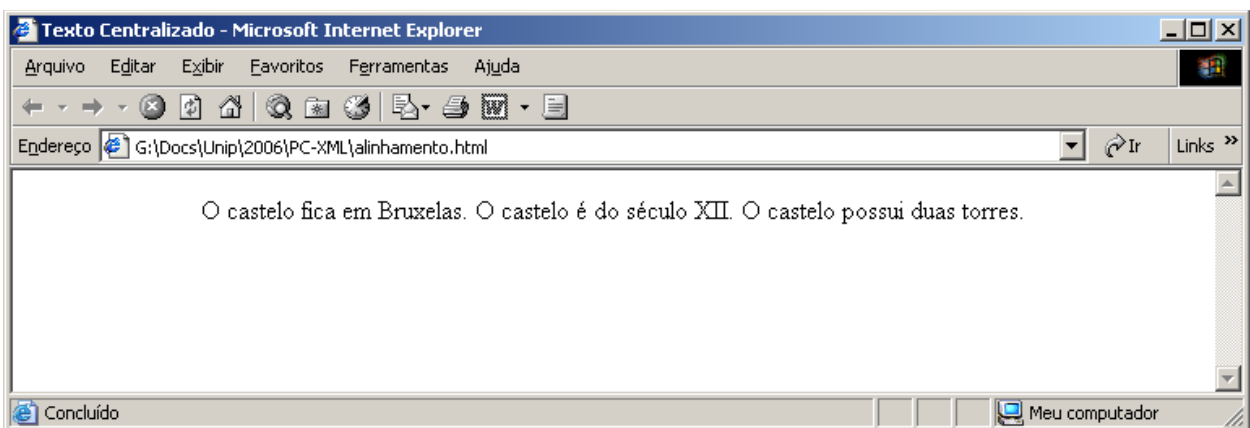


Figura 1: Alinhamento

## 20.3. Headings

HTML suporta seis níveis de cabeçalhos. Cada cabeçalho utiliza estilo de formatação de caracteres large, frequentemente bold para identificá-los como um cabeçalho. Veja o seguinte exemplo, veja também a Figura 2:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
```

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

```
<html>
```

```
<body>
```

```
<h1>Heading 1</h1>
```

```
<h2>Heading 2</h2>
```

```
<h3>Heading 3</h3>
```

```
<h4>Heading 4</h4>
```

```
<h5>Heading 5</h5>
```

```
<h6>Heading 6</h6>
```

```
<p>O castelo fica em Bruxelas. O castelo é do século XII. O castelo possui duas torres. O castelo possui uma ponte suspensa.</p>
```

```
</body>
```

```
</html>
```

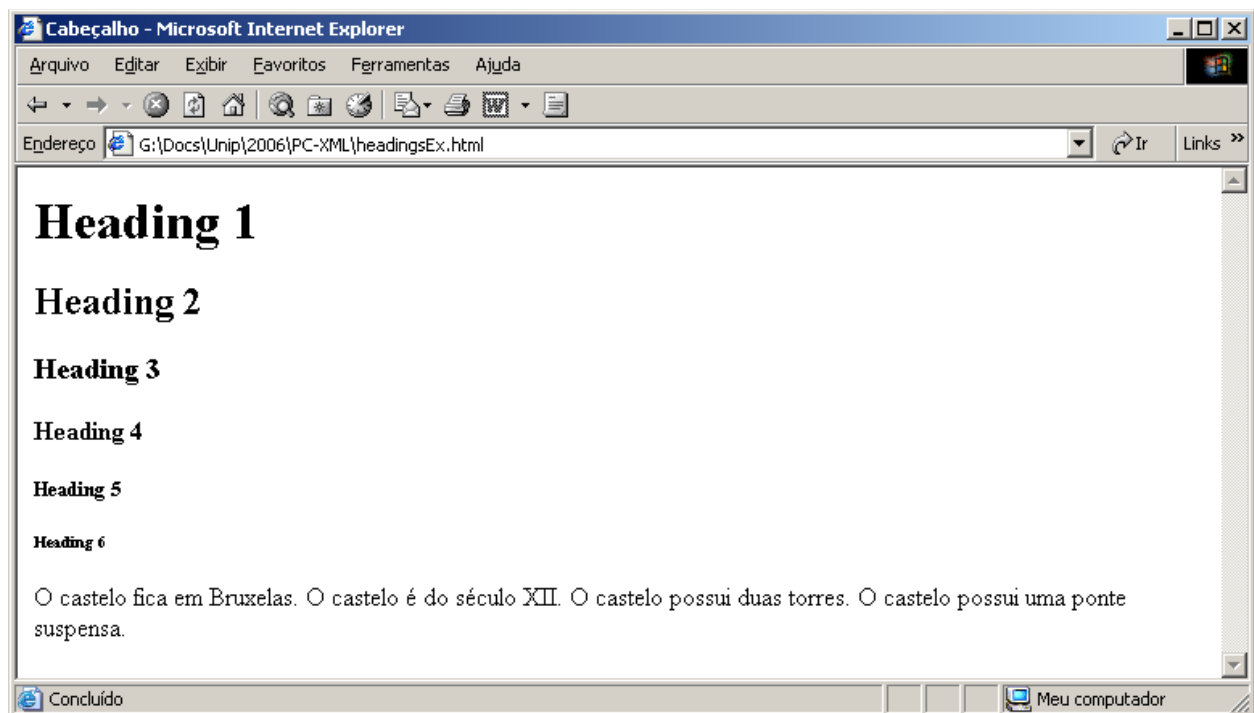


Figura 2:Exemplo de Cabeçalho

Os seis níveis começa pelo nível 1, maior, mais importante e vai até o nível 6, o menor, menos importante. Embora existam seis níveis de cabeçalho, você provavelmente utilizará somente três ou quatro deles. Você não precisa utilizar o <h1> e <h2> para utilizar o <h3>.

## 20.4. Superscripts e Subscripts

Existem duas tags, `<sup>` e `<sub>`, para formatar de texto superscript e subscript. O código seguinte mostra um exemplo de cada tag:

```
<p>Este é um texto normal.</p>
<p>Este é o 16.<sup>o</sup> dia do mês.</p>
<p>Tanques de água são marcados com H<sub>2</sub>O.</p>
```

Subscripts são letras ou números que aparecem menores que o resto do texto do documento e abaixo da linha tal como 2 em H<sub>2</sub>O.

Superscripts são também menores que o restante do texto e aparece no topo da linha, tal como “o” em 16.<sup>o</sup> de março.

Para marcar subscripts e superscripts em XHTML, você utilize o `<sub>` e `<sup>` elementos respectivamente.

## 20.5. Elementos em linha: Marcação de Caracteres

O nível mais simples de marcação em HTML é o nível de caráter, como em um editor de texto, vocês modificam a formatação dos caracteres individualmente. Esta seção apresenta a formatação básica inline:

- Bold (`<b>`)
- Italic (`<i>`)
- Big text (`<big>`)
- Small text (`<small>`)
- Teletype (monospaced) text (`<tt>`)

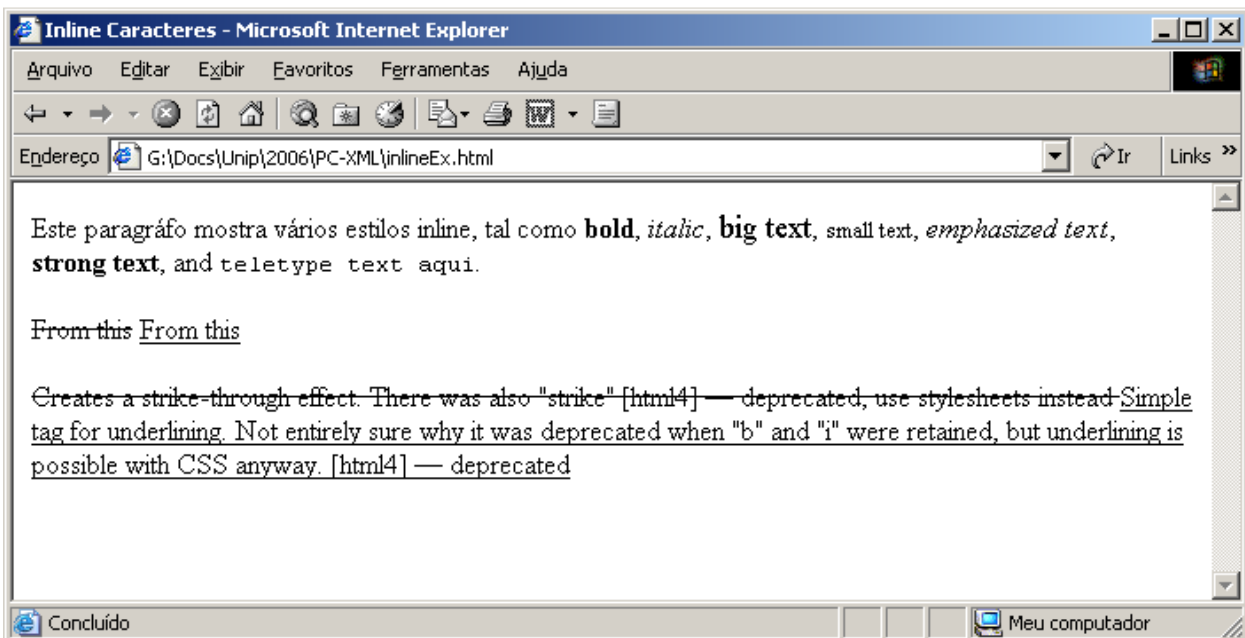
Por exemplo, considere o seguinte parágrafo, veja Figura 3:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<title>Inline Caracteres</title>
<body>
  <p>Este parágrafo mostra vários estilos inline, tal como <b>bold</b>, <i>italic</i>,
<big>big text</big>, <small>small text</small>, <em>emphasized text</em>, <strong>strong
text</strong>, and <tt>teletype text aqui</tt>.</p>
  <p><del>From this</del> <ins>From this</ins></p>
  <p>
    <s>
      Creates a strike-through effect. There was also "strike"
      [html4] — deprecated, use stylesheets instead
    </s>
    <u>
      Simple tag for underlining. Not entirely sure why it was deprecated when "b" and "i"
      were retained, but underlining is possible with CSS anyway.
      [html4] — deprecated
    </u>
  </p>
</body>
</html>
```

Observe que várias tags inline, tal como *strikethrough* (<s>) e *underline* (<u>) tags, estão em desuso na especificação corrente. Mesmo a tag de fonte (<font>) está em desuso em favor do estilo *spanning* (veremos mais tarde). Como para as tags *strikethrough* e *underline*, que foram substituídas por *delete* (<del>) e *insert* (<ins>), que são utilizadas para revisão de texto (*delete* para ignorar texto, *insert* para inserir texto).

## 21. Tags de Estilo Lógico

- <em> cria ênfase, e possui aparência de texto itálico. Equivale ao <i>.
- <strong> cria forte ênfase, e possui aparência de texto negrito. Equivale ao <b>.
- <code> é destinado a apresentar exemplos de código de programas de computador, utilize fonte mono-spaced. Equivale ao <tt>.
- <blockquote> é uma tag de obstrução que seja usado incluir citações multi-linha de outras fontes. É indicado geralmente como recortado de ambos os lados.
- <cite> é usado incluir o título de um trabalho a que esteja sendo consultado atualmente. É indicado geralmente como texto itálico.
- <q> é uma citação curta de uma outra fonte. Os browsers modernos indicarão o texto contido com as marcas da citação adicionadas em ambos os lados.
- <address> envolvido na informação de contato, incluindo endereço eletrônico.
- <var> marca nome de variáveis. Utilizado para escrever termos técnicos relacionados com códigos de programação
- <samp> é utilizado para mostrar a saída de algum código.



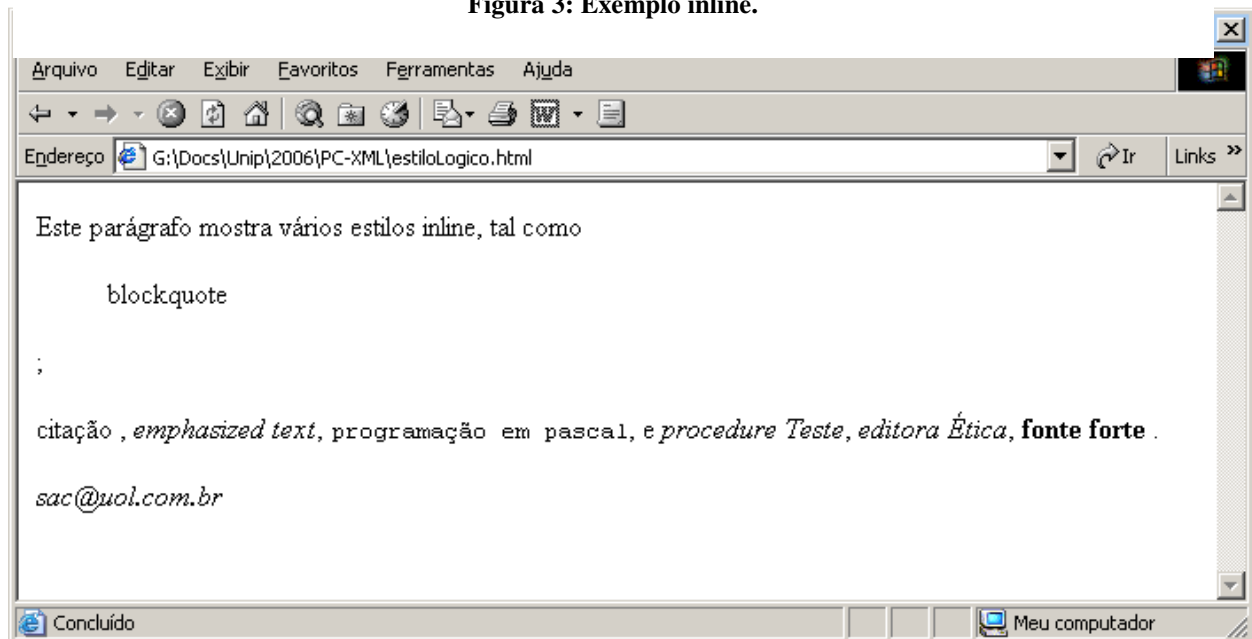


Por exemplo, considere o seguinte parágrafo:

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

<html>
  <title>Estilo Lógico</title>
  <body>
    <p>Este parágrafo mostra vários estilos inline, tal como</p>
    <blockquote>blockquote</blockquote>;
    <p><q>citação </q>, <em>emphasized text</em>, <samp>programação em pascal</samp>,
    e <var>procedure Teste</var>, <cite>editora Ética</cite>, <strong>fonte forte
    </strong>.</p>
    <address>sac@uol.com.br</address>
  </body>
</html>
```

**Figura 3: Exemplo inline.**



**Figura 4: Página de estilo lógico.**

## 22. Ligações internas

Em vez de ter que recorrer à tarefa árdua de rolar-se páginas longas, você pode fazer seus leitores muito felizes oferecendo lhes saltos da página como uma modalidade alternativa do transporte em torno de seu local. Enquanto os povos se tornaram mais preguiçosos, os saltos da página levantaram-se na popularidade, assim que evitar ter seu local não visitado pela juventude de hoje, executa estes DAE (dispositivo automático de entrada) navegacionais simples.

```
<a href="#section">link</a>
<p id="section" name="section"></p>
```

Desde que todas as tags possam utilizar o atributo `id`, você pode ligação para qualquer elemento da sua página. O mais útil é se você adicionar a ligação a um título ou a um parágrafo específico.

### 22.1. Nome das secções

A melhor coisa sobre saltos em páginas, é que eles são fáceis de fazer. Basicamente, eles são ligações (links), eles utilizam a mesma tag `<a>`, mas ligações que ligam partes do documento são feitas nomeando partes da sua página, e então fazendo a ligação referenciar esta secção. Saltos na página são feitos utilizando o atributo `name` da tag `a`. Assim, você deve utilizar `<a id="top" name="top"></a>` próximo do topo do seu documento entre as tags `<body>`. Então você coloca no lugar que você que colocar a ligação `<a href="#top">link to top</a>`.

Observe que (`#`), diz ao browser que é uma secção de uma página que será procurada, e não uma página ou um diretório. Lembre-se sempre, `#` vai ao href e NÃO no name.

### 22.2. Ligações para partes de outros documentos

Isto é o melhor. Você pode atingir secções específicas de outras paginas adicionando `#name` no final do href,

```
<a href="http://www.yourhtmlsource.com/
text/internallinks.html#top"></a>
```

## 23. Como formatar texto <font>.

O elemento <font> especifica o tipo de letra, o tamanho e a cor.

### 23.1. Diferenças entre o HTML e o XHTML

- O elemento font foi desaprovado em HTML 4.01.
- O elemento font não é suportado nas DTDs para HTML 4.01 Strict e XHTML 1.0 Strict.

### 23.2. Exemplo

#### Código Fonte

```
<font size="3" color="red">
  Isto é algum texto
</font>
<font size="1" color="blue">
  Isto é algum texto
</font>
<font face="arial" color="red">
  Isto é algum texto
</font>
```

#### Output

Isto é algum texto Isto é algum texto Isto é algum texto

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "DTD/xhtml1-transitional.dtd">
<html>
  <head>
    <title>Fonte Desaprovada</title>
  </head>
  <body>
    <font size="1" face="Helvetica,Arial,Verdana,sans-serif" color="blue" lang="en-US" class=".small"
  style="font-family: Times,serif;" id="confusingElement" >This is an HTML deprecated element!</font>
  </body>
</html>
```

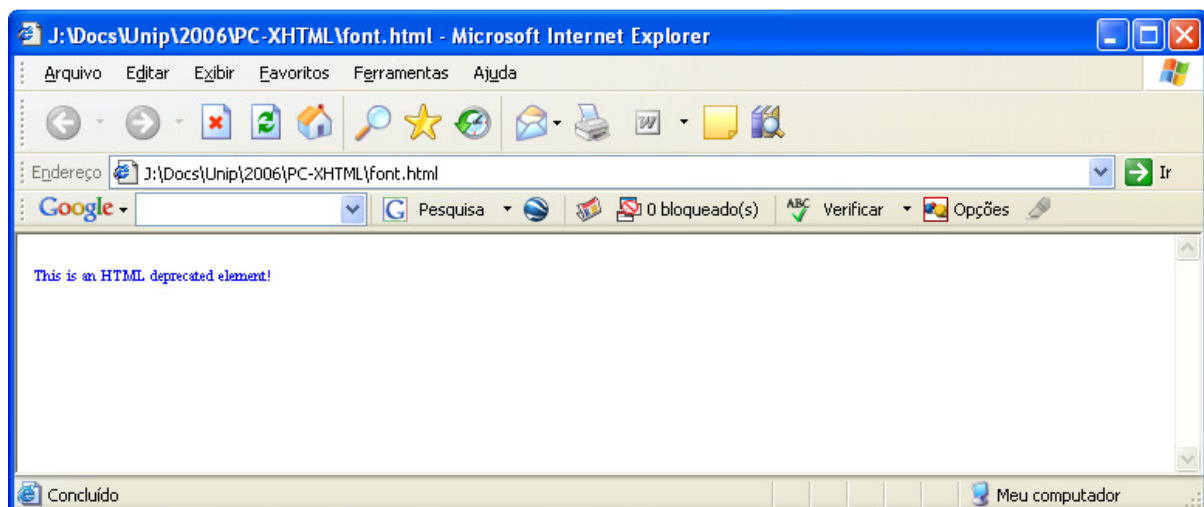


Figura 5: Exemplo de fonte.

### 23.3. Atributos Opcionais

A coluna **DTD** indica as DTDs que aceitam o atributo: S=Strict, T=Transitional, e F=Frameset.

Atributo	Valor	Descrição	DTD
color	rgb(x,x,x) #xxxxxx <i>nomeDeCor</i>	Define a cor do texto. Desaprovado. Use TF CSS.	
Face	list_of_fontnames	Define o tipo de letra a usar. Desaprovado. TF Use CSS.	
Size	Um número entre 1 e 7. Define o tamanho das letras. Desaprovado. TF Se o elemento <basefont> definir um valor por omissão este número poderá situar-se entre -6 e 6	Use CSS.	



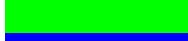





## 24. Como definir cores para web

Podemos obter qualquer cor à nossa escolha combinando em proporções corretas três cores base: vermelho (*red*), verde (*green*) e azul (*blue*).

### 24.1. Formas de exprimir cores

É recomendada para exprimir cores baseia-se em notação hexadecimal. Nesta forma as cores exprimem-se usando três números hexadecimais que definem as quantidades de vermelho, verde e azul, que entram na composição de uma determinada cor. O valor mais baixo de uma determinada cor é 0 (#00 na notação hexadecimal usada em CSS) e o valor mais alto é 255 (#FF em notação hexadecimal.) Assim, a cor preta tem 0 vermelho, 0 verde e 0 azul que se escreve na forma #000000. Já o branco tem 255 vermelho, 255 verde e 255 azul que se escreve como #FFFFFF. O amarelo forte tem 255 de vermelho, 255 de verde e zero de azul e escreve-se como #FFFF00.

A tabela seguinte mostra os resultados de diversas combinações de cores:

Cor	Código Hexadecimal	Valor RGB (decimal)
	#000000	rgb(0,0,0)
	#FF0000	rgb(255,0,0)
	#00FF00	rgb(0,255,0)
	#0000FF	rgb(0,0,255)
	#FFFF00	rgb(255,255,0)
	#00FFFF	rgb(0,255,255)
	#FF00FF	rgb(255,0,255)
	#C0C0C0	rgb(192,192,192)
	#FFFFFF	rgb(255,255,255)




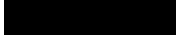



## 24.2. Cores Oficiais

A tabela seguinte mostra as 16 cores cujos nomes foram definidos oficialmente pelo W3C. Todos os browsers reconhecem estes nomes pelo que pode usá-los sem qualquer problema:

## 24.3. Cores com Nomes Atribuídos Oficialmente

<b>Aqua</b> (#00FFFF)	<b>Black</b> (#000000)	<b>Blue</b> (#0000FF)	<b>Fuchsia</b> (#FF00FF)
<b>Green</b> (#008000)	<b>Gray</b> (#808080)	<b>Lime</b> (#00FF00)	<b>Maroon</b> (#800000)
<b>Navy</b> (#000080)	<b>Olive</b> (#808000)	<b>Purple</b> (#800080)	<b>Red</b> (#FF0000)
<b>Silver</b> (#C0C0C0)	<b>Teal</b> (#008080)	<b>White</b> (#FFFFFF)	<b>Yellow</b> (#FFFF00)

Apesar de não estarem definidos oficialmente, os nomes de cor apresentados a seguir são reconhecidos por todos os browsers relevantes com exceção das versões muito antigas.

Cor	Código Hexadecimal	Nome da Cor
	#F0F8FF	AliceBlue
	#FAEBD7	AntiqueWhite
	#7FFFD4	Aquamarine
	#000000	Black
	#0000FF	Blue
	#8A2BE2	BlueViolet
	#A52A2A	Brown

## 24.4. Mais de 16 milhões de cores diferentes

Combinando as 256 quantidades possíveis de vermelho com as 256 quantidades possíveis de verde e as 256 quantidades possíveis de azul conseguimos criar mais de 16 milhões de cores diferentes (256×256×256).

Praticamente todos os monitores de computador modernos estão preparados para apresentar mais de 16 milhões de cores diferentes. No entanto, é preciso ter em conta que os novos terminais móveis a cores que existem no mercado (em telefones móveis e em PDAs) em regra possuem paletas mais reduzidas. Alguns mostram apenas 256 cores, outros 4096 ou 65536.

## 25. Como fazer alinhamento no XHTML

### 25.1. O Alinhamento

O Alinhamento do XHTML é importante para o bom entendimento do código utilizado, para isso, precisamos fazer o alinhamento das Tag's de forma a tornar o código inteligível e de forma a fazer a abertura e fechamentos da Tag's de forma correta e respeitando a rigidez exigida pelo XHTML.

### 25.2. Exemplo

```
<?xml-stylesheet href="http://www.w3.org/StyleSheets/TR/W3C-REC.css"
type="text/css"?>
<?xml-stylesheet href="#internalStyle" type="text/css"?>
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>An internal stylesheet example</title>
<style type="text/css" id="internalStyle">
    code {
        color: green;
        font-family: monospace;
        font-weight: bold;
    }
</style>
</head>
<body>
<p>
    This is text that uses our
    <code>internal stylesheet</code>.
</p>
</body>
</html>
```

## 26. Criação de listas ordenadas <ol>, <li> e <ul>

### 26.1. Elemento <ol>

O elemento <ol> contém os itens de uma lista ordenada ("ordered list").

#### 26.1.1. Diferenças entre o HTML e o XHTML

- Em HTML 4.01 os atributos compact, start e type foram desaprovados no elemento <ol>. Use CSS.
- Os atributos compact, start e type no elemento <ol> não são suportados nas DTDs para HTML 4.01 Strict e XHTML 1.0 Strict.

## 26.1.2. Exemplo

Código Fonte	Output
<code>&lt;ol&gt;</code>	1. Rum
<code>&lt;li&gt;Rum&lt;/li&gt;</code>	2. Bagaceira
<code>&lt;li&gt;Bagaceira&lt;/li&gt;</code>	
<code>&lt;/ol&gt;</code>	

## 26.1.3. Atributos Opcionais

A coluna **DTD** indica as DTDs que aceitam o atributo: S=Strict, T=Transitional, e F=Frameset.

Atributo	Valor	Descrição	DTD
compact	compact_rendering	Desaprovado. Use CSS.	TF
start	start_on_number	Especifica o valor em que se inicia a numeração dos itens. Desaprovado. Use CSS.	TF
type	A a I i 1	Especifica o tipo de lista. Desaprovado. Use CSS.	TF

## 26.2. Elemento <li>

O elemento <li> define um item numa lista com vários itens. Este elemento é usado dentro de listas ordenadas (<ol>) e listas não ordenadas (<ul>).

### 26.2.1. Diferenças entre o HTML e o XHTML

- Em HTML 4.01 os atributos type e value foram desaprovados no elemento <li>. Use CSS.
- Os atributos type e value no elemento <li> não são suportados nas DTDs para HTML 4.01 Strict e XHTML 1.0 Strict.

### 26.2.2. Exemplos

Código Fonte	Output
<code>&lt;ol&gt;</code>	1. Rum
<code>&lt;li&gt;Rum&lt;/li&gt;</code>	2. Bagaceira
<code>&lt;li&gt;Bagaceira&lt;/li&gt;</code>	
<code>&lt;/ol&gt;</code>	
<code>&lt;ul&gt;</code>	• Rum
<code>&lt;li&gt;Rum&lt;/li&gt;</code>	• Bagaceira
<code>&lt;li&gt;Bagaceira&lt;/li&gt;</code>	
<code>&lt;/ul&gt;</code>	

## 26.2.3. Atributos Opcionais

A coluna **DTD** indica as DTDs que aceitam o atributo: S=Strict, T=Transitional, e F=Frameset.

Atributo	Valor	Descrição	DTD
type	A a I i 1 disc square circle	Especifica o tipo de lista. Desaprovado. Use CSS.	TF
value	número_de_ordem_do_item	Desaprovado. Use CSS.	TF

## 26.3. Elemento <ul>

O elemento <ul> define uma lista não ordenada ("unordered list").

### 26.3.1. Diferenças entre o HTML e o XHTML

- Em HTML 4.01 os atributos compact e type foram desaprovados no elemento <ul>. Use CSS.
- Os atributos compact e type no elemento <ul> não são suportados nas DTDs para HTML 4.01 Strict e XHTML 1.0 Strict.

### 26.3.2. Exemplo

Código Fonte	Output
<ul>	• Rum
<li>Rum</li>	• Bagaceira
<li>Bagaceira</li>	
</ul>	

### 26.3.3. Atributos Opcionais

A coluna **DTD** indica as DTDs que aceitam o atributo: S=Strict, T=Transitional, e F=Frameset.

Atributo	Valor	Descrição	DTD
compact	compact_rendering	Desaprovado. Use CSS.	TF
type	disc square circle	Especifica o tipo de lista. Desaprovado. Use CSS.	TF



## 27. Criação de listas de definição <dl> - <dt> - <dd>

### 27.1. Elemento <dl>

O elemento <dl> cria lista de definições.

#### 27.1.1. Exemplo

Código Fonte	Output
<dl>	Rum
<dt>Rum</dt>	Bebida espirituosa muito apreciada pelos piratas das Caraíbas
<dd>Bebida espirituosa muito apreciada	Bagaceira
pelos piratas das Caraíbas</dd>	Bebida portuguesa com alto teor alcoólico
<dt>Bagaceira</dt>	
<dd>Bebida portuguesa com alto teor	
alcoólico</dd>	
</dl>	

### 27.2. Elemento <dt>

O elemento <dt> contém a definição de um termo numa lista de definições.

#### 27.2.1. Exemplo

Código Fonte	Output
<dl>	Rum
<dt>Rum</dt>	Bebida espirituosa muito apreciada pelos piratas das Caraíbas
<dd>Bebida espirituosa muito apreciada	Bagaceira
pelos piratas das Caraíbas</dd>	Bebida portuguesa com alto teor alcoólico
<dt>Bagaceira</dt>	
<dd>Bebida portuguesa com alto teor	
alcoólico</dd>	
</dl>	

### 27.3. Elemento <dd>

O elemento <dd> serve para apresentar descrições de termos em listas de definições.

#### 27.3.1. Exemplo

Código Fonte	Output
<pre>&lt;dl&gt; &lt;dt&gt;Rum&lt;/dt&gt; &lt;dd&gt;Bebida espirituosa muito apreciada pelos piratas das Caraíbas&lt;/dd&gt; &lt;dt&gt;Bagaceira&lt;/dt&gt; &lt;dd&gt;Bebida portuguesa com alto teor alcoólico e extraordinários poderes medicinais&lt;/dd&gt; &lt;/dl&gt;</pre>	<pre>Rum Bebida espirituosa muito apreciada pelos piratas das Caraíbas Bagaceira Bebida portuguesa com alto teor alcoólico e extraordinários poderes medicinais</pre>

## 28. Como incluir imagens <img>


### 28.1. Elemento <img>

O elemento <img> insere uma imagem no documento.

#### 28.1.1. Diferenças entre o HTML e o XHTML

- Este é um elemento vazio. Em HTML ele não possui etiqueta de fecho.
- Em XHTML o elemento <img> deve ser fechado.
- Em HTML 4.01 os atributos align, border, hspace e vspace foram desaprovaados no elemento <img>. Use CSS.
- Os atributos align, border, hspace e vspace do elemento <img> não são suportados nas DTDs para HTML 4.01 Strict e XHTML 1.0 Strict.

#### 28.1.2. Exemplo

Código Fonte	Output
<pre>&lt;img src="ext/magoo.gif" alt="O Senhor Magoo"/&gt;</pre>	

#### 28.1.3. Atributos Obrigatórios

A coluna **DTD** indica as DTDs que aceitam o atributo: S=Strict, T=Transitional, e F=Frameset.

Atributo	Valor	Descrição	DTD
alt	text	Define uma curta descrição da imagem	STF
src	URL	A URL da imagem é mostrada	STF

### 28.1.4. Atributos Opcionais

Atributo	Valor	Descrição	DTD
align	top bottom middle left right	Como alinhar a imagem relativamente ao texto envolvente. Desaprovado. Use CSS.	TF
border	pixels	Define a linha de fronteira a desenhar em redor da imagem. Desaprovado. Use CSS.	TF
height	pixels %	Define a altura da imagem	STF
hspace	pixels	Especifica a quantidade de espaço em branco a inserir à esquerda e a direita da imagem. Desaprovado. Use CSS.	TF
ismap	URL	Indica que a imagem tem um mapa definido sobre a sua área.	STF
longdesc	URL	URL de uma página que contém uma descrição extensa da imagem.	STF
usemap	URL	Define a imagem como tendo um mapa definido sobre si. Para perceber a utilidade disto consulte as explicações dos elementos <map> e <area>.	STF
vspace	pixels	Especifica a quantidade de espaço em branco a inserir acima e abaixo da imagem. Desaprovado. Use CSS.	TF
width	pixels %	Especifica a largura da imagem	STF

## MÓDULO 4

### 29. Criando Tabela

#### 29.1. Elementos <table>, <tr>, <td>

O elemento <table> define uma tabela. Dentro de uma tabela podemos colocar cabeçalhos, linhas, rodapés, células, parágrafos, outras tabelas e diversos elementos do HTML. O elemento <tr> insere uma linha numa tabela. O elemento <td> insere uma célula numa tabela.

##### 29.1.1. Diferenças entre o HTML e o XHTML

- Em HTML 4.01 os atributos align e bgcolor foram desaprovaados no elemento <table>. Use CSS.
- Os atributos align e bgcolor no elemento <table> não são suportados nas DTDs para HTML 4.01 Strict e XHTML 1.0 Strict.
- Os atributos "bgcolor", "height", "width", e "nowrap" do elemento <td> foram desaprovaados em HTML 4.01. Use CSS.
- Os atributos bgcolor, height, width, e nowrap do elemento <td> não são suportados nas DTDs para HTML 4.01 Strict e XHTML 1.0 Strict.
- Em HTML 4.01 o atributo bgcolor foi desaprovaado para o elemento <tr>. Use CSS.
- O atributo bgcolor no elemento <tr> não é suportado nas DTDs para HTML 4.01 Strict e XHTML 1.0 Strict.

##### 29.1.2. Exemplo

Código Fonte	Output		
<pre>&lt;table border = "1"&gt; &lt;tr&gt; &lt;td&gt;Célula A&lt;/td&gt; &lt;td&gt;Célula B&lt;/td&gt; &lt;/tr&gt; &lt;/table&gt;</pre>	<table border="1"><tr><td>Célula A</td><td>Célula B</td></tr></table>	Célula A	Célula B
Célula A	Célula B		

## 29.1.3. Atributos Opcionais

A coluna **DTD** indica as DTDs que aceitam o atributo: S=Strict, T=Transitional, e F=Frameset.

Atributo	Valor	Descrição	DTD
align	left	Alinha a tabela. Desaprovado. Use CSS.	TF
	center		
	right		
bgcolor	rgb(x,x,x)	Especifica a cor de fundo para a tabela. Desaprovado. Use CSS.	TF
	#xxxxxx <i>nomeDeCor</i>		
border	pixels	Especifica a espessura da linha de contorno.	STF
cellpadding	pixels	Especifica o espaço em branco que fica entre o conteúdo das células e a sua fronteira.	STF
	%		
cellspacing	pixels	Especifica o espaço em branco que fica entre células adjacentes.	STF
	%		
frame	void	Especifica a forma como são apresentadas as linhas de contorno.	STF
	above		
	below		
	hsides		
	lhs		
	rhs		
	vsides		
	box		
	border		
	rules		
groups			
rows			
cols			
summary	texto	Fornece um sumário da tabela para ser lido por software de síntese de voz em browsers não visuais.	STF
	all		
width	%	Especifica a largura da tabela.	STF
	pixels		

**Dica:** Faça border="0" para que não seja desenhada nenhuma linha de contorno.

**Nota:** Deve usar-se em conjunto com o atributo border.

**Nota:** Deve usar-se em conjunto com o atributo border.

## 29.2. Elementos <thead>, <tbody>, <tfoot>

Define um cabeçalho de tabela, `thead`. Define um corpo de tabela, `tbody`. Define um rodapé de tabela, `tfoot`.

Os elementos `thead`, `tfoot` e `tbody` ajudam-nos a agrupar as linhas de uma tabela. Quando criamos uma tabela podemos precisar de uma linha de cabeçalho, várias linhas de dados e de

uma linha com somas totais ao fundo. Esta forma de dividir uma tabela permite ao browser fazer o deslocamento vertical ("scroll") da página mantendo os cabeçalhos e os rodapés sempre visíveis. Quando imprimimos uma página escrita deste modo e a tabela ocupa mais do que uma página, os cabeçalhos e os rodapés podem aparecer repetidos em todas as páginas que contêm a tabela.

## 29.2.1. Exemplo

### Código Fonte

```
<table border = "1">
<thead>
<tr>
<td>Este texto está sob um elemento thead</td>
</tr>
</thead>
<tfoot>
<tr>
<td>Este texto está sob um elemento tfoot</td>
</tr>
</tfoot>
<tbody>
<tr>
<td>Este texto está sob um elemento tbody</td>
</tr>
</tbody>
</table>
```

### Output

Este texto está sob um elemento thead
Este texto está sob um elemento tfoot
Este texto está sob um elemento tbody

## 29.2.2. Atributos Opcionais

A coluna **DTD** indica as DTDs que aceitam o atributo: S=Strict, T=Transitional, e F=Frameset.

Atributo	Valor	Descrição	DTD
align	right left center justify char	Define o alinhamento do texto nas células	STF
char	carácter	Especifica o carácter a partir do qual se faz o alinhamento do texto.	STF
charoff	pixels %	Define o desvio ("offset") do alinhamento para o primeiro carácter a partir do qual se faz o alinhamento	STF
valign	top middle bottom baseline	<p><b>Nota:</b> Usa-se apenas quando também se usar align="char"</p> <p><b>Nota:</b> Usa-se apenas quando também se usar align="char"</p>	

### 29.3. Elemento <caption>

Este elemento define uma legenda para a tabela. O elemento <caption> deve vir imediatamente a seguir ao elemento <table>. É permitido definir mais do que uma legenda para uma tabela. Normamente a legenda é colocada ao centro e por cima da tabela.

#### 29.3.1. Diferenças entre o HTML e o XHTML

- Em HTML 4.01 o atributo align foi desaprovado no elemento <caption>. Use CSS.
- O atributo align no elemento <caption> não é suportado nas DTDs para HTML 4.01 Strict e XHTML 1.0 Strict.

#### 29.3.2. Exemplo

##### Código Fonte

```
<table border="1">
<caption>Isto é uma legenda</caption>
<tr>
<td>Célula 1</td>
<td>Célula 2</td>
</tr>
</table>
```

##### Output

Isto é uma legenda	
Célula 1	Célula 2

#### 29.3.3. Atributos Opcionais

A coluna **DTD** indica as DTDs que aceitam o atributo: S=Strict, T=Transitional, e F=Frameset.

Atributo	Valor	Descrição	DTD
align	left right top bottom	Como alinhar texto dentro de um elemento <caption>. Desaprovado. Use CSS.	TF

### 29.4. Elemento <th>

Insere uma célula com um cabeçalho.

#### 29.4.1. Diferenças entre o HTML e o XHTML

- Os atributos "bgcolor", "height", "width", e "nowrap" do elemento <th> foram desaprovados em HTML 4.01. Use CSS.
- Os atributos bgcolor, height, width, e nowrap do elemento <th> não são suportados nas DTDs para HTML 4.01 Strict e XHTML 1.0 Strict.

### 29.4.2. Exemplo

#### Código Fonte

```
<table border = "1">
<tr>
  <th>Cabeçalho 1</th>
  <th>Cabeçalho 2</th>
</tr>
<tr>
  <td>Cabeçalho A</td>
  <td>Cabeçalho B</td>
</tr>
</table>
```

#### Output

Cabeçalho 1	Cabeçalho 2
Célula A	Célula B

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"DTD/xhtml1-transitional.dtd">
<html>
<head>
<title>Exemplo</title>
</head>
<body>
<h4>Uma célula que ocupa duas colunas:</h4>
```

```
<table border="1">
<tbody>
<tr valign="top">
<th class="tabelaDescr">Nome</th>
<th colspan="2">Número de sócio do Belenenses</th>
</tr>
```

```
<tr valign="top">
<td>Zé das Couves</td>
<td>31</td>
<td>31</td>
</tr>
</tbody>
</table>
```

<h4>Uma célula que ocupa duas linhas:</h4>

```
<table border="1">
<tbody>
<tr valign="top">
<th class="tabelaDescr">Nome:</th>
<td>Zé das Couves</td>
</tr>
```

```
<tr valign="top">
<th rowspan="2">Número de sócio do Belenenses:</th>
```



```
<td>31</td>
</tr>
```

```
<tr valign="top">
<td>31</td>
</tr>
</tbody>
</table>
```

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

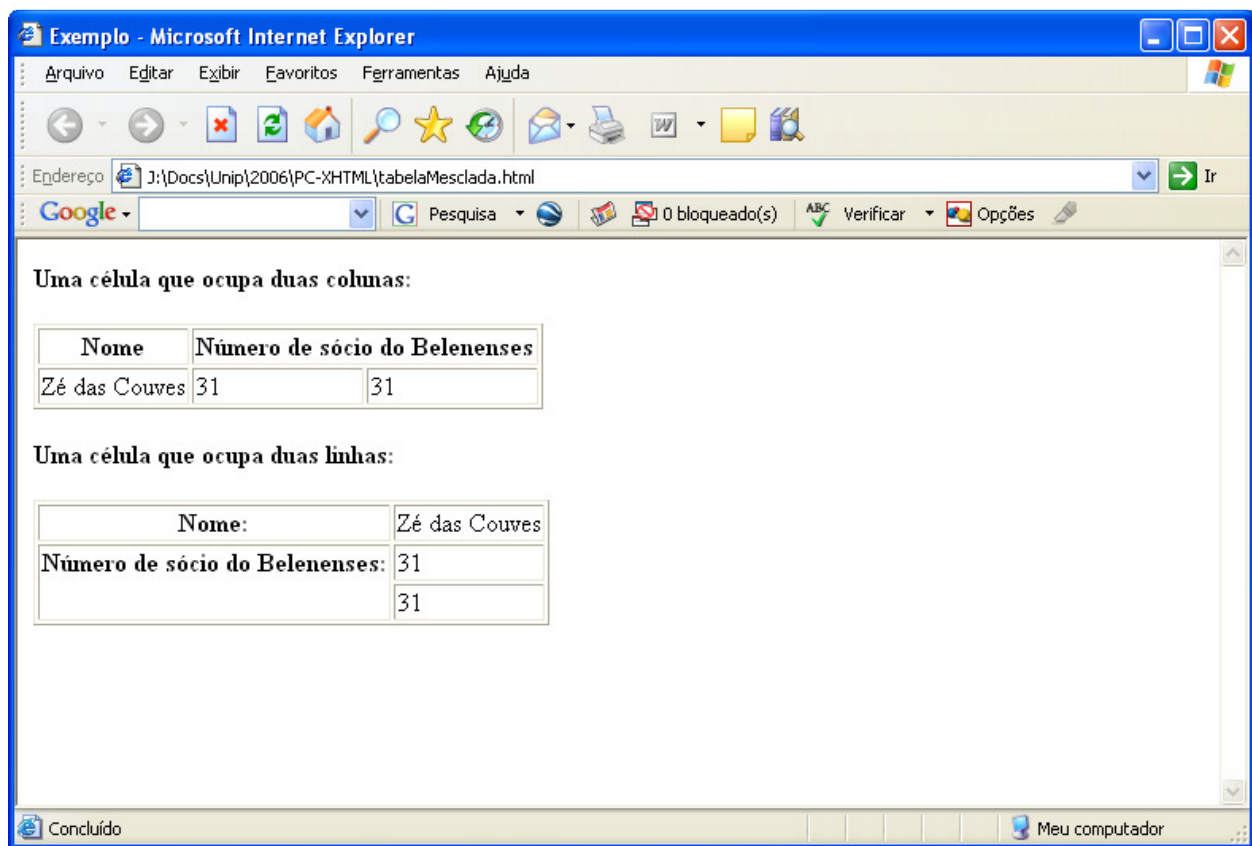


Figura 6: Tabela com coluna e linha mesclada.

## 29.4.3. Atributos Opcionais

A coluna **DTD** indica as DTDs que aceitam o atributo: S=Strict, T=Transitional, e F=Frameset.

Atributo	Valor	Descrição	DTD
abbr	abbr_text	Especifica uma forma abreviada para o conteúdo da célula	STF
align	left right center justify char	Especifica o alinhamento horizontal do conteúdo da célula	STF
axis	category_names	Atribui um nome à célula	STF
bgcolor	rgb(x,x,x) #xxxxxx <i>nomeDeCor</i>	Especifica a cor de fundo para a célula. Desaprovado. Use CSS.	TF
char	carácter	Especifica o carácter a partir do qual se faz o alinhamento do texto.	STF
		<b>Nota:</b> Usa-se apenas quando também se usar align="char"	
charoff	pixels %	Define o desvio ("offset") do alinhamento para o primeiro carácter a partir do qual se faz o alinhamento	STF
		<b>Nota:</b> Usa-se apenas quando também se usar align="char"	
colspan	número	Define quantas colunas são ocupadas pelo elemento <td>	STF
headers	header_cells_id	Uma lista de IDs separados por espaços. Os IDs fornecem informação acerca dos cabeçalhos das células e servem para ajudar os browsers apenas de texto a representar as células.	STF
height	pixels	Especifica a altura de uma célula. Desaprovado. Use CSS.	TF
nowrap	nowrap	Indica se se deve impedir que o browser insira automaticamente mudanças de linha. Desaprovado. Use CSS.	TF
rowspan	número	Define quantas linhas são ocupadas pelo elemento <td>	STF
scope	col colgroup row rowgroup	Indica se a célula fornece informação de cabeçalho relativa à linha a que pertence, ou relativa à coluna a que pertence, ou relativa ao grupo de linhas a que pertence, ou relativa ao grupo de colunas (colgroup) a que pertence.	STF
valign	top middle bottom baseline	Especifica o alinhamento vertical do conteúdo da célula	STF
width	pixels %	Especifica a largura de uma célula. Desaprovado. Use CSS.	TF

## MODULO 5

### 30. Elemento <form>

O elemento <form> cria um formulário sobre a página para recolher dados a introduzir pelo utilizador. O formulário pode conter campos de texto, caixas de validação, "radiobuttons" e outros elementos para introduzir dados. Os dados recolhidos num formulário são enviados para o servidor para processamento.

#### 30.1. Exemplo

##### Código Fonte

```
<form action="processar.html"
method="get">

Primeiro nome:
<input type="text" name="pnome"
value="Robin" />
<br />
Último nome:
<input type="text" name="unome" value="dos
Bosques" />
<br />
<input type="submit" value="Submeter" />

</form>

<p>
Ao clicar no botão "Submeter" (ou "Submit") o
conteúdo do formulário será enviado para a
página processar.html para tratamento.
</p>
```

##### Output

Primeiro nome:

Último nome:

Ao clicar no botão "Submeter" (ou "Submit") o conteúdo do formulário será enviado para a página processar.html para tratamento.

#### 30.2. Atributos Obrigatórios

A coluna **DTD** indica as [DTDs](#) que aceitam o atributo: S=Strict, T=Transitional, e F=Frameset.

Atributo	Valor	Descrição	DTD
action	URL	URL do recurso para onde vão ser enviados os dados STF quando for pressionado o botão de submissão.	STF

## 30.3. Atributos Opcionais

Atributo	Valor	Descrição	DTD
accept	lista com tipos de conteúdo	Uma lista com os tipos de conteúdo (separados por espaços) que o servidor consegue processar correctamente.	STF
accept-charset	charset_list	Uma lista com os conjuntos de caracteres (separados por espaços) que podem ser utilizados para representar os dados do formulário. O valor por omissão é "unknown".	STF
enctype	mimetype	O tipo MIME (Multipurpose Internet Mail Extensions) usado para codificar o conteúdo do formulário.	STF
method	get post	Método usado para enviar os dados para o servidor. O valor por omissão é "get".	STF

**method="get":** Este método envia o conteúdo do formulário incorporado no URL com que faz a chamada ao servidor. O aspecto do URL é algo parecido com isto: URL?var1=valor1&var2=valor2.  
**Nota:** Se o conteúdo do formulário tiver mais de 100 caracteres deve usar method="post".

**method="post":** Este método envia os dados do formulário separadamente. Para isso é gerado um segundo pedido HTTP (com o método get o envio é mais rápido pois faz -se tudo numa única operação.).

name	form_name	Define um nome único que identifica o formulário.	TF
target	_blank _self _parent _top	Especifica a janela (ou a moldura) onde se abre a ligação ao clicar. <ul style="list-style-type: none"> <li>• _blank - a ligação abre-se numa nova janela</li> <li>• _self - a ligação abre-se na própria janela que contém o elemento &lt;a&gt;</li> <li>• _parent - (só para ligações que estão numa moldura, ou "frame") a ligação abre-se no documento que contém a moldura (que terá sido criada através de um &lt;frame&gt; ou de um &lt;iframe&gt;)</li> <li>• _top - a ligação abre-se no documento que ocupa a posição mais elevada na hierarquia (aquele que não está dentro de nenhum &lt;frameset&gt; nem num &lt;iframe&gt;)</li> </ul>	TF

## 31. <frame>

Define uma subjanela ou moldura ("frame")

### 31.1. Diferenças entre o HTML e o XHTML

- Este é um elemento vazio. Em HTML ele não possui etiqueta de fecho.
- Em XHTML o elemento <frame> deve ser fechado.

### 31.2. Exemplo

#### Código Fonte

#### Output

```
<a href="molduras_1.html" target="_blank">Faça
clique aqui para ver o resultado</a>
```

#### O código contido no ficheiro

"ext/molduras\_1.html" é:

```
<html>
<frameset cols="140, *">
  <frame src ="nav_molduras.html"/>
  <frame src ="moldura0.html" name="principal"/>
</frameset>
</html>
```

### 31.3. Atributos Opcionais

A coluna **DTD** indica as [DTDs](#) que aceitam o atributo: S=Strict, T=Transitional, e F=Frameset.

Atributo	Valor	Descrição	DTD
frameborder	0 1	Indica se deve ser desenhada ou não uma linha em redor da moldura.	F
longdesc	URL	URL de uma página que contém uma descrição extensa do conteúdo da moldura. Útil para ajudar os utilizadores de browsers que não suportam molduras.	F
marginheight	pixels	Define as espessuras das margens superior e inferior da moldura	F
marginwidth	pixels	Define as espessuras das margens esquerda e direita da moldura	F
name	frame_name	Define um nome único a dar à moldura objecto (usa-se para comunicar com os scripts, que podem estar em outra moldura)	F
noresize	noresize	Quando utilizado (o valor "noresize" é o único permitido), o tamanho da moldura deixa de poder ser alterado.	F
scrolling	yes no auto	Determina se existem ou não barras de deslocamento ("scroll bars".) O valor "auto" significa automático (o browser decide se deve ou não mostrar "scroll bars").	F
src	URL	O URL da página que vai ser apresentada na moldura	F

## 32. <iframe>

O elemento iframe insere uma subjanela, ou moldura, dentro de uma página da Web ("inline frame"). A nova moldura contém a sua própria página e assemelha-se a uma moldura ("frame") criada com os elementos <frame> e <frameset>, mas está dentro de uma página normal (no meio dos seus conteúdos) em vez de estar num frameset.

### 32.1. Diferenças entre o HTML e o XHTML

- Mesmo que não tenha conteúdo, em XHTML este elemento tem obrigatoriamente de ser fechado.

### 32.2. Atributos Opcionais

A coluna **DTD** indica as DTDs que aceitam o atributo: S=Strict, T=Transitional, e F=Frameset.

Atributo	Valor	Descrição	DTD
align	left right top middle bottom	Especifica o modo como a moldura (ou subjanela) é alinhada na página que a contém.	TF
frameborder	1 0	Indica se é desenhada ou não uma linha em redor da moldura.	TF
height	pixels %	Define a altura da moldura	TF
longdesc	URL	URL de uma página que contém uma descrição extensa do conteúdo da moldura	TF
marginheight	pixels	Define as espessuras das margens superior e inferior da moldura	TF
marginwidth	pixels	Define as espessuras das margens esquerda e direita da moldura	TF
name	frame_name	Define um nome único a dar a moldura (usado para comunicar com os scripts, que podem estar em outras molduras)	TF
scrolling	yes no auto	Especifica como são apresentadas as barras de deslocamento ("scroll bars")	TF
src	URL	O URL da página que vai ser apresentada na moldura	TF
width	pixels %	Define a largura da moldura	TF