

(X)HTML

Manual de Referência

Prof. M.Sc. Ricardo Maroquio

Cachoeiro de Itapemirim, 10 de abril de 2009

Direitos Autorais

Este material foi elaborado pelo Prof. M.Sc. Ricardo Maroquio Bernardo para utilização exclusiva em cursos por ele ministrados. A utilização deste material sem prévia autorização escrita do autor viola a Lei 9.610 de 19/02/1998, e o(a) infrator(a) poderá sofrer todas as penalidades legais aplicáveis. Se desejar utilizar este material para outra finalidade senão a supracitada, entre em contato com o autor através do email *maroquio@gmail.com*.

Sumário

| | | |
|-----|--|----|
| 1 | Introdução | 7 |
| 2 | Novidades da HTML 4.01 | 8 |
| 2.1 | Novos Rótulos (<i>Tags</i>) | 8 |
| 2.2 | Separação da Apresentação e da Estrutura das Páginas | 8 |
| 2.3 | Acessibilidade | 9 |
| 2.4 | Internacionalização | 9 |
| 2.5 | Folhas de Estilo (CSS) | 9 |
| 2.6 | Execução de Scripts no Navegador | 9 |
| 2.7 | Frames | 10 |
| 2.8 | Tabelas Avançadas | 10 |
| 2.9 | Considerações Sobre o Capítulo | 10 |
| 3 | A Estrutura de um Documento HTML | 11 |
| 3.1 | Elementos e Rótulos (<i>Tags</i>) | 11 |
| 3.2 | Atributos | 12 |
| 3.3 | Caracteres Especiais | 12 |
| 3.4 | Comentários | 12 |
| 3.5 | Um Documento HTML Completo | 12 |
| 3.6 | Validando um Documento HTML | 13 |
| 3.7 | Considerações Sobre o Capítulo | 13 |
| 4 | Elementos de Alto Nível | 14 |
| 4.1 | HTML | 14 |
| 4.2 | HEAD | 14 |
| 4.3 | BODY | 15 |
| 4.4 | Considerações Sobre o Capítulo | 16 |
| 5 | Elementos de Cabeçalho (HEAD) | 17 |
| 5.1 | BASE | 17 |
| 5.2 | LINK | 17 |
| 5.3 | META | 19 |
| 5.4 | SCRIPT | 20 |
| 5.5 | STYLE | 21 |
| 5.6 | TITLE | 21 |
| 5.7 | Considerações Sobre o Capítulo | 21 |
| 6 | Elementos em Nível de Bloco | 22 |
| 6.1 | ADDRESS | 22 |
| 6.2 | BLOCKQUOTE | 22 |
| 6.3 | DEL | 23 |
| 6.4 | DIV | 23 |

| | | |
|-----------|---|-----------|
| 6.5 | H1 .. H6..... | 23 |
| 6.6 | HR..... | 24 |
| 6.7 | INS | 24 |
| 6.8 | NOSCRIPT | 25 |
| 6.9 | P..... | 25 |
| 6.10 | PRE | 25 |
| 6.11 | Considerações Sobre o Capítulo | 25 |
| 7 | Elementos Comuns em Nível de Linha | 26 |
| 7.1 | Elementos para Composição de Frases..... | 26 |
| 7.2 | Elementos para Formatação de Fonte de Letra..... | 27 |
| 7.3 | Considerações Sobre o Capítulo | 28 |
| 8 | Elementos Especiais em Nível de Linha..... | 29 |
| 8.1 | A | 29 |
| 8.2 | BDO | 30 |
| 8.3 | BR | 30 |
| 8.4 | IMG..... | 30 |
| 8.5 | MAP..... | 31 |
| 8.6 | AREA..... | 31 |
| 8.7 | OBJECT..... | 32 |
| 8.8 | PARAM | 33 |
| 8.9 | Q..... | 34 |
| 8.10 | SCRIPT | 34 |
| 8.11 | SPAN..... | 34 |
| 8.12 | SUB | 34 |
| 8.13 | SUP | 35 |
| 8.14 | Considerações Sobre o Capítulo | 35 |
| 9 | Elementos para Criação de Tabelas..... | 36 |
| 9.1 | TABLE..... | 36 |
| 9.2 | CAPTION | 37 |
| 9.3 | COLGROUP | 37 |
| 9.4 | COL | 38 |
| 9.5 | THEAD | 39 |
| 9.6 | TFOOT..... | 39 |
| 9.7 | TBODY | 40 |
| 9.8 | TR..... | 41 |
| 9.9 | TD | 42 |
| 9.10 | TH | 43 |
| 9.11 | Exemplo..... | 44 |
| 9.12 | Considerações Sobre o Capítulo | 44 |
| 10 | Elementos para Criação de Formulários..... | 45 |
| 10.1 | FORM..... | 45 |
| 10.2 | BUTTON..... | 45 |
| 10.3 | FIELDSET | 46 |

10.4 LEGEND..... 47

10.5 INPUT 47

Índice de Figuras

1 Introdução

A linguagem HTML foi concebida com o intuito de facilitar a formatação de documentos disponibilizados em meio digital, principalmente através da Internet. Atualmente, praticamente todas as páginas disponíveis nos sites fazem uso da linguagem HTML para exibir seus conteúdos.

Este material aborda de forma referencial os principais elementos da linguagem, não tendo como objetivo servir como fonte exclusiva de leitura para o aprendizado da linguagem. Por ser um manual de referência, deve ser utilizado apenas como material de apoio em cursos presenciais e como fonte de consulta para pessoas que já tem certo domínio da linguagem e de seu uso.

Até o advento da HTML, outras linguagens surgiram com um propósito semelhante, porém, não tão abrangente quanto a HTML. As primeiras linguagens de marcação tratavam apenas de conteúdo textual. Foi para suprir esta carência que a linguagem HTML foi criada. Com a HTML, pode-se criar documentos com textos e conteúdo multimídia, aumentando consideravelmente o apelo visual dos documentos.

Até a conclusão deste material, a linguagem HTML estava na versão 4.01, e uma nova linguagem chamada XHTML, derivada da HTML, tornava-se o padrão para a diagramação de conteúdo visual para a Internet.

Vale ressaltar que a linguagem HTML não é uma linguagem de programação, isto é, não possui qualquer tipo de lógica de controle. Trata-se de uma linguagem de marcação, com o propósito de definir a formatação visual de conteúdo para a Internet.

Os próximos capítulos apresentam os elementos principais da linguagem individualmente, incluindo a sintaxe de utilização, as indicações de uso e alguns exemplos simples.

2 Novidades da HTML 4.01

2.1 Novos Rótulos (*Tags*)

Muitos elementos novos (*tags*) passaram a fazer parte da HTML 4.01. A listagem a seguir apresenta estes elementos.

- ABBR - Abreviação
- ACRONYM - Acrônimo
- BDO - Substituição do BiDi
- BUTTON - Botão
- COL - Coluna de tabela
- COLGROUP - Grupo de colunas de tabela
- DEL - Texto deletado
- FIELDSET - Grupo de controles de formulário
- FRAME - Frame
- FRAMESET - Conjunto de frames
- IFRAME - Frame embutido
- INS - Texto inserido
- LABEL - Rótulo de campo de formulário
- LEGEND - Título de conjunto de campos
- NOFRAMES - Definição de conteúdo quando não há suporte a frames
- NOSCRIPT - Definição de conteúdo quando não há suporte a scripts
- OBJECT - Objeto
- OPTGROUP - Grupo de opções
- Q - Texto entre aspas
- S - Texto tachado
- SPAN - Área de conteúdo sem quebra
- TBODY - Corpo de tabela
- TFOOT - Rodapé de tabela
- THEAD - Cabeçalho de tabela

Conforme já mencionado, a linguagem HTML consiste de um conjunto de rótulos (*tags*) que definem como o conteúdo rotulado será apresentado. Os rótulos supracitados serão detalhados nos capítulos mais adiante, portanto, a não compreensão de alguns termos é completamente normal.

2.2 Separação da Apresentação e da Estrutura das Páginas

Muitos recursos da HTML 3.2 caíram em desuso ou foram simplesmente abandonados, principalmente rótulos relacionados com a apresentação, pois a HTML 4.01 estimula o uso de folhas de estilo (CSS), visando separar a parte estrutural de um documento de sua apresentação.

Através do uso de folhas de estilo para separar as características visuais de um documento de suas características estruturais, é possível tornar o documento bem mais flexível, principalmente no que diz respeito à independência de dispositivos de saída. Isso significa que um documento pode ser mostrado com uma mesma estrutura, mas com características visuais diferentes, em celulares, computadores e outros

tipos de dispositivo. A separação da apresentação visual e da estrutura de um documento também permite a alteração do visual de um documento inteiro apenas alterando uma folha de estilos, facilitando muito a manutenção dos sites.

2.3 Acessibilidade

Grande parte das melhorias da HTML 4.01 no quesito acessibilidade deve-se ao fato da separação da parte estrutural e da parte de apresentação através do uso de folhas de estilo. Como as características visuais podem ser alteradas de forma independente da estrutura, podem-se criar documentos que apresentem fontes de letra grandes, com cores chamativas, com instruções para dispositivos táteis etc., permitindo a inclusão de mais de 600 milhões ao de pessoas ao redor do mundo que possuem algum tipo de deficiência.

As tabelas e os formulários são os elementos que mais passaram por revisões neste sentido. Outros elementos, como as imagens e vídeos, ganharam descrições longas para facilitar a compreensão daqueles que são incapazes de ver a imagem.

2.4 Internacionalização

Para permitir a representação das diferentes línguas existentes ao redor do planeta, a HTML 4.0 adotou o *Universal Character Set* para a apresentação de textos. A versão anterior adotava um conjunto de caracteres que limitava-se a algumas línguas da Europa. A nova adoção é baseada no Unicode 2.0 e é capaz de representar quase todas as línguas do planeta.

Os atributos LANG e DIR são novos na HTML 4.01 e estão presentes em quase todos os elementos. Estes atributos permitem configurar a língua e a direção do texto que será mostrado no documento.

Outro aprimoramento da linguagem está na exibição de símbolos matemáticos e de letras gregas, bem como de outros caracteres especiais.

2.5 Folhas de Estilo (CSS)

A HTML 4.01 adiciona novos recursos para uso de folhas de estilo, que definem qual será a aparência dos elementos de um documento. Os novos atributos ID, CLASS e STYLE permitem associar informações de estilo para elementos específicos. Os rótulos LINK e STYLE possuem dois novos atributos, TYPE e MEDIA, que especificam, respectivamente, a linguagem usada na definição dos estilos e a mídia em que este estilo figurará.

2.6 Execução de Scripts no Navegador

A HTML 4.01 aprimorou a execução de scripts no navegador (*client-side script*) adicionando vários novos atributos. O rótulo SCRIPT inclui agora atributos para se especificar a linguagem do script, a associação de um script externo e a execução direta de um script. De forma complementar, foram adicionados novos atributos de eventos para permitir a execução de scripts que respondam às mais diversas ações do usuário em um documento, como o clique de um botão do mouse, o pressionamento de uma tecla do teclado, a passagem do cursor do mouse sobre o elemento etc.

O rótulo NOSCRIPT, também novo na HTML 4.01, permite a inclusão de um conteúdo alternativo para navegadores que não suportam execução de scripts.

2.7 Frames

O uso de frames na HTML 4.01 permite a apresentação de múltiplos documentos em uma única janela do navegador. O modelo de frames não sofreu alterações significativas desde o surgimento deste conceito, porém, atualmente desaconselha-se o uso de frames devido à falta de interatividade intrínseca desta técnica.

2.8 Tabelas Avançadas

A forma de apresentação de dados tabulares da HTML 3.2 foi enriquecida na HTML 4.01, de forma a permitir agrupamento de linhas e colunas visando flexibilizar composição da tabela e melhorar questões de acessibilidade. O uso de agrupamento de linhas (THEAD, TFOOT, TBODY) permite aos navegadores renderizar títulos e rodapés estáticos com conteúdo rolável, melhorando a legibilidade de tabelas grandes.

2.9 Considerações Sobre o Capítulo

Este capítulo apresentou algumas novidades da linguagem HTML 4.01 para a versão antecessora, 3.2. Dando continuidade, o capítulo seguinte apresenta a estrutura completa de um documento HTML, com uma descrição sucinta de cada elemento que compõe o documento.

3 A Estrutura de um Documento HTML

Este capítulo visa apresentar os elementos principais de uma página HTML, principalmente no que diz respeito à sua estrutura e organização.

3.1 Elementos e Rótulos (*Tags*)

Elementos são estruturas que descrevem parte de um documento HTML. Por exemplo, o elemento P representa um parágrafo, enquanto o elemento A representa um link para outro documento.

Um elemento é composto por três partes: o rótulo inicial (*tag* de abertura), o conteúdo e o rótulo final (*tag* de fechamento). Um rótulo (*tag*) é um texto especial (de marcação) que é delimitado pelos caracteres < e >. Um rótulo final inclui uma / depois do <. Por exemplo, o elemento P tem um rótulo inicial, <P>, e um rótulo final, </P>. Os rótulos inicial e final delimitam o conteúdo do elemento.

```
<P>Este é um parágrafo</P>
```

Os nomes dos rótulos podem ser escritos tanto em maiúsculas quanto em minúsculas. Isso significa que tanto faz utilizar <p> ou <P>, por exemplo. Entretanto, a W3C, entidade que padroniza diversas coisas relacionadas com a Internet, recomenda o uso de rótulos em letras minúsculas, sendo que, no futuro, esta forma será adotada como padrão.

Um rótulo não pode se misturar com outro rótulo, mas pode conter outro rótulo. Isso significa que não se pode ter algo como o exemplo a seguir:

```
<P>Este é um parágrafo <A>com um link</P></A>
```

Contudo, pode-se perfeitamente ter algo como o exemplo a seguir:

```
<P>Este é um parágrafo <A>com um link</A></P>
```

Este último exemplo define um parágrafo com um link no meio do texto do parágrafo. Alguns elementos permitem omitir o rótulo de fechamento. Por exemplo, o rótulo LI, que define um elemento de uma lista, pode ter seu rótulo de fechamento omitido desde que em seguida tenha-se um novo rótulo LI ou um rótulo de fechamento da lista, conforme o exemplo a seguir:

```
<UL>
  <LI>Primeiro item da lista, sem rótulo de fechamento.
  <LI>Segundo item da lista, com rótulo de fechamento (opcional).</LI>
  <LI>Terceiro elemento da lista, sem rótulo de fechamento.
</UL>
```

Alguns elementos não possuem rótulo de fechamento porque simplesmente não possuem conteúdo, como ocorre com os rótulos BR e HR, que definem, respectivamente, uma quebra de linha e uma linha horizontal. Estes elementos devem ser usados na forma
 e <HR />.

Apesar da opcionalidade de alguns rótulos de fechamento, é altamente recomendável que sempre se utilize tais rótulos, tendo em vista que a linguagem XHTML, próximo padrão a ser adotado pela W3C, não aceita elementos sem rótulos finais.

3.2 Atributos

Os atributos de um elemento definem várias de suas propriedades. Por exemplo, o elemento para definição de links (A) requer usa um atributo chamado HREF para saber para onde o navegador será direcionado quando o usuário clicar neste link, conforme o exemplo a seguir:

```
<A HREF="Contato.html">Página de contato</A>
```

Atributos são informados apenas no rótulo inicial, nunca no rótulo final. Um atributo sempre terá a forma *NomeDoAtributo="ValorDoAtributo"*. O valor do atributo deve ser delimitado por aspas simples ou duplas. Assim como os rótulos, os nomes de atributos da HTML 4.01 não fazem diferença entre maiúsculas e minúsculas, mas recomenda-se o uso de letras minúsculas.

3.3 Caracteres Especiais

Alguns caracteres são especialmente utilizados para marcação, como o caractere <. Os caracteres de marcação, como os rótulos, não são apresentados pelo navegador. Eles simplesmente "dizem" ao navegador como determinado conteúdo deve ser mostrado. Caso se queira mostrar um destes caracteres no navegador, deve-se utilizar uma representação especial. O caractere <, por exemplo, é representado por <. O ponto-e-vírgula faz parte da representação. A representação de caracteres especiais sempre inicia por & e termina com um ; (ponto-e-vírgula). Mais adiante há uma tabela completa dos caracteres especiais e suas representações.

3.4 Comentários

Um *comentário* nada mais é do que um texto que será totalmente ignorado pelo navegador no momento em que este for renderizar seu documento para o usuário. Ele serve para adicionar informações sobre trechos de código, autor, data etc. Para inserir um comentário, basta iniciar a linha com <!-- e finalizar com -->. Além disso, não se deve usar -- no texto do comentário. O exemplo a seguir demonstra o uso de um comentário:

```
<!-- Este é um comentário válido -->
```

3.5 Um Documento HTML Completo

Um documento HTML inicia com uma declaração DOCTYPE, que indica a versão do HTML utilizada no documento. Caso esta declaração seja omitida, o navegador tenta identificar a versão do HTML por sua própria conta. Para evitar este gasto de tempo, é recomendável utilizar a declaração DOCTYPE em todos os documentos HTML.

Em seguida, vem o elemento HTML, que, por sua vez, contém os elementos HEAD e BODY. O HEAD contém informações sobre o documento em si, como o título, palavras-chave etc., enquanto BODY delimita o conteúdo real do documento, conteúdo este composto por vários outros elementos da HTML de forma estruturada, formando-se blocos de código. O exemplo a seguir apresenta um documento HTML com conteúdo bem reduzido:

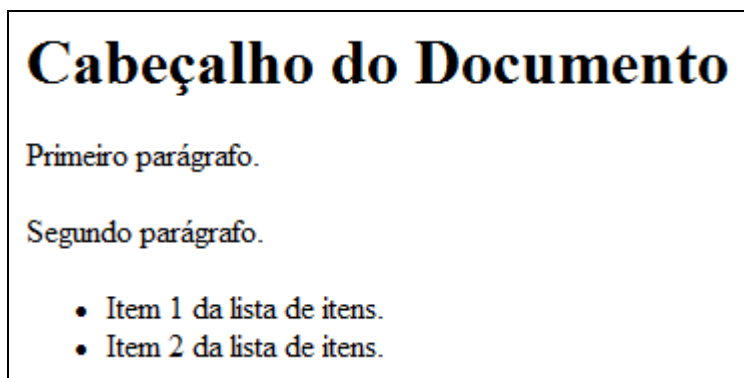
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN"
    "http://www.w3.org/TR/REC-html40/strict.dtd">
<HTML>
  <HEAD>
    <TITLE>Título do Documento</TITLE>
  </HEAD>
```

```

<BODY>
  <H1>Cabeçalho do Documento</H1>
  <P>Primeiro parágrafo.</P>
  <P>Segundo parágrafo.</P>
  <UL>
    <LI>Item 1 da lista de itens.</LI>
    <LI>Item 2 da lista de itens.</LI>
  </UL>
</BODY>
</HTML>

```

O código acima fará com que o navegador apresente um documento com a aparência da figura a seguir:



Obviamente, o documento da figura acima está longe de ser considerado uma página HTML de um site. No decorrer deste material, são apresentadas referências com a sintaxe e a forma de utilização dos principais elementos da linguagem HTML.

3.6 Validando um Documento HTML

É recomendável sempre validar um documento HTML para verificar a presença de possíveis erros, como falta de aspas, falta de caracteres especiais, nomes de atributos e/ou de rótulos inválidos, entre outras coisas. Tais erros nem sempre tornam-se aparentes nas páginas renderizadas pelos navegadores, pois estes são capazes de detectar e corrigir determinados erros durante a exibição de um documento HTML. Contudo, navegadores diferentes realizam esta correção de forma diferente, podendo até mesmo não realizá-la, deixando de exibir um texto importante do documento, por exemplo.

Para validar um documento HTML pode-se utilizar o validador oficial da W3C, disponível no endereço <http://validator.w3.org/>.

3.7 Considerações Sobre o Capítulo

Este capítulo apresentou de forma sucinta e objetiva a estrutura de um documento HTML. Foram apresentados os principais elementos que definem um documento, elementos estes que estarão presentes em praticamente todos os exemplos apresentados no restante deste material.

Os capítulos seguintes iniciam uma abordagem referencial de cada um dos principais elementos da HTML. Cada capítulo aborda um grupo de elementos, categorizados de acordo com o escopo de atuação destes elementos no documento.

4 Elementos de Alto Nível

Os elementos de alto nível são aqueles que delimitam o documento, suas características e seu conteúdo. São eles: HTML, HEAD e BODY, apresentados a seguir.

4.1 HTML

O elemento HTML é o elemento de mais alto nível em um documento HTML. Na verdade, este elemento pode conter os elementos HEAD e BODY, nesta ordem. Esses elementos, no entanto, contêm praticamente todos os outros elementos HTML para se criar um documento.

- Subelementos: HEAD e BODY.
- Elementos que o contém: (não se aplica).

4.1.1 Atributos

O elemento HTML contém apenas os **atributos de internacionalização**, que são LANG e DIR, descritos a seguir.

LANG: especifica a língua que será usada no conteúdo do elemento e de seus subelementos quando estes não possuírem uma definição própria do atributo LANG. Exemplos de valores para este atributo são "en" (Inglês), "en-US" (Inglês Americano), "pt-BR" (Português do Brasil) etc. O uso de LANG não é totalmente suportado pelos navegadores, sendo muito utilizado por ferramentas de busca no momento de indexar um documento HTML agrupando-o por língua em sua base de buscas. Este atributo também é usado por navegadores que usam sintetizadores de voz, de forma que o sotaque do sintetizador esteja de acordo com a língua falada pelo usuário. Ainda, o atributo é usado para definir formatação de valores monetários e outras características visuais dependentes de cultura. Se o atributo LANG for definido no elemento HTML, a configuração afeta todo o documento. Para se atribuir uma língua diferente para uma parte específica do documento, basta redefinir o atributo LANG no elemento que delimita esta parte específica.

DIR: especifica o sentido que o texto será renderizado pelo navegador. Pode assumir os valores "ltr" (esquerda para a direita), que é o valor padrão, ou "rtl" (direita para a esquerda). Na língua portuguesa, por exemplo, os textos são lidos da direita para a esquerda. Já na língua hebraica, lê-se da direita para a esquerda.

4.2 HEAD

O elemento HEAD contém informações sobre o documento, tais como título, palavras-chave, folhas de estilo, descrição etc. É um elemento necessário em todos os documentos, e geralmente vem seguido pelo elemento BODY.

O conteúdo do elemento HEAD costuma não ser renderizado, com exceção do atributo TITLE, que aparece na barra de títulos do navegador.

- Subelementos: um elemento TITLE, um elemento opcional BASE, e zero ou mais elementos SCRIPT, STYLE, META, LINK ou OBJECT.
- Elementos que o contém: HTML.

4.2.1 Atributos

O elemento HEAD contém os atributos de internacionalização (conforme definições do elemento HTML) e o atributo PROFILE, descrito a seguir.

PROFILE: atributo opcional que dá a localização de um perfil de metadados. Um perfil define propriedades que podem ser usadas pelos elementos META e LINK dentro de HEAD. Não há um formato pré-definido para estes perfis.

4.3 BODY

Este elemento contém o corpo do documento. É um elemento necessário para se definir o conteúdo de um documento. Este conteúdo normalmente é composto por outros elementos que definirão características específicas de cada parte do documento.

- Subelementos: um ou mais **elementos de bloco** (vistos mais adiante), além dos elementos SCRIPT, INS e DEL.
- Elementos que o contém: HTML.

4.3.1 Atributos

O elemento BODY contém os atributos de internacionalização LANG e DIR (conforme definições do elemento HTML). Além dos atributos de internacionalização, o elemento BODY possui ainda os chamados **atributos comuns** (ID, CLASS, STYLE e TITLE), que levam esse nome por estarem presentes em quase todos os elementos HTML. Os atributos comuns são descritos a seguir.

ID: é um atributo que identifica o elemento de forma exclusiva em um documento, ou seja, não se pode ter dois elementos com o mesmo ID. Este atributo pode assumir valores que não contenham caracteres de espaço. Os valores devem iniciar com caracteres de A a Z (maiúsculos ou minúsculos), e podem ser seguidos por estes mesmos caracteres ou por números de 0 a 9, por hífen (-), sublinhado (_), dois pontos (:) e pontos (.). Um documento pode, por exemplo, ter parágrafos identificados, como no exemplo a seguir:

```
<P ID="primeiro_paragrafo">Este é o conteúdo do primeiro parágrafo</P>
```

É possível usar o atributo ID para aplicar uma formatação visual usando uma classe de uma folha de estilos (CSS). O ID também pode servir como destino para links, quando referenciados no atributo HREF do elemento A, apresentado mais adiante.

CLASS: este atributo permite definir que um elemento é membro de uma ou mais classes CSS. Classes CSS permitem a criação de diferentes aparências para um mesmo elemento HTML. Pode-se ter, por exemplo, um parágrafo com um estilo chamativo e outro com um estilo mais discreto, ambos usando o elemento P com classes CSS diferentes, como mostra o exemplo a seguir:

```
<P CLASS="discreto">Este é um texto que aparecerá de forma discreta.</P>
```

```
<P CLASS="chamativo">Este é um texto que aparecerá de forma chamativa.</P>
```

Obviamente, o que definirá a aparência de cada elemento serão as definições das classes CSS "discreto" e "chamativo", e a criação de tais definições não está no escopo deste material.

STYLE: este atributo permite a especificação direta de uma definição de estilo sem o uso de classes CSS. Neste caso, diz-se que o estilo é definido **em linha** (*inline*), e deve-se especificar no início do documento,

dentro do elemento HEAD, um elemento META que defina qual o padrão de estilos utilizados pelo documento em questão, conforme o exemplo a seguir:

```
<META HTTP-EQUIV="Content-Style-Type" CONTENT="text/css">
```

Na maioria dos casos, os atributos ID e CLASS são mais aconselhados para a definição de estilos do que o atributo STYLE, tendo em vista que os estilos poderão ser seletivamente aplicados aos elementos de acordo com o dispositivo em que estes forem exibidos, além de separar melhor o conteúdo e a forma como ele será apresentado, facilitando a manutenção do documento.

TITLE: este atributo permite definir um título para o elemento, que é mostrado pelos navegadores como forma de dica (*tooltip*) quando o ponteiro do mouse é repousado sobre o elemento. Este atributo é bastante usado para fornecer uma descrição sucinta do conteúdo dos elementos IMG, A, LINK e OBJECT. É também muito usado para exibir a forma longa dos elementos de abreviaturas ABBR e ACRONYM.

4.3.2 Atributos de Eventos Comuns

O elemento BODY contém ainda atributos de eventos comuns que podem ocorrer pela interação do usuário com o documento. Estes atributos são usados para a execução de scripts no navegador. A listagem a seguir descreve todos os **atributos de eventos comuns**:

- ONCLICK: quando o botão do mouse é clicado sobre o elemento;
- ONDBLCLICK: quando o botão do mouse é clicado duas vezes sobre o elemento;
- ONMOUSEDOWN: quando o botão do mouse é pressionado sobre o elemento;
- ONMOUSEUP: quando o botão do mouse é liberado sobre o elemento;
- ONMOUSEOVER: quando o ponteiro do mouse está sobre o elemento;
- ONMOUSEMOVE: quando o ponteiro do mouse se move sobre o elemento;
- ONMOUSEOUT: quando o ponteiro do mouse é movido para fora do elemento;
- ONKEYPRESS: quando uma tecla é pressionada e liberada sobre um elemento;
- ONKEYDOWN: quando uma tecla é pressionada sobre um elemento;
- ONKEYUP: quando uma tecla é liberada sobre um elemento.

Além dos atributos de eventos comuns, o elemento BODY possui atributos de eventos exclusivos, que são:

- ONLOAD: quando o documento está sendo carregado;
- ONUNLOAD: quando o documento está sendo fechado.

4.4 Considerações Sobre o Capítulo

Este capítulo apresentou os elementos de alto nível de um documento HTML, incluindo os elementos HEAD e BODY, bem como seus atributos. Também foram apresentadas algumas características dos documentos HTML em geral. O capítulo seguinte aborda em detalhes os subelementos do elemento HEAD, apresentando suas funções e atributos.

5 Elementos de Cabeçalho (HEAD)

Os elementos de cabeçalho são BASE, LINK, META, SCRIPT, STYLE e TITLE, descritos individualmente a seguir.

5.1 BASE

O elemento BASE define um endereço base para se utilizar na resolução de endereços (URLs) relativos presentes no documento. Um documento só pode conter um elemento BASE, e este deve figurar dentro de um elemento HEAD, antes de qualquer elemento que faça uso de endereços relativos.

- Subelementos: (não se aplica).
- Elementos que o contém: HEAD.

5.1.1 Atributos

Este elemento contém apenas um atributo, descrito a seguir.

HREF: especifica uma URL absoluta (p.ex., <http://www.marquiao.com>) para ser usada na resolução de URLs relativas presentes no documento. Se um documento contiver um link apontando para o documento "contato.html", o navegador buscará este documento dentro da URL absoluta.

5.2 LINK

O elemento LINK permite associar recursos externos ao documento. O elemento LINK é opcional, podendo aparecer várias vezes dentro do elemento HEAD de um documento, associando o documento com diversos recursos externos.

Os atributos REL e REV definem a natureza desta associação entre os recursos e o documento. REL define um relacionamento do documento com o recurso, enquanto REV define um relacionamento do recurso com o documento, conforme os exemplos a seguir:

```
<LINK REL="Glossary" HREF="glossario_curriculo.html" />
```

```
<LINK REL="Subsection" HREF="formacao.html" />
```

O primeiro elemento LINK diz que o documento contém um glossário, definido pelo documento "glossario_curriculo.html". Já o segundo elemento LINK, diz que o documento é uma subseção de outro documento chamado "formacao.html".

- Subelementos: (não se aplica).
- Elementos que o contém: HEAD.

5.2.1 Atributos

Este elemento contém todos os atributos comuns, mais os atributos REL, REV, HREF, TYPE, MEDIA, HREFLANG e CHARSET, descritos a seguir.

REL e REV: define um tipo de relacionamento do documento com o recurso (REL), sendo que o recurso pode ser parte do documento ou o documento pode ser parte do recurso. Pode assumir os seguintes valores:

- *Made*: especifica um link de contato com o autor do documento.
- *Alternate*: especifica uma versão alternativa do documento. Pode ser usado para indicar uma tradução de um documento (quando há uma definição para o atributo HREFLANG) ou para indicar qual o dispositivo de saída que deve ser usado nesta associação (impressora, vídeo etc.);
- *StyleSheet*: especifica o caminho de um arquivo externo de folhas de estilo CSS para o documento. Pode ser combinado com *Alternate* para definir um arquivo alternativo a ser escolhido pelo usuário.
- *Start*: especifica o primeiro documento de uma coleção de documentos.
- *Next*: especifica o próximo documento de uma seqüência sugerida de leitura. Navegadores presentes em TVs, dispositivos móveis etc., carregam com antecedência documentos identificados com "Next" para evitar intervalos na leitura de dois documentos que formem uma seqüência.
- *Prev*: especifica o documento anterior em uma seqüência sugerida de leitura.
- *Contents*: especifica uma lista de assuntos (tipo um sumário) de um documento.
- *Index*: especifica o índice de um documento (por palavras-chave).
- *Glossary*: especifica um glossário dos termos usados nos documentos.
- *Copyright*: especifica um documento com informações sobre direitos autorais sobre o documento.
- *Chapter*: especifica que o documento é um capítulo de uma coleção de documentos.
- *Section*: especifica que o documento é uma seção de uma coleção de documentos.
- *Subsection*: especifica que o documento é uma subseção de uma coleção de documentos.
- *Appendix*: especifica que o documento é um apêndice de uma coleção de documentos..
- *Help*: especifica que o documento é um documento de ajuda.
- *Bookmark*: especifica um documento chave relacionado. O atributo TITLE provê um título para o marcador.

HREF: especifica a URL para o recurso externo associado ao documento.

TYPE: especifica o tipo de conteúdo do recurso externo associado. Os valores mais comuns são "text/html", "image/jpeg", "model/vrml", "video/quicktime", "application/java", "text/css", e "text/javascript".

MEDIA: especifica o dispositivo de saída para o qual o recurso é designado. Se REL for igual a "StyleSheet", por exemplo, é possível restringir um arquivo de folhas de estilo para impressão, ou para navegadores sonoros. Os possíveis valores do atributo são mostrados na listagem a seguir, e podem ser combinados separando-os por vírgula.

- *screen*: (valor padrão) para exibição em telas de computadores, não paginadas;
- *tty*: para exibição em mostradores com grades de pontos táteis;
- *tv*: para exibição em TVs de baixa resolução e sem opção de rolagem;
- *projection*: para exibição em projetores;
- *handheld*: para exibição em dispositivos móveis, geralmente caracterizados por acessarem os documentos usando conexões lentas;
- *print*: para exibição em material impresso;
- *braille*: para exibição em dispositivos táteis em braille;
- *aural*: para exibição em sintetizadores de voz;
- *all*: para exibição em todos os dispositivos.

HREFLANG: mesma função do atributo LANG presente no elemento HTML.

CHARSET: definição da codificação de caracteres utilizada pelo recurso, podendo assumir os valores ISO-8859-1, SHIFT_JIS e UTF-8. A seguir, são listados diversos exemplos de uso do elemento LINK:

```
<LINK REL="Prev" HREF="inicio.html" TITLE="Página Inicial" />
<LINK REL="Next" HREF="capitulo2.html" TITLE="Estruturas de Navegação" />
<LINK REL="Start" HREF="capa.html" TITLE="HTML - Manual de Referência" />
<LINK REL="Copyright" HREF="copyright.html" TITLE="Direitos Autorais" />
<LINK REL="Made" HREF="mailto:maroquio@gmail.com" />
<LINK REL="StyleSheet" HREF="blocos.css" TITLE="EstiloAtual" TYPE="text/css" />
<LINK REL="StyleSheet" HREF="tabelas.css" TITLE="EstiloAtual" TYPE="text/css" />
<LINK REL="StyleSheet" HREF="formularios.css" TITLE="EstiloAtual"
  TYPE="text/css" />
<LINK REL="Alternate" HREF="index.fr.html" HREFLANG="fr" LANG="fr"
  TITLE="Version française" />
<LINK REL="Alternate" HREF="index.ja.html" HREFLANG="ja" CHARSET="SHIFT_JIS"
  TITLE="Japanese version" />
<LINK REL="Alternate" HREF="/arquivo/manualHTML.pdf" TYPE="application/pdf"
  MEDIA="print" TITLE="Versão em PDF">
```

5.3 META

O elemento META permite a inclusão de metadados no documento, como palavras-chave que descrevem o conteúdo do documento, um resumo do documento, dados sobre o autor etc. Pode-se incluir quantos elementos META forem necessários, sempre dentro do elemento HEAD do documento.

- Subelementos: (não se aplica).
- Elementos que o contém: HEAD

5.3.1 Atributos

O elemento META possui todos os atributos de internacionalização, bem como os atributos específicos NAME, CONTENT, SCHEME e HTTP-EQUIV, descritos a seguir.

NAME: especifica o nome de uma propriedade do documento que se queira fornecer mais informações ou esclarecimentos. É um atributo opcional.

CONTENT: especifica a descrição de uma propriedade definida pelo atributo NAME. É um atributo opcional. Este atributo não pode conter valores com elementos HTML.

SCHEME: atributo opcional que permite definir o formato do valor de uma propriedade, como uma data. Possíveis valores são "Month-Day-Year" e "Day-Month-Year".

HTTP-EQUIV: usado no lugar do atributo NAME para indicar que a propriedade é um cabeçalho HTTP. Alguns servidores enviam os cabeçalhos HTTP especificados nos elementos META antes mesmo de enviar o documento, de forma que o navegador dê um tratamento distinto ao documento.

Alguns exemplos de uso:

- Define uma data para o documento expirar.

```
<META HTTP-EQUIV="Expires" CONTENT="Sun, 22 Mar 1998 16:18:35 GMT">
```

- Define que os scripts *inline* executados pelo navegador são do tipo JavaScript.

```
<META HTTP-EQUIV="Content-Script-Type" CONTENT="text/javascript">
```

- Define que os estilos *inline* usados pelo navegador são do tipo CSS.

```
<META HTTP-EQUIV="Content-Style-Type" CONTENT="text/css">
```

- Define a codificação de caracteres do documento para SHIFT_JIS (uma codificação Japonesa).

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=SHIFT_JIS">
```

- Informa ao navegador que ele deve carregar a URL <http://www.marquiao.com/> 10 segundos após o fim do carregamento do documento atual. Não é suportado por todos os navegadores.

```
<META HTTP-EQUIV="Refresh" CONTENT="10; URL=http://www.marquiao.com/">
```

5.4 SCRIPT

O elemento SCRIPT permite a inclusão de um script que será executado pelo navegador do usuário que está acessando o documento. Através de scripts, pode-se dar mais interatividade ao documento, executando ações em resposta às interações do usuário com o documento. Pode-se, por exemplo, incluir um script que verifique se determinado campo de um formulário foi preenchido antes de enviá-lo.

Uma observação importante é que nem todos os navegadores suportam ou permitem a execução de scripts. Neste caso, é possível definir partes do documento específicas para estes navegadores, usando o elemento NOSCRIPT, apresentado mais adiante.

- Subelementos: apenas código de script em alguma linguagem (p.ex., JavaScript).
- Elementos que o contém: HEAD (zero ou vários), **elementos de bloco** (zero ou vários; vistos mais adiante) e **atributos de eventos comuns** presentes na maioria dos elementos HTML (versão *inline*; veja os atributos na seção 4.3.2).

5.4.1 Atributos

TYPE: atributo necessário que define o tipo de mídia da linguagem de script (p.ex., *text/javascript*). Contudo, alguns navegadores suportam apenas o atributo LANGUAGE, que é obsoleto nos navegadores mais atuais.

LANGUAGE: define o nome da linguagem de script usada. Valores possíveis são "JavaScript", "JavaScript1.1", "JavaScript1.2", "VBScript". Tanto faz usar letras maiúsculas ou minúsculas nos caracteres do nome da linguagem. Se for informada uma linguagem não suportada pelo navegador, ele simplesmente vai ignorar o elemento SCRIPT. No caso da ausência do atributo LANGUAGE, os navegadores estabelecem como padrão a linguagem JavaScript 1.0. Como não é possível definir a linguagem do script no atributo TYPE, recomenda-se utilizar o atributo LANGUAGE para definir a versão da linguagem utilizada.

SRC: permite especificar um arquivo externo com código de script (p.ex., *scripts.js*) que contenha algo a ser utilizado pelo script que está sendo definido como conteúdo do elemento SCRIPT.

CHARSET: permite definir a codificação de caracteres usada pelo script do arquivo externo associado pelo atributo SRC.

DEFER: faz com que o navegador aguarde a renderização completa da página antes de ativar o uso dos scripts, evitando, por exemplo, respostas a eventos do documento antes que ele esteja completamente renderizado.

O exemplo a seguir mostra um script em JavaScript que inclui um texto no documento.

```
<SCRIPT TYPE="text/javascript">
    document.write("Texto a ser inserido no documento.");
</SCRIPT>
```

5.5 STYLE

O elemento STYLE incorpora uma folha de estilos a um documento. Pode-se incluir quantos elementos STYLE forem necessários dentro do elemento HEAD. Uma folha de estilos define a formatação dos elementos HTML de um documento, permitindo a separação da parte de conteúdo e da parte de apresentação do documento. Foge ao escopo deste documento uma explicação mais detalhada sobre folhas de estilo.

- Subelementos: apenas códigos de folhas de estilo.
- Elementos que o contém: HEAD (zero ou vários), atributos STYLE (versão *inline*).

5.5.1 Atributos

Além dos atributos de internacionalização, o elemento STYLE possui os atributos TYPE, MEDIA e TITLE, apresentados a seguir.

TYPE: usado para especificar o tipo de mídia da linguagem de estilo. Para folhas de estilo em cascata (CSS), usa-se "text/css".

TITLE: atributo opcional que permite incluir um título ao estilo. Alguns navegadores permitem selecionar qual folha de estilo será aplicada ao documento, através do título. Portanto, folhas de estilo com o mesmo título são consideradas iguais.

MEDIA: usado para especificar a mídia à qual a folha de estilos será aplicada. Os possíveis valores são os mesmos apresentados na seção 5.2.1.

Um exemplo de folha de estilos é mostrado a seguir.

```
<STYLE TYPE="text/css" MEDIA="screen">
    BODY { background: url(fundo.gif) red; color: black }
    P EM { background: yellow; color: black }
    .texto { margin-left: 5px; margin-right: 5px }
</STYLE>
```

5.6 TITLE

O elemento TITLE permite definir o título do documento. Cada documento só pode ter um título dentro do elemento HEAD. O elemento TITLE é usado para exibir o título do documento na barra de títulos do navegador e em links de ferramentas de busca. Um bom título deve ser bem específico em relação ao documento que representa. Uma recomendação é que o título tenha no máximo 60 caracteres.

- Subelementos: apenas texto plano (não pode conter outros elementos HTML).
- Elementos que o contém: HEAD.

5.7 Considerações Sobre o Capítulo

Este capítulo apresentou em detalhes os subelementos HTML do elemento HEAD, bem como suas funções e atributos. O capítulo seguinte apresenta os elementos de bloco da HTML, elementos estes que ocupam uma ou mais linhas, não permitindo, por padrão, que outros elementos fiquem à sua esquerda ou direita.

6 Elementos em Nível de Bloco

Elementos em nível de bloco são caracterizados por não permitirem, por padrão, outros elementos HTML à sua esquerda ou direita. Um elemento de bloco não pode estar contido no meio do texto de um parágrafo, por exemplo. Os elementos de bloco são: ADDRESS, BLOCKQUOTE, CENTER, DEL, DIV, H1, H2, H3, H4, H5, H6, HR, INS, ISINDEX, NOSCRIPT, P e PRE, apresentados individualmente a seguir.

6.1 ADDRESS

O elemento ADDRESS permite inserir informações de contato em um documento. Estas informações podem incluir o nome do autor do documento, o endereço de email, o telefone, o endereço profissional etc. Não se deve usar o elemento ADDRESS para mostrar todos os endereços de email ou CEPs, por exemplo. Seu uso é recomendado para prover informações sobre o autor do documento, ou o responsável pelo documento.

- Subelementos: qualquer elemento de conteúdo da HTML.
- Elementos que o contém: BLOCKQUOTE, BODY, BUTTON, DD, DEL, DIV, FIELDSET, FORM, INS, LI, MAP, NOFRAMES, NOSCRIPT, OBJECT, TD, TH.

6.1.1 Atributos

O elemento ADDRESS possui apenas os atributos chamados de **atributos comuns** (ID, CLASS, STYLE e TITLE), apresentados na seção 4.3.1.

A seguir é apresentado um exemplo do uso de ADDRESS.

```
<ADDRESS>
  Qualquer questão sobre este curso, entre em contato pelo email
  <A HREF="mailto:maroquio@gmail.com">maroquio@gmail.com</A>,
  ou através pelo telefone 9999-9999.
</ADDRESS>
```

6.2 BLOCKQUOTE

O elemento BLOCKQUOTE é usado para se definir citações longas em um documento. Para citações curtas deve-se utilizar o elemento Q. A principal diferença entre estes elementos é que BLOCKQUOTE pode conter outros elementos de bloco, como P, por exemplo. Entretanto, BLOCKQUOTE não pode estar contido em um elemento P ou em qualquer outro elemento de linha (A, SPAN etc.).

- Subelementos: elementos de bloco, SCRIPT.
- Elementos que o contém: BLOCKQUOTE, BODY, BUTTON, DD, DEL, DIV, FIELDSET, FORM, INS, LI, MAP, NOFRAMES, NOSCRIPT, OBJECT, TD, TH.

6.2.1 Atributos

Além dos atributos comuns, BLOCKQUOTE possui o atributo CITE, descrito a seguir.

CITE: usado para definir uma URL da fonte das informações mostradas no conteúdo do BLOCKQUOTE.

Um exemplo de uso de BLOCKQUOTE é apresentado a seguir:

```
<BLOCKQUOTE CITE="http://www.marquiao.com/QuemSou.aspx">
  <P>Durante o mestrado, estudei e me aprofundei em Robótica, Inteligência
  Artificial, Computação Gráfica, Física Newtoniana etc.</P>
</BLOCKQUOTE>
```

6.3 DEL

O elemento DEL apresenta qualquer conteúdo que deva figurar como excluído do documento. Este elemento é indicado para situações em que se deseja mostrar algo excluído de versões anteriores do documento. Usando CSS, é possível ocultar estes elementos, ou mostrá-los como tachados (riscados).

- Subelementos: elementos de bloco e elementos de linha.
- Elementos que o contém: elementos de bloco e elementos de linha.

6.3.1 Atributos

Além dos atributos comuns, o elemento DEL possui os atributos CITE e DATETIME, descritos a seguir.

CITE: usado para definir a URL de um documento que contenha uma justificativa para a exclusão do conteúdo.

DATETIME: atributo opcional que define a data e a hora da exclusão, no formato YYYY-MM-DDThh:mm:ssTZD.

Um exemplo de uso do elemento DEL é apresentado a seguir.

```
<DEL CITE="http://www.marquiao.com/cai/v1.3/mudacas.txt" DATETIME="1999-07-
31T19:30:00-03:00" TITLE="Função obsoleta"><P>Função que permite o agendamento
de conexões, com a inicialização automática de downloads.</P></DEL>
```

6.4 DIV

O elemento DIV define um bloco genérico de conteúdo. Por padrão, ele não muda o visual dos elementos que contiver. Portanto, com o uso de folhas de estilo, é possível aplicar diversas características visuais a estes blocos. Desde o posicionamento até bordas e imagens de fundo podem ser configuradas através de folhas de estilo. O elemento em linha equivalente ao DIV é o SPAN.

- Subelementos: elementos de bloco e elementos de linha.
- Elementos que o contém: BLOCKQUOTE, BODY, BUTTON, DD, DEL, DIV, FIELDSET, FORM, INS, LI, MAP, NOFRAMES, NOSCRIPT, OBJECT, TD, TH.

O elemento DIV só possui os atributos comuns.

6.5 H1 .. H6

Os elementos H1, H2, H3, H4, H5 e H6 são usados para intitular partes do documento. O elemento H1 é para títulos de nível 1 e o elemento H6 para títulos de nível 6, usando-se a mesma lógica para os demais elementos. Por padrão, os navegadores renderizam elementos H1 usando fontes grandes, em negrito. Elementos H2 usam fontes um pouco menores, e o tamanho vai diminuindo progressivamente para os elementos H3, H4, H5 e H6. Usando-se folhas de estilo pode-se facilmente redefinir a aparência destes elementos.

- Subelementos: elementos de linha.
- Elementos que o contém: BLOCKQUOTE, BODY, BUTTON, DD, DEL, DIV, FIELDSET, FORM, INS, LI, MAP, NOFRAMES, NOSCRIPT, OBJECT, TD, TH.

Estes elementos possuem apenas os atributos comuns. Um exemplo de uso é apresentado a seguir.

```
<H1>Capítulo 1 – Introdução</H1>
<P>Texto do Capítulo 1...</P>
<H2>1.1 – Histórico</H2>
<P>Descrição do histórico...</P>
```

6.6 HR

O elemento HR é usado quando se deseja incluir uma linha horizontal que divide duas partes do documento, ficando uma parte acima da linha e outra abaixo da linha. Pode ser usado estruturalmente como um divisor de seções.

- Subelementos: (não se aplica).
- Elementos que o contém: BLOCKQUOTE, BODY, BUTTON, DD, DEL, DIV, FIELDSET, FORM, INS, LI, MAP, NOFRAMES, NOSCRIPT, OBJECT, TD, TH.

O elemento HR possui apenas os atributos comuns (seção 4.3.1) e os atributos de eventos comuns (seção 4.3.2).

6.7 INS

O elemento INS apresenta qualquer conteúdo que deva figurar como incluído no documento. Este elemento é indicado para situações em que se deseja mostrar algo que não existia em versões anteriores do documento. Usando folhas de estilo, é possível mostrar estes elementos com algum destaque.

- Subelementos: elementos de bloco e elementos de linha.
- Elementos que o contém: elementos de bloco e elementos de linha.

6.7.1 Atributos

Além dos atributos comuns, o elemento INS possui os atributos CITE e DATETIME, descritos a seguir.

CITE: usado para definir a URL de um documento que contenha uma justificativa para a inclusão do conteúdo.

DATETIME: atributo opcional que define a data e a hora da inclusão, no formato *YYYY-MM-DDThh:mm:ssTZD*.

Um exemplo de uso do elemento INS é apresentado a seguir.

```
<INS CITE="http://www.marquiao.com/cai/v1.3/mudacas.txt" DATETIME="1999-07-31T19:30:00-03:00" TITLE="Nova Função"><P>Função que permite a desconexão automática agendada para um horário.</P></INS>
```


6.8 NOSCRIPT

O elemento NOSCRIPT é usado para fornecer um conteúdo alternativo quando documentos que usam scripts precisam ser exibidos em navegadores que não oferecem suporte a scripts ou a determinadas linguagens de script. Se o navegador oferecer suporte a scripts, o conteúdo de NOSCRIPT é ignorado.

- Subelementos: elementos de bloco.
- Elementos que o contém: BLOCKQUOTE, BODY, BUTTON, DD, DEL, DIV, FIELDSET, FORM, INS, LI, MAP, NOFRAMES, NOSCRIPT, OBJECT, TD, TH.

O elemento NOSCRIPT possui apenas os atributos comuns.

6.9 P

O elemento P é usado para se criar parágrafos em um documento. A linguagem HTML não especifica uma apresentação visual distinta para o elemento P. Por padrão, seu conteúdo aparece como um bloco de texto normal. Deve-se observar que não é recomendado incluir elementos de bloco (DIV, P) como conteúdo de um elemento P.

- Subelementos: elementos de linha.
- Elementos que o contém: BLOCKQUOTE, BODY, BUTTON, DD, DEL, DIV, FIELDSET, FORM, INS, LI, MAP, NOFRAMES, NOSCRIPT, OBJECT, TD, TH.

O elemento P possui apenas os atributos comuns.

6.10 PRE

O elemento PRE é usado para exibir texto pré-formatado, como o código fonte de um programa ou um poema. Nestes casos, é importante que os espaços em branco e as quebras de linha ocorram da forma como foram digitados no documento. No conteúdo do elemento PRE, os espaços em branco não são comprimidos em um único espaço, e as quebras de linha automática são desativadas.

- Subelementos: elementos de linha, **exceto** IMG, OBJECT, APPLET, BIG, SMALL, SUB, SUP, FONT, BASEFONT.
- Elementos que o contém: BLOCKQUOTE, BODY, BUTTON, DD, DEL, DIV, FIELDSET, FORM, INS, LI, MAP, NOFRAMES, NOSCRIPT, OBJECT, TD, TH.

Este elemento possui apenas os atributos comuns. A seguir, é apresentado um exemplo de uso do elemento PRE para mostrar um trecho de código fonte em JavaScript.

```
<PRE>
    function MostrarMensagem {
        alert("Esta é uma mensagem de alerta!");
    }
</PRE>
```

6.11 Considerações Sobre o Capítulo

Este capítulo apresentou os elementos de bloco da HTML, elementos estes que, por padrão, não permitem outros elementos em qualquer de seus lados, ocupando uma faixa horizontal inteira do documento. São normalmente usados para conter outros elementos. O capítulo seguinte apresenta elementos comuns em nível de linha da HTML, elementos estes que podem figurar em uma linha de conteúdo sem quebrá-la.

7 Elementos Comuns em Nível de Linha

Os elementos comuns em nível de linha são geralmente usados dentro de parágrafos, e são divididos em duas categorias: elementos para composição de frases e elementos para formatação de fonte de letra. Uma característica comum destes elementos é que eles possuem apenas os atributos comuns, podem conter apenas elementos de linha, e podem estar contidos em elementos de linha e elementos de bloco. As seções seguintes apresentam os elementos comuns em nível de linha.

7.1 Elementos para Composição de Frases

Os elementos para composição de frases são aqueles que alteram a aparência de toda uma frase ou de parte dela, dando um sentido diferente o conteúdo. Os elementos incluídos nesta categoria são: ABBR, ACRONYM, CITE, CODE, DEL, DFN, EM, INS KBD, SAMP, STRONG e VAR. As seções a seguir abordam cada elemento separadamente.

7.1.1 ABBR

O elemento ABBR é usado para se criar abreviações. Neste caso, o atributo TITLE pode ser usado para conter a forma longa da abreviação, que é mostrada pelos navegadores através de uma dica (*tooltip*). Se a palavra abreviada for pronunciável, o elemento ACRONYM é mais indicado.

7.1.2 ACRONYM

O elemento ACRONYM é usado para se criar acrônimos, que são versões reduzidas e pronunciáveis de uma ou mais palavras. Neste caso, o atributo TITLE pode ser usado para conter a forma longa do acrônimo, que é mostrada pelos navegadores através de uma dica (*tooltip*).

7.1.3 CITE

O elemento CITE é usado em citações, tais como para títulos de livros ou revistas, nomes de museus etc. Os navegadores normalmente renderizam o conteúdo de CITE em itálico, mas esse padrão pode ser alterado via folhas de estilo. Sendo CITE um elemento estrutural, deve-se dar preferência a seu uso quando o conteúdo a ser exibido for uma citação, em vez de se usar o elemento I, por exemplo.

7.1.4 CODE

O elemento CODE é usado para se exibir código-fonte em alguma linguagem de computador. Os navegadores renderizam o conteúdo de CODE usando uma fonte mono-espaçada, sendo que este efeito pode ser alterado usando-se folhas de estilo. Uma vez que CODE é um elemento estrutural, ele também carrega um significado, sendo seu uso indicado quando se quer exibir código-fonte em alguma linguagem computacional, como HTML, Pascal, C# etc. O elemento CODE é muito usado junto com o elemento PRE, de forma que a indentação do código a ser exibido fique aparente.

7.1.5 DEL

Ver seção 6.3.

7.1.6 DFN

O elemento DFN é usado para se exibir a definição de instância de um termo. Os navegadores normalmente renderizam o conteúdo de DFN em itálico, mas esse padrão pode ser alterado via folhas de estilo. Sendo DFN um elemento estrutural, deve-se dar preferência a seu uso quando o conteúdo for uma definição, em vez de se usar o elemento I, por exemplo.

7.1.7 EM

O elemento EM é usado para se exibir um conteúdo enfatizado. Os navegadores normalmente renderizam o conteúdo de EM em itálico, mas esse padrão pode ser alterado via folhas de estilo. Sendo EM um elemento estrutural, deve-se dar preferência a seu uso quando o conteúdo deva aparecer de forma enfatizada, em vez de se usar o elemento I, por exemplo.

7.1.8 INS

Ver seção 6.7.

7.1.9 KBD

O elemento KBD denota algo a ser digitado pelo usuário. Os navegadores normalmente renderizam o conteúdo de KBD usando fonte mono-espaçada, mas esse padrão pode ser alterado via folhas de estilo. Sendo KBD um elemento estrutural, deve-se dar preferência a seu uso quando o conteúdo a ser exibido deva figurar como algo a ser digitado pelo usuário.

7.1.10 SAMP

O elemento SAMP denota um exemplo de saída (resposta) de um programa ou script. Os navegadores normalmente renderizam o conteúdo de SAMP usando fonte mono-espaçada, mas esse padrão pode ser alterado via folhas de estilo. Sendo SAMP um elemento estrutural, deve-se dar preferência a seu uso quando o conteúdo a ser exibido for algo que deva figurar como a saída de um programa.

7.1.11 STRONG

O elemento STRONG é usado quando se deseja dar uma grande ênfase a um conteúdo. Os navegadores normalmente renderizam o conteúdo de STRONG em negrito, mas esse padrão pode ser alterado via folhas de estilo. Sendo STRONG um elemento estrutural, deve-se dar preferência a seu uso quando o conteúdo a ser exibido seja algo que deva realmente ser enfatizado, em vez de se usar B, por exemplo, apenas para negritar.

7.1.12 VAR

O elemento VAR é usado quando se deseja exibir uma variável ou um argumento. Sendo VAR um elemento estrutural, deve-se dar preferência a seu uso quando o conteúdo a ser exibido for uma variável, em vez de se usar I, por exemplo, para apenas deixar em itálico.

7.2 Elementos para Formatação de Fonte de Letra

Os elementos usados exclusivamente para formatar fontes de letra são os elementos BOLD, BIG, I, SMALL e TT, descritos individualmente nas seções a seguir.

7.2.1 B

O elemento B é usado para apresentar conteúdo em negrito. Em determinadas situações, deve-se considerar a possibilidade de se usar STRONG em vez de B, tendo em vista que STRONG tem um significado estrutural.

7.2.2 BIG

O elemento BIG é usado para apresentar conteúdo com uma fonte de letra grande. Em determinadas situações, deve-se considerar a possibilidade de se usar STRONG ou H1 (Hn) em vez de B, tendo em vista que STRONG e H1 têm um significado estrutural.

7.2.3 I

O elemento I é usado para apresentar conteúdo em itálico. Em determinadas situações, deve-se considerar a possibilidade de se usar DFN, EM, VAR ou CITE, tendo em vista que esses elementos têm um significado estrutural.

7.2.4 SMALL

O elemento BIG é usado para apresentar conteúdo com uma fonte de letra pequena. Geralmente é usado para exibir informações sem qualquer ênfase, como as notas de direitos autorais do rodapé de uma página.

7.2.5 TT

O elemento TT é usado para apresentar conteúdo com fonte de letra mono-espaçada. Em determinadas situações, deve-se considerar a possibilidade de se usar CODE, SAMP ou KBD, tendo em vista que esses elementos têm um significado estrutural.

7.3 Considerações Sobre o Capítulo

Este capítulo apresentou diversos elementos comuns em nível de linha da HTML, ou seja, cujos conteúdos podem aparecer no meio de uma frase sem quebrar a linha. O capítulo seguinte apresenta elementos especiais em nível de linha da HTML, elementos estes que se diferenciam dos elementos comuns por possuírem atributos que permitem configurar a apresentação/comportamento do conteúdo exibido.

8 Elementos Especiais em Nível de Linha

Este capítulo apresenta elementos especiais de linha da HTML, elementos estes que podem figurar normalmente em uma linha de conteúdo, sem quebrá-la. Estes elementos são considerados especiais por possuírem diversos atributos que definem o comportamento de seu conteúdo, e não apenas a forma de apresentação. Ao contrário dos elementos de bloco, elementos de linha geralmente não contêm outros elementos HTML. Entretanto, alguns elementos apresentados neste capítulo podem conter outros elementos. Os elementos de linha especiais são: A, BDO, BR, IMG, MAP, AREA, OBJECT, PARAM, Q, SCRIPT, SPAN, SUB e SUP, descritos a seguir.

8.1 A

O elemento A é usado para criar links em um documento HTML. Links são atalhos de navegação para partes do documento ou para outros documentos. Quando clicados, estes links levam até o destino informado no atributo HREF. O elemento A também pode ser usado para rotular uma região de um documento longo, permitindo que o destino de um link seja esta região. Para rotular uma região usa-se o atributo NAME do elemento para dar um nome à região.

- Subelementos: elementos de linha, **exceto** A.
- Elementos que o contém: elementos de bloco, elementos de linha, **exceto** A.

8.1.1 Atributos

Os atributos do elemento A incluem os atributos comuns mais os atributos HREF, NAME, REL, VER, TYPE, TARGET, HREFLANG, CHARSET, ACCESSKEY, TABINDEX, SHAPE e COORDS, descritos a seguir.

HREF: usado para determinar a URL de destino do link.

NAME: permite nomear uma região do documento alcançável por um link. Um elemento A com o atributo NAME igual a “alvo” pode ser acessível no mesmo documento configurando-se o atributo HREF de outro elemento A como “#alvo”, ou, se for em outro documento, “outrodocumento.html#alvo”.

REL: define a relação do documento corrente com o documento de destino do link. Os tipos de relação podem ser consultados na seção 5.2.1.

REV: define a relação do documento de destino do link com o documento corrente. Os tipos de relação podem ser consultados na seção 5.2.1.

TYPE: usado para determinar o tipo de conteúdo do documento, podendo assumir valores como “text/HTML”, “image/jpeg”, “model/vrml”, “video/quicktime”, “application/Java”, “text/css” e “text/javascript”, evitando que navegadores tentem mostrar documentos que não são capazes.

TARGET: usado normalmente com frames. Contudo, como o uso de frames está obsoleto, pode-se utilizar este atributo para que o documento de destino do link seja aberto em uma nova janela. Para isso, usa-se o valor “_blank” no atributo.

HREFLANG: idioma do conteúdo do link.

CHARSET: codificação de texto usada no texto do link.

ACCESSKEY: permite especificar um único caractere que servirá de tecla de atalho para acessar o link.

TABINDEX: define a ordem de tabulação do elemento. A ordem de tabulação determina quando o elemento receberá o foco enquanto o usuário pressiona a tecla TAB do teclado. Quanto menor for o valor de TABINDEX, mais cedo ele será focado.

SHAPE e COORDS: mesma funcionalidade dos atributos do elemento AREA. Usado quando o elemento A é um subelemento de OBJECT.

8.1.2 Atributos de Eventos

Os atributos de eventos do elemento A são ONFOCUS e ONBLUR, descritos a seguir.

ONFOCUS: ocorre quando o elemento recebe o foco.

ONBLUR: ocorre quando o elemento perde o foco.

8.2 BDO

O elemento BDO faz com que o sentido de escrita do texto de seu conteúdo possa ser configurado. Este elemento sobrescreve o algoritmo padrão de renderização de texto do navegador e permite que se defina, através do atributo DIR, se o texto deverá figurar da esquerda para a direita (padrão) ou da direita para a esquerda.

- Subelementos: elementos de linha.
- Elementos que o contém: elementos de bloco, elementos de linha.

O elemento BDO possui apenas os atributos comuns e os atributos de internacionalização.

8.3 BR

O elemento BR força uma quebra de linha na linha em que for colocado. Pode ser usado, por exemplo, para quebrar linhas de texto de uma informação de endereço em um elemento ADDRESS. Algumas pessoas utilizam o BR para quebrar linhas dentro de parágrafos (elemento P) e células de tabelas (elemento TD). Nestes casos, se o usuário alterar o tamanho da fonte do navegador, estas quebras podem ter um efeito indesejado, com linhas sendo quebradas duas vezes, devido às quebras automáticas realizadas pelo navegador.

- Subelementos: (não se aplica).
- Elementos que o contém: elementos de bloco, elementos de linha.

O elemento BR possui apenas os atributos comuns.

8.4 IMG

O elemento IMG é usado para se adicionar uma imagem dentro de uma linha de conteúdo. A imagem pode estar nos formatos JPG ou GIF, sendo que os navegadores mais atuais suportam também o formato PNG, um formato que, além de possuir boa compressão, possui a capacidade de apresentar transparência.

- Subelementos: (não se aplica).
- Elementos que o contém: elementos de bloco, elementos de linha, exceto PRE.

8.4.1 Atributos

Os atributos do elemento IMG incluem os atributos comuns mais os atributos SRC, ALT, LONGDESC, WIDTH, HEIGHT, USEMAP e ISMAP, descritos a seguir.

SRC: usado para se definir a URL do local onde a imagem se encontra.

ALT: usado para se prover um texto alternativo que será mostrado caso a imagem não possa ser mostrada, seja qual for o motivo.

LONGDESC: usado para se prover uma descrição longa a respeito da imagem mostrada, de forma que pessoas que não possam ver a imagem possam ao menos mentalizá-la a partir de sua descrição textual.

WIDTH: permite definir uma largura fixa para a imagem, de forma que ela ocupe apenas a área que tenha sido reservada para ela.

HEIGHT: assim como o atributo WIDTH, permite definir uma altura fixa para a imagem, de forma que ela ocupe apenas a área que tenha sido reservada para ela.

USEMAP: permite definir um mapa para a imagem, dividindo-a em regiões clicáveis que podem ter ações distintas. Um mapa é definido pelo elemento MAP. Se o mapa estiver na mesma página, usa-se apenas o valor “#nomedomapa”. Se estiver em outro documento, usa-se o endereço completo da URL seguido do valor já citado.

ISMAP: a existência deste atributo no elemento faz com que as coordenadas da região clicada seja enviada ao servidor, para que ele interprete a região com algum processamento *server-side*. Caso o atributo não seja informado, a região clicada é interpretada diretamente pelo navegador.

8.5 MAP

O elemento MAP define um mapa que pode ser usado para dividir uma imagem ou um objeto em regiões clicáveis que se comportam como os links (elemento A), sendo mostrados como atalhos para regiões do próprio documento ou como atalhos para outros documentos. O elemento MAP, apesar de poder conter outros elementos HTML, foi originalmente concebido para conter elementos AREA, que, por sua vez, definem as regiões clicáveis.

- Subelementos: elementos de bloco, um ou mais elementos AREA.
- Elementos que o contém: elementos de bloco e elementos de linha.

8.5.1 Atributos

Além dos atributos comuns, o elemento MAP possui o atributo NAME, descrito a seguir.

NAME: mesma função do atributo ID, que é a de prover um identificador único para o elemento. Seu preenchimento é necessário, pois alguns navegadores não reconhecem o atributo ID.

8.6 AREA

O elemento AREA é usado dentro do elemento MAP para definir uma região do mapa. Cada região do mapa é uma parte da imagem que pode ter uma ação específica quando clicada.

- Subelementos: (não se aplica).

- Elementos que o contém: MAP.

8.6.1 Atributos

Os atributos do elemento AREA incluem os atributos comuns mais os atributos SHAPE, COORDS, HREF, NOHREF, ALT e TABINDEX, descritos a seguir.

SHAPE e COORDS: especifica uma das divisões da imagem. Pode ter as seguintes configurações:

- *default*: especifica a imagem inteira;
- *rect*: especifica uma região retangular, definindo-se COORDS="x1, y1, x2, y2", como coordenadas do canto superior esquerdo e do canto inferior direito, nesta ordem.
- *circle*: especifica uma região circular, definindo-se COORDS="centro-x, centro-y, raio";
- *poly*: especifica uma região poligonal irregular, definindo-se COORDS="x1, y1, x2, y2, ..., xN, yN".

HREF: especifica o link para o recurso de destino, podendo ser um documento HTML ou uma imagem JPEG, por exemplo.

NOHREF: indica que a região não tem link.

ALT: provê um texto alternativo à imagem, caso ela não seja exibida.

TABINDEX: define a ordem de tabulação da região corrente.

8.6.2 Atributos de Eventos

Os atributos de eventos do elemento AREA são ONFOCUS e ONBLUR, descritos a seguir.

ONFOCUS: ocorre quando o elemento recebe o foco.

ONBLUR: ocorre quando o elemento perde o foco.

8.7 OBJECT

O elemento OBJECT é usado para se incluir no documento HTML objetos tais como imagens, vídeos, *applets* Java, cenários 3D em VRML etc. Cada um destes tipos de objetos possui um plug-in próprio para exibi-lo. Muitas vezes, a exibição destes objetos requer a atribuição de alguns parâmetros. Sons e vídeos, por exemplo, têm o parâmetro *volume* para ser informado. Estes parâmetros podem ser informados usando-se o elemento PARAM, visto mais diante.

- Subelementos: PARAM, seguido de elementos de bloco e/ou elementos de linha.
- Elementos que o contém: HEAD, elementos de bloco, elementos de linha, exceto PRE.

8.7.1 Atributos

Os atributos do elemento OBJECT incluem os atributos comuns mais os atributos DATA, CLASSID, ARCHIVE, CODEBASE, WIDTH, HEIGHT, NAME, USEMAP, TYPE, CODETYPE, STANDBY, TABINDEX e DECLARE, descritos a seguir.

DATA: define a URL do objeto representado pelo elemento.

WIDTH e HEIGHT: definem, respectivamente, a largura e a altura do objeto. Alguns navegadores exigem o preenchimento destes atributos. Portanto, recomenda-se sempre informá-los.

CLASSID: usado para se especificar uma implementação do objeto. Por exemplo, applets Java ou Python e controles ActiveX, todos provêm implementações dos objetos incorporados, que são especificadas através do atributo CLASSID. A seguir são apresentados alguns exemplos.

```
<OBJECT CLASSID="java:ComeCome.class" CODETYPE="application/java" WIDTH="400"
HEIGHT="250" STANDBY="Pronto para jogar Come-Come?" TITLE="Come-Come"></OBJECT>
```

DATA: define a URL do objeto representado pelo elemento.

TYPE: especifica o tipo de mídia do recurso referenciado pelo atributo DATA, evitando que o navegador tente renderizar algo que não suporta.

CODETYPE: especifica o tipo de conteúdo do recurso referenciado pelo atributo CLASSID.

STANDBY: permite definir um texto descritivo que é mostrado enquanto o objeto está sendo carregado.

ARCHIVE: permite especificar nomes de arquivos separados por espaços em branco, possibilitando o navegador de antecipar o download destes arquivos usando a mesma conexão, conseqüentemente, reduzindo o tempo de espera do usuário. Por exemplo, o formato padrão de arquivos Java é JAR.

DECLARE: permite instruir o navegador para que o objeto não seja carregado imediatamente. Desta forma, pode-se ativar o carregamento do objeto através do clique em um link, como mostra o exemplo a seguir.

```
<OBJECT DECLARE="DECLARE" ID="comecome" TITLE="Come-Come" WIDTH="400"
HEIGHT="250" CLASSID="java:ComeCome.class" CODETYPE="application/java"
STANDBY="Pronto para jogar Come-Come?" ></OBJECT>
```

...

```
<P>Pronto para <A HREF="#comecome">jogar Come-Come</A>?</P>
```

USEMAP: usado quando o objeto incorporado é uma imagem que deve conter regiões clicáveis que levam a diferentes destinos. Neste caso, o uso do elemento IMG é recomendado, pois é específico para a exibição de imagens no documento. Para o elemento OBJECT, este atributo deve ser preenchido com o nome (ID ou NAME) do mapa (MAP) a ser usado pelo objeto.

TABINDEX: define a ordem de tabulação do elemento no documento.

8.8 PARAM

O elemento PARAM fornece valores de parâmetros para a exibição de objetos pelo elemento OBJECT. Um elemento OBJECT pode conter vários elementos PARAM. Objetos como vídeos, cliques de áudio e cenários em VRML são manipulados por plug-ins. Cada plug-in reconhece determinados parâmetros. Isso pode dificultar o uso dos elementos OBJECT e PARAM, pois não é possível saber qual plug-in o navegador do usuário usará para exibir determinado conteúdo. Para amenizar este problema, parâmetros não suportados são simplesmente ignorados pelos navegadores. O exemplo a seguir ilustra o uso de PARAM em um elemento OBJECT.

```
<OBJECT DATA="Intro.mov" TYPE="video/quicktime" TITLE="Introdução" WIDTH="150"
HEIGHT="150">
  <PARAM NAME="pluginspage" VALUE="http://quicktime.apple.com/">
  <PARAM NAME="autoplay" VALUE="true">
</OBJECT>
```

- Subelementos: (não se aplica).
- Elementos que o contém: APPLET e OBJECT.

8.8.1 Atributos

O elemento PARAM possui os atributos NAME, VALUE, VALUETYPE, TYPE e ID, descritos a seguir.

NAME: define o nome do parâmetro cujo valor será alterado.

VALUE: define o valor do parâmetro que será alterado.

VALUETYPE: define o tipo de valor do atributo VALUE. O padrão é *data*, portanto, pode-se usar *ref* quando o atributo VALUE especificar uma URL de um arquivo que contenha vários parâmetros para a apresentação do objeto.

TYPE: especifica o tipo de mídia referente ao recurso representado pelo parâmetro.

ID: especifica um identificador único para o parâmetro no documento.

8.9 Q

O elemento Q é usado para citações curtas, que apareçam dentro de frases. Para citações longas, deve-se usar o elemento BLOCKQUOTE.

- Subelementos: elementos de linha.
- Elementos que o contém: elementos de bloco, elementos de linha.

8.9.1 Atributos

O elemento Q possui, além dos atributos comuns, o atributo CITE, descrito a seguir.

CITE: atributo opcional que permite definir uma URL para a fonte da citação em questão.

8.10 SCRIPT

Ver seção 5.5.

8.11 SPAN

O elemento SPAN define uma área genérica de linha de conteúdo, isto é, não insere quebras de linha. Por padrão, ele não muda o visual do seu conteúdo. Portanto, com o uso de folhas de estilo, é possível aplicar diversas características visuais a estes elementos. Desde bordas e imagens de fundo até configurações de fonte podem ser configuradas através de folhas de estilo. O elemento em bloco equivalente ao SPAN é o DIV. O elemento SPAN deve ser utilizado quando nenhum outro elemento de linha HTML provê o efeito desejado. Para definições de negrito e itálico, por exemplo, podem-se utilizar os elementos B e I.

- Subelementos: elementos de linha.
- Elementos que o contém: elementos de bloco, elementos de linha.

O elemento SPAN possui apenas os atributos comuns.

8.12 SUB

O elemento SUB é usado para definir textos subscritos, em fórmulas químicas e matemáticas, por exemplo.

- Subelementos: elementos de linha.

- Elementos que o contém: elementos de bloco, elementos de linha.

O elemento SUB possui apenas os atributos comuns.

8.13 SUP

O elemento SUB é usado para definir textos sobrescritos, em expoentes de matemáticas, por exemplo.

- Subelementos: elementos de linha.
- Elementos que o contém: elementos de bloco, elementos de linha.

O elemento SUP possui apenas os atributos comuns.

8.14 Considerações Sobre o Capítulo

Este capítulo apresentou alguns elementos especiais em nível de linha, ou seja, que podem ser adicionados no meio do documento sem ocupar uma faixa horizontal deste. O próximo capítulo apresenta elementos para apresentação de tabelas, elementos estes usados para apresentação de dados tabulares ou para diagramação do layout de um documento.

9 Elementos para Criação de Tabelas

Este capítulo apresenta os elementos HTML referentes à criação de tabelas. Estes elementos são usados em situações em que se deseja apresentar dados de forma tabular. Muitas pessoas usam tabelas para definir o layout de uma página, dividindo-a em regiões que será ocupadas por elementos como cabeçalho, rodapé, menu lateral etc. Entretanto, esta não é uma prática recomendada, pois existe uma metodologia denominada *tabeless* criada especificamente para definição de layout de páginas.

Os elementos abordados neste capítulo são os elementos TABLE, CAPTION, COLGROUP, COL, THEAD, TFOOT, TBODY, TR, TD e TH, descritos individualmente nas seções a seguir.

9.1 TABLE

O elemento TABLE permite a apresentação de um conteúdo de forma multidimensional, em linhas e colunas. A quantidade de subelementos que um elemento TABLE deve conter depende exclusivamente do conteúdo que se deseja exibir.

- Subelementos: Um elemento opcional CAPTION, seguido de zero ou mais elementos COL e COLGROUP, seguido de um elemento opcional THEAD, um elemento opcional TFOOT e um ou mais elementos TBODY.
- Elementos podem contê-lo: BLOCKQUOTE, BODY, BUTTON, DD, DEL, DIV, FIELDSET, FORM, INS, LI, MAP, NOSCRIPT, OBJECT, TD e TH.

9.1.1 Atributos

Além dos atributos comuns, o elemento TABLE contém os atributos SUMMARY, WIDTH, BORDER, FRAME, RULES, CELSPACING e CELLPADDING, descritos a seguir.

SUMMARY: usado para se definir uma descrição resumida da tabela.

WIDTH: usado para se definir a largura da tabela.

HEIGHT: usado para se definir a altura da tabela.

BORDER: usado para se definir a largura da borda de uma tabela.

FRAME: usado para se definir quais bordas externas deverão ser apresentadas. Pode conter um dos valores apresentados a seguir:

- void: sem borda;
- above: somente superior;
- below: somente inferior;
- hside: somente esquerda e direita;
- lhs: somente esquerda;
- rhs: somente direita;
- vside: superior e inferior;
- box ou border: todas as bordas.

RULES: usado para se definir quais bordas internas (entre as células) deverão ser apresentadas. Este atributo não é suportado de forma regular pelos navegadores. Pode conter um dos valores apresentados a seguir:

- none: sem borda;
- groups: somente em torno de grupos de colunas ou grupos de linhas;
- rows: somente em torno de linhas;
- cols: somente em torno de colunas;
- all: em torno de todas as células.

CELLSPACING: usado para se definir o espaçamento em pixels entre as células (margem externa da célula).

CELLPADDING: usado para se definir o espaçamento em pixels entre uma célula e seu conteúdo (margem interna da célula).

9.2 CAPTION

O elemento CAPTION é usado para se definir um título para a tabela. Quando for usado, deve ser o primeiro subelemento do elemento TABLE.

- Subelementos: elementos de linha.
- Elementos que o contém: TABLE.

O elemento CAPTION possui apenas os atributos comuns.

9.3 COLGROUP

O elemento COLGROUP é usado para se agrupar um conjunto de colunas adjacentes de uma tabela. Quando usado, deve figurar após o elemento CAPTION e antes do elemento THEAD, e sempre dentro do elemento TABLE. É possível usar COLGROUP combinado com folhas de estilos para causar um efeito diferenciado sobre o grupo de colunas.

- Subelementos: zero ou mais elementos COL.
- Elementos que o contém: TABLE.

9.3.1 Atributos

Além dos atributos comuns, o elemento COLGROUP contém os atributos SPAN, WIDTH, ALIGN, CHAR, CHAROFF e VALIGN, descritos a seguir.

SPAN: define o número de colunas abrangidas pelo grupo.

WIDTH: define a largura de cada coluna pertencente ao grupo.

ALIGN: define o alinhamento horizontal do conteúdo das células do grupo. Os possíveis valores são:

- left: alinhamento à esquerda;
- center: alinhamento centralizado;
- right: alinhamento à direita;
- justify: alinhamento justificado;
- char: alinhamento a partir do caractere especificado no atributo CHAR.

CHAR: define o caractere a ser usado para se alinhar um conteúdo usando o atributo ALIGN com o valor “char”.

CHAROFF: define o valor do recuo a ser aplicado no alinhamento usando ALIGN com o valor “char”.

VALIGN: define o alinhamento vertical do conteúdo das células do grupo. Os possíveis valores são:

- top: alinhamento superior;
- middle: alinhamento centralizado verticalmente;
- bottom: alinhamento inferior;
- baseline: alinhamento da primeira linha de conteúdo baseado em uma linha imaginária que passa por todas as células de uma linha.

9.4 COL

O elemento COL é usado para se definir atributos comuns a colunas de uma tabela. Quando usado, COL deve figurar após o elemento opcional CAPTION e antes do elemento opcional THEAD, e sempre como um subelemento de TABLE. COL pode estar contido diretamente na tabela, e não apenas dentro de COLGROUP. Ao contrário de COLGROUP, COL não agrupa colunas, mas apenas define atributos comuns a células de uma ou mais colunas.

- Subelementos: (não se aplica).
- Elementos que o contém: COLGROUP e TABLE.

9.4.1 Atributos

Além de possuir os atributos comuns, o elemento COL possui ainda os atributos SPAN, WIDTH, ALIGN, CHAR, CHAROFF e VALIGN, descritos a seguir.

SPAN: define quantas colunas serão configuradas pelos atributos definidos em COL.

WIDTH: define a largura de cada coluna.

ALIGN: define o alinhamento horizontal do conteúdo das células do grupo. Os possíveis valores são:

- left: alinhamento à esquerda;
- center: alinhamento centralizado;
- right: alinhamento à direita;
- justify: alinhamento justificado;
- char: alinhamento a partir do caractere especificado no atributo CHAR.

CHAR: define o caractere a ser usado para se alinhar um conteúdo usando o atributo ALIGN com o valor “char”.

CHAROFF: define o valor do recuo a ser aplicado no alinhamento usando ALIGN com o valor “char”.

VALIGN: define o alinhamento vertical do conteúdo das células do grupo. Os possíveis valores são:

- top: alinhamento superior;
- middle: alinhamento centralizado verticalmente;

- bottom: alinhamento inferior;
- baseline: alinhamento da primeira linha de conteúdo baseado em uma linha imaginária que passa por todas as células de uma linha.

9.5 THEAD

O elemento THEAD define uma ou mais linhas que serão consideradas cabeçalho da tabela. Um elemento TABLE deve conter apenas um elemento THEAD, o qual deve figurar após um elemento CAPTION, COL ou COLGROUP, e precede o atributo opcional TFOOT e o elemento necessário TBODY. Usando THEAD para se definir o cabeçalho de uma tabela, permitindo que o navegador, ao imprimir o conteúdo da tabela, repita o cabeçalho caso a tabela seja quebrada em mais de uma página. Além disso, pode-se definir o corpo da tabela com barra de rolagem e deixar o cabeçalho fixo.

- Subelementos: Um ou mais elementos TR.
- Elementos que o contém: TABLE.

9.5.1 Atributos

Além de possuir os atributos comuns, o elemento THEAD possui ainda os atributos ALIGN, CHAR, CHAROFF e VALIGN, descritos a seguir.

ALIGN: define o alinhamento horizontal do conteúdo das células do grupo. Os possíveis valores são:

- left: alinhamento à esquerda;
- center: alinhamento centralizado;
- right: alinhamento à direita;
- justify: alinhamento justificado;
- char: alinhamento a partir do caractere especificado no atributo CHAR.

CHAR: define o caractere a ser usado para se alinhar um conteúdo usando o atributo ALIGN com o valor “char”.

CHAROFF: define o valor do recuo a ser aplicado no alinhamento usando ALIGN com o valor “char”.

VALIGN: define o alinhamento vertical do conteúdo das células do grupo. Os possíveis valores são:

- top: alinhamento superior;
- middle: alinhamento centralizado verticalmente;
- bottom: alinhamento inferior;
- baseline: alinhamento da primeira linha de conteúdo baseado em uma linha imaginária que passa por todas as células de uma linha.

9.6 TFOOT

O elemento TFOOT define uma ou mais linhas que serão consideradas como rodapé da tabela. Um elemento TABLE deve conter apenas um elemento TFOOT, o qual deve figurar após o elemento THEAD e antes do elemento TBODY. Usando TFOOT para se definir o cabeçalho de uma tabela, permite-se que o navegador, ao imprimir o conteúdo da tabela, repita o cabeçalho caso a tabela seja quebrada em mais de uma página. Além disso, pode-se definir o corpo da tabela com barra de rolagem e deixar o rodapé fixo.

- Subelementos: Um ou mais elementos TR.
- Elementos que o contém: TABLE.

9.6.1 Atributos

Além de possuir os atributos comuns, o elemento TFOOT possui ainda os atributos ALIGN, CHAR, CHAROFF e VALIGN, descritos a seguir.

ALIGN: define o alinhamento horizontal do conteúdo das células do grupo. Os possíveis valores são:

- left: alinhamento à esquerda;
- center: alinhamento centralizado;
- right: alinhamento à direita;
- justify: alinhamento justificado;
- char: alinhamento a partir do caractere especificado no atributo CHAR.

CHAR: define o caractere a ser usado para se alinhar um conteúdo usando o atributo ALIGN com o valor “char”.

CHAROFF: define o valor do recuo a ser aplicado no alinhamento usando ALIGN com o valor “char”.

VALIGN: define o alinhamento vertical do conteúdo das células do grupo. Os possíveis valores são:

- top: alinhamento superior;
- middle: alinhamento centralizado verticalmente;
- bottom: alinhamento inferior;

baseline: alinhamento da primeira linha de conteúdo baseado em uma linha imaginária que passa por todas as células de uma linha.

9.7 TBODY

O elemento TBODY é usado para agrupar as linhas de dados de uma tabela. Uma tabela deve ter um ou mais elementos TBODY, que por sua vez deve figurar logo após o elemento TFOOT. Usando agrupamento de linhas com TBODY, podem-se utilizar folhas de estilo para se definir uma formatação visual diferente para cada grupo.

- Subelementos: Um ou mais elementos TR.
- Elementos que o contém: TABLE.

9.7.1 Atributos

Além de possuir os atributos comuns, o elemento TBODY possui ainda os atributos ALIGN, CHAR, CHAROFF e VALIGN, descritos a seguir.

ALIGN: define o alinhamento horizontal do conteúdo das células do grupo. Os possíveis valores são:

- left: alinhamento à esquerda;
- center: alinhamento centralizado;
- right: alinhamento à direita;
- justify: alinhamento justificado;

- char: alinhamento a partir do caractere especificado no atributo CHAR.

CHAR: define o caractere a ser usado para se alinhar um conteúdo usando o atributo ALIGN com o valor “char”.

CHAROFF: define o valor do recuo a ser aplicado no alinhamento usando ALIGN com o valor “char”.

VALIGN: define o alinhamento vertical do conteúdo das células do grupo. Os possíveis valores são:

- top: alinhamento superior;
- middle: alinhamento centralizado verticalmente;
- bottom: alinhamento inferior;
- baseline: alinhamento da primeira linha de conteúdo baseado em uma linha imaginária que passa por todas as células de uma linha.

9.8 TR

O elemento TR define uma linha de uma tabela. É recomendado que TR esteja contido dentro de algum dos elementos de agrupamento de linha (THEAD, TFOOT e TBODY). Quando THEAD e TFOOT não são definidos na tabela, o elemento TBODY pode ser suprimido, colocando-se os elementos TR diretamente dentro de TABLE.

- Subelementos: um ou mais elementos TD ou TH.
- Elementos que o contém: THEAD, TFOOT, TBODY e TABLE.

9.8.1 Atributos

Além de possuir os atributos comuns, o elemento TR possui ainda os atributos ALIGN, CHAR, CHAROFF e VALIGN, descritos a seguir.

ALIGN: define o alinhamento horizontal do conteúdo das células do grupo. Os possíveis valores são:

- left: alinhamento à esquerda;
- center: alinhamento centralizado;
- right: alinhamento à direita;
- justify: alinhamento justificado;
- char: alinhamento a partir do caractere especificado no atributo CHAR.

CHAR: define o caractere a ser usado para se alinhar um conteúdo usando o atributo ALIGN com o valor “char”.

CHAROFF: define o valor do recuo a ser aplicado no alinhamento usando ALIGN com o valor “char”.

VALIGN: define o alinhamento vertical do conteúdo das células do grupo. Os possíveis valores são:

- top: alinhamento superior;
- middle: alinhamento centralizado verticalmente;
- bottom: alinhamento inferior;
- baseline: alinhamento da primeira linha de conteúdo baseado em uma linha imaginária que passa por todas as células de uma linha.

9.9 TD

O elemento TD define uma célula de dados de uma tabela. Elementos TD devem figurar dentro de elementos TR, que por sua vez podem conter também elementos TH, que definem cabeçalhos de linhas ou colunas.

- Subelementos: elementos de linha e elementos de bloco.
- Elementos que o contém: TR.

9.9.1 Atributos

Além dos atributos comuns, o elemento TD possui o elemento ROWSPAN, COLSPAN, HEADERS, ABBR, SCOPE, ALIGN, CHAR, CHAROFF e VALIGN, descritos a seguir.

ROWSPAN e COLSPAN: definem, respectivamente, o número de linhas e o número de colunas que serão mescladas pela célula. Uma coluna mesclada ocupa o espaço de duas ou mais células. O valor deste atributo deve ser a quantidade de células a mesclar. O valor especial 0 (zero) define que a célula se expandirá até a última linha ou coluna da tabela.

HEADERS: permite definir os IDs dos elementos TH que são cabeçalhos da célula, separados por espaços.

ABBR: define uma versão abreviada do conteúdo da célula. Só deve ser usado se a tabela possuir informações de cabeçalho.

SCOPE: define as células para as quais a célula em questão será cabeçalho. Trata-se de uma alternativa ao uso de elementos de cabeçalho (TH e THEAD). Os possíveis valores do atributo são:

- row: quando a célula serve de cabeçalho para o resto da linha;
- col: quando a célula serve de cabeçalho para o resto da coluna;
- row-group: quando a célula serve de cabeçalho para o resto do grupo de linhas;
- col-group: quando a célula serve de cabeçalho para o resto do grupo de colunas.

ALIGN: define o alinhamento horizontal do conteúdo das células do grupo. Os possíveis valores são:

- left: alinhamento à esquerda;
- center: alinhamento centralizado;
- right: alinhamento à direita;
- justify: alinhamento justificado;
- char: alinhamento a partir do caractere especificado no atributo CHAR.

CHAR: define o caractere a ser usado para se alinhar um conteúdo usando o atributo ALIGN com o valor “char”.

CHAROFF: define o valor do recuo a ser aplicado no alinhamento usando ALIGN com o valor “char”.

VALIGN: define o alinhamento vertical do conteúdo das células do grupo. Os possíveis valores são:

- top: alinhamento superior;
- middle: alinhamento centralizado verticalmente;
- bottom: alinhamento inferior;

- baseline: alinhamento da primeira linha de conteúdo baseado em uma linha imaginária que passa por todas as células de uma linha.

9.10TH

O elemento TH define uma célula de cabeçalho de uma tabela. Elementos TH devem figurar dentro de elementos TR, que por sua vez podem conter também elementos TD, que definem dados de tabelas.

- Subelementos: elementos de linha e elementos de bloco.
- Elementos que o contém: TR.

9.10.1 Atributos

Além dos atributos comuns, o elemento TH possui o elemento ROWSPAN, COLSPAN, HEADERS, ABBR, SCOPE, ALIGN, CHAR, CHAROFF e VALIGN, descritos a seguir.

ROWSPAN e COLSPAN: definem, respectivamente, o número de linhas e o número de colunas que serão mescladas pela célula. Uma coluna mesclada ocupa o espaço de duas ou mais células. O valor deste atributo deve ser a quantidade de células a mesclar. O valor especial 0 (zero) define que a célula se expandirá até a última linha ou coluna da tabela.

HEADERS: permite definir os IDs dos elementos TH que são cabeçalhos da célula, separados por espaços.

ABBR: define uma versão abreviada do conteúdo da célula. Só deve ser usado se a tabela possuir informações de cabeçalho.

SCOPE: define as células para as quais a célula em questão será cabeçalho. Trata-se de uma alternativa ao uso de elementos de cabeçalho (TH e THEAD). Os possíveis valores do atributo são:

- row: quando a célula serve de cabeçalho para o resto da linha;
- col: quando a célula serve de cabeçalho para o resto da coluna;
- row-group: quando a célula serve de cabeçalho para o resto do grupo de linhas;
- col-group: quando a célula serve de cabeçalho para o resto do grupo de colunas.

ALIGN: define o alinhamento horizontal do conteúdo das células do grupo. Os possíveis valores são:

- left: alinhamento à esquerda;
- center: alinhamento centralizado;
- right: alinhamento à direita;
- justify: alinhamento justificado;
- char: alinhamento a partir do caractere especificado no atributo CHAR.

CHAR: define o caractere a ser usado para se alinhar um conteúdo usando o atributo ALIGN com o valor "char".

CHAROFF: define o valor do recuo a ser aplicado no alinhamento usando ALIGN com o valor "char".

VALIGN: define o alinhamento vertical do conteúdo das células do grupo. Os possíveis valores são:

- top: alinhamento superior;
- middle: alinhamento centralizado verticalmente;

- bottom: alinhamento inferior;
- baseline: alinhamento da primeira linha de conteúdo baseado em uma linha imaginária que passa por todas as células de uma linha.

9.11 Exemplo

```
<TABLE SUMMARY="Este é o resumo do conteúdo da tabela.">
  <CAPTION>Título da tabela</CAPTION>
  <COLGROUP CLASS="classe-css-grupo1">
  <COLGROUP CLASS="classe-css-grupo2" SPAN="5">
  <THEAD>
    <TR>
      <TH SCOPE="col">TituloColuna1</TH>
      <TH SCOPE="col">TituloColuna2</TH>
      <TH SCOPE="col">TituloColuna3</TH>
      <TH SCOPE="col">TituloColuna4</TH>
      <TH SCOPE="col">TituloColuna5</TH>
    </TR>
  </THEAD>
  <TFOOT CLASS="classe-css-rodape">
    <TR>
      <TD COLSPAN="5">
        Este é o texto da célula de rodapé, que ocupa 5 colunas.
      </TD>
    </TR>
    <TR>
      <TD COLSPAN="5">
        Segunda linha de texto do rodapé.
      </TD>
    </TR>
  </TFOOT>
  <TBODY>
    <TR>
      <TD SCOPE="row">Dados Coluna 1</TD>
      <TD>Dados Coluna 2</TD>
      <TD>Dados Coluna 3</TD>
      <TD>Dados Coluna 4</TD>
      <TD>Dados Coluna 5</TD>
    </TR>
    ...
  </TBODY>
</TABLE>
```

As reticências antes do fechamento do elemento TBODY (</TBODY>) seriam ocupadas por outras linhas de dados.

9.12 Considerações Sobre o Capítulo

Este capítulo apresentou alguns elementos especiais em nível de linha, ou seja, que podem ser adicionados no meio do documento sem ocupar uma faixa horizontal deste. O próximo capítulo apresenta elementos para apresentação de formulários e campos, elementos estes usados para permitir que o usuário envie dados para o servidor para que estes sejam devidamente processados.

10 Elementos para Criação de Formulários

Este capítulo apresenta os elementos HTML referentes à criação de formulários. Os formulários são utilizados para permitir que o usuário envie dados ao servidor, sejam estes relativos ao preenchimento de algum tipo de cadastro ou alguma outra informação que deva ser processada e/ou armazenada no servidor.

Os elementos abordados neste capítulo são os elementos FORM, BUTTON, FIELDSET, LEGEND, INPUT, LABEL, SELECT, OPTGROUP, OPTION e TEXTAREA, descritos individualmente nas seções a seguir.

10.1 FORM

O elemento FORM permite a apresentação de um formulário que pode ser submetido ao servidor para processamento.

- Subelementos: Qualquer elemento em nível de bloco, exceto FORM.
- Elementos podem contê-lo: BLOCKQUOTE, BODY, DD, DEL, DIV, FIELDSET, INS, LI, MAP, NOSCRIPT, OBJECT, TD e TH.

10.1.1 Atributos

Além dos atributos comuns, o elemento FORM contém os atributos ACTION, METHOD, ACCEPT-CHARSET e ENCTYPE, descritos a seguir.

ACTION: define a URL da ação a ser executada ao submeter o formulário, ou seja, quem processará os dados enviados pelo formulário.

METHOD: usado para se definir a forma de envio dos dados. Pode ter os valores GET ou POST, sendo que o primeiro envia os dados pela URL e o segundo envia através de um pacote de dados.

ACCEPT-CHARSET: usado para se definir a codificação de caracteres de texto usada pelo formulário.

10.1.2 Atributos de Eventos

Os atributos de eventos do elemento FORM são ONSUBMIT e ONRESET, descritos a seguir.

ONSUBMIT: ocorre quando o formulário é enviado.

ONRESET: ocorre quando o formulário tem seus campos apagados (redefinidos para vazio).

10.2 BUTTON

Elemento presente a partir do HTML 4.0, o elemento BUTTON permite a apresentação de um botão submeter os dados do formulário ao servidor para processamento. Pode ser usado também para limpar os campos do formulário ou para executar alguma outra ação, em JavaScript, por exemplo. Substitui o elemento INPUT do tipo BUTTON, apresentando alguns avanços, como uso de imagens, por exemplo. Pode-se usar vários elementos BUTTON em um mesmo formulário, passando valores diferentes para os atributos NAME e VALUE, de forma que se possa executar uma ação diferente de acordo com o botão clicado para enviar o formulário.

- Subelementos: Elementos em nível de linha, exceto A, INPUT, SELECT, TEXTAREA, LABEL, BUTTON, e IFRAME, e elementos em nível de bloco, exceto FORM, ISINDEX, e FIELDSET.
- Elementos podem contê-lo: Elementos em nível de bloco e em nível de linha, exceto BUTTON.

10.2.1 Atributos

Além dos atributos comuns, o elemento BUTTON contém os atributos NAME, VALUE, TYPE, DISABLED, ACCESSKEY e TABINDEX, descritos a seguir.

NAME: define o nome do campo que será enviado junto aos pares nome/valor do formulário.

VALUE: define o nome do campo que será enviado junto aos pares nome/valor do formulário.

TYPE: define o tipo de botão, podendo ser: SUBMIT, para submeter o formulário ao servidor; RESET, para limpar os campos do formulário; ou BUTTON, para executar diferente das ações anteriores.

DISABLED: define o botão como desabilitado, impossibilitando-o de receber foco ou de ser clicado.

ACCESSKEY: define um caractere de atalho para o botão.

TABINDEX: define a ordem do elemento quando o usuário navega pelos elementos do formulário usando a tecla TAB.

10.2.2 Atributos de Eventos

Os atributos de eventos do elemento FORM são ONFOCUS e ONBLUR, descritos a seguir.

ONFOCUS: ocorre quando o elemento recebe o foco.

ONBLUR: ocorre quando o elemento perde o foco.

10.3 FIELDSET

O elemento FIELDSET define um grupo de elementos em um formulário. Este grupo pode ser configurado separadamente, podendo-se atribuir estilos específicos a ele, por exemplo.

- Subelementos: Um elemento LEGEND seguido por zero ou mais elementos em nível de bloco e/ou elementos em nível de linha.
- Elementos que podem contê-lo: BLOCKQUOTE, BODY, CENTER, DD, DEL, DIV, FIELDSET, FORM, IFRAME, INS, LI, MAP, NOFRAMES, NOSCRIPT, OBJECT, TD e TH.

10.3.1 Atributos

Além dos atributos comuns, o elemento BUTTON contém os atributos NAME, VALUE, TYPE, DISABLED, ACCESSKEY e TABINDEX, descritos a seguir.

NAME: define o nome do campo que será enviado junto aos pares nome/valor do formulário.

VALUE: define o nome do campo que será enviado junto aos pares nome/valor do formulário.

TYPE: define o tipo de botão, podendo ser: SUBMIT, para submeter o formulário ao servidor; RESET, para limpar os campos do formulário; ou BUTTON, para executar diferente das ações anteriores.

DISABLED: define o botão como desabilitado, impossibilitando-o de receber foco ou de ser clicado.

ACCESSKEY: define um caractere de atalho para o botão.

TABINDEX: define a ordem do elemento quando o usuário navega pelos elementos do formulário usando a tecla TAB.

10.4 LEGEND

O elemento LEGEND define um título para grupo de elementos em um formulário definido pelo elemento FIELDSET. O elemento LEGEND deve figurar como o primeiro elemento interno ao FIELDSET.

- Subelementos: Elementos em nível de linha.
- Elementos que podem contê-lo: FIELDSET.

10.4.1 Atributos

Além dos atributos comuns, o elemento LEGEND possui o atributo ACCESSKEY, define um caractere de atalho para o grupo que o usa como título.

10.5 INPUT

O elemento INPUT permite apresentar um controle de entrada de dados em um formulário. Apesar de ser mais utilizado dentro de formulários, o elemento INPUT pode .

- Subelementos: Elementos em nível de linha, exceto A, INPUT, SELECT, TEXTAREA, LABEL, BUTTON, e IFRAME, e elementos em nível de bloco, exceto FORM, ISINDEX, e FIELDSET.
- Elementos podem contê-lo: Elementos em nível de bloco e em nível de linha, exceto BUTTON.

10.5.1 Atributos

Além dos atributos comuns, o elemento BUTTON contém os atributos NAME, VALUE, TYPE, DISABLED, ACCESSKEY e TABINDEX, descritos a seguir.

NAME: define o nome do campo que será enviado junto aos pares nome/valor do formulário.

VALUE: define o nome do campo que será enviado junto aos pares nome/valor do formulário.

TYPE: define o tipo de botão, podendo ser: SUBMIT, para submeter o formulário ao servidor; RESET, para limpar os campos do formulário; ou BUTTON, para executar diferente das ações anteriores.

DISABLED: define o botão como desabilitado, impossibilitando-o de receber foco ou de ser clicado.

ACCESSKEY: define um caractere de atalho para o botão.

TABINDEX: define a ordem do elemento quando o usuário navega pelos elementos do formulário usando a tecla TAB.

10.5.2 Atributos de Eventos

Os atributos de eventos do elemento FORM são ONFOCUS e ONBLUR, descritos a seguir.

ONFOCUS: ocorre quando o elemento recebe o foco.

ONBLUR: ocorre quando o elemento perde o foco.

