Faculdade de Economia, Administração e Contabilidade da USP

EAD-451 – Informática Aplicada à Administração

Internet

Construção de Páginas Web

com HTML e Scripts

Prof. Antonio Geraldo da Rocha Vidal

SUMÁRIO

Introdução à Linguagem HTML

Hypertext Markup Language (HTML) é a linguagem usada para criar páginas de hipertexto na Web. Ela utiliza um conjunto de marcadores (ou tags) para identificar as partes do documento ou página e definir como o texto aparecerá quando visto em um navegador ou *Browser* Web.

Cada página HTML é composta de duas partes: o cabeçalho e um corpo opcional. Os cabeçalhos das páginas HTML provêem informações gerais sobre a página, como tipo de página, identificação do servidor e capacidade, informações administrativas e descrição do corpo. O corpo da página, se presente, contém um bloco de informações representando os dados que serão apresentados no navegador.

O exemplo a seguir é uma página HTML bem simples:

```
<HTML>
<HEAD>
<TITLE>Exemplo de página HTML</TITLE>
</HEAD>
<BODY>
<H1>Ola, mundo!</H1>

Este é um documento HTML bem simples.

</BODY>
</HTML>
```

As páginas Web podem ser ou estáticas ou dinâmicas. Uma página Web estática é uma página HTML que pode ter hyperlinks para outras páginas e arquivos, mas não atualiza dados no servidor Web e não é atualizada pelas ações ou informações do usuário sobre o servidor. A maioria das páginas Web ainda são estáticas.

Uma página Web dinâmica ou ativa fornece meios para o usuário interagir com um servidor Web, atualizando ou alterando dados armazenados no servidor. Por exemplo, uma página Web ativa é normalmente constituída por um formulário, como um formulário de ordem ou compra on-line, através do qual o usuário acessa e submete informações para um servidor Web em uma aplicação de comércio eletrônico. Uma página Web ativa pode conter também controles ActiveX ou programas escritos em linguagens de scripting, que fornecem à página capacidades de processamento de informações.

Ferramentas

Para um desenvolvedor de aplicações profissional, as ferramentas ou produtos mais significantes oferecidos para o ambiente Web são aqueles que permitem criar, publicar, ou navegar por páginas Web. Neste curso abordaremos essencialmente as ferramentas desenvolvidas pela Microsoft pois, entre outras razões, estão atualmente entre as mais utilizadas pelos profissionais que desenvolvem páginas Web.

Criação de conteúdo

?? **Microsoft Office**

Você pode usar quaisquer das aplicações do Microsoft Office para editar documentos HTML existentes ou criar novos. Os aplicativos do Office são uma excelente solução para a criação de páginas Web simples.

?? **Microsoft FrontPage**

Você pode usar o Microsoft FrontPage para criar páginas Web sofisticadas e para administrar um Web Site integrado.

?? **Microsoft Visual InterDev**

Você pode usar o Visual InterDev para construir Web Sites dinâmicos. Com recursos para desenvolvimento visual e poderosas ferramentas para acesso a bases de dados, o Visual InterDev fornece um completo ambiente de desenvolvimento, tecnicamente avançado, por permitir a construção tanto de aplicações Internet como Intranet.

Servidor Web

?? **Servidor de Informações Internet da Microsoft (IIS)**

O Servidor de Informações Internet da Microsoft (IIS) é totalmente integrado com o sistema operacional Windows NT Server, além de ser especialmente projetado para disponibilizar uma grande variedade de recursos como servidor Intranet.

?? **Servidor Web Pessoal da Microsoft**

O Servidor Web Pessoal da Microsoft (PWS) é projetado para aplicações de pequena escala, sendo ideal para em ambientes de desenvolvimento ou testes e para a execução de aplicações Web de pequena escala.

Navegador ou Browser

?? **Internet Explorer da Microsoft**

O Microsoft Internet Explorer da Microsoft é uma das ferramentas atualmente mais utilizadas para visualizar informações, estáticas e dinâmicas, na Internet. Em conjunto com o NetScape Navigator são as ferramentas mais utilizadas para visualização de páginas HTML.

Gerenciador de Banco de Dados Microsoft SQL Server

?? **Microsoft SQL Server**

O gerenciador de Banco de Dados Microsoft SQL Server está totalmente integrado ao sistema operacional Windows NT Server, ao servidor Web Internet Information Server e é, portanto, a ferramenta adequada para armazenar dados e conteúdo ativo a ser apresentado em páginas Web. O Microsoft Transaction Server (MTS) é um sistema gerenciador de transações, também integrado ao Windows NT Server, ideal para o

gerenciamento de transações e execução da lógica e de regras de negócio necessárias às páginas Web ativas.

?? Outros Bancos de Dados

Interatividade

Você pode usar o HTML Dinâmico para criar páginas ricas, interativas que exigem menos comunicação com o servidor para exibir seu conteúdo. As características do HTML Dinâmico permitem o processamento interativo de páginas Web no cliente sem que seja necessário fazer inúmeros pedidos ao servidor Web para mais informações:

?? **Controles de multimídia**

Um conjunto de controles multimídia é automaticamente instalado com o navegador Internet Explorer 4.0 ou superior. Estes controles fornecem efeitos gráficos especiais, como arte de transições e texto, sem exigir transferência de imagens através da rede. Além disso, o HTML Dinâmico suporta gráficos estruturados, um tipo de controle multimídia que descreve um *metafile*. Estes controles produzem gráficos pequenos que parecem melhores que imagens as tradicionais.

?? **Cache Local de Dados**

Antes do HTML Dinâmico ser desenvolvido, sempre era necessário para contactar o servidor Web e/ou o servidor de base de dados quando o usuário desejasse ter uma visão diferente dos dados em uma página. Utilizando o HTML Dinâmico, os desenvolvedores podem usar conexões de dados para embutir uma fonte de dados em uma página. Com isso, os usuários podem classificar e filtrar a fonte de dados como se estivessem acessando uma base de dados, sem, entretanto, permanecerem conectados no servidor de base de dados original.

?? **Outlining**

O recurso de Outlining mostra ou esconde trechos de texto baseado em eventos do usuário como apontando ou clicando o mouse. Por exemplo, você pode usar o outlining para criar um índice dinâmico, ou para fazer um texto escondido aparecer a partir de um cabeçalho, quando um usuário selecionar este cabeçalho. O HTML dinâmico pode criar dinamicamente um novo trecho de código HTML em uma página Web, em resposta a uma ação do usuário, sem ter que se comunicar com o servidor. Esse recurso permite que os usuários abram e exibam menos páginas para achar as informações que precisam. Realizando a maioria do processamento no lado de cliente o HTML Dinâmico melhora significativamente desempenho das páginas Web.

Projeto de Páginas Web

Ao projetar a interface do usuário para uma página Web, você deve considerar as seguintes recomendações:

?? Minimize o tempo de transferência (*download*) das páginas:

Mantenha seu projeto de página simples. Minimize o uso de gráficos complexos que tomam muito tempo para serem carregados.

?? Mantenha as informações acessíveis:

As principais informações de um Web Site devem estar no máximo dois hyperlinks abaixo da Home Page, ou seja, sua página inicial. Os usuários ficam freqüentemente confusos e frustrados quando precisam navegar por mais de dois hyperlinks para alcançar as informações importantes.

?? Forneça mídia alternativa:

Alguns navegadores podem não ter recursos para suportar toda a funcionalidade de uma página, como por exemplo, gráficos, vídeos e arquivos de som. Se você fornecer um método alternativo para apresentar informações semelhantes (um gráfico em vez de um arquivo de vídeo, ou um texto em lugar de um gráfico), mais usuários poderão visitar seu Web Site. Além disso, esta estratégia fornece também compatibilidade com navegadores mais antigos.

?? Minimize o uso de som:

Transferir arquivos de som através da Internet exige atualmente uma razoável capacidade de comunicação. Isso significa que usuários com conexões de baixa capacidade levarão muito tempo para carregar páginas que tocam muitos sons.

?? Coordenação de cores:

Se você utilizar um arquivo padrão bitmap como figura de fundo, procure ter certeza de que ele não dificultará a leitura da página. Alguns arquivos bitmap de fundo tornam o texto das páginas Web simplesmente ilegíveis.

?? Páginas para Intranet:

Se você estiver projetando páginas Web para uma Intranet, deverá focar a facilitação do acesso a informações críticas para os usuários (como formulários e bases de dados corporativas).

?? Páginas para Internet:

Se você estiver projetando páginas Web para a Internet, deverá focar em informações corporativas e institucionais e mensagens de marketing. Em ambos os casos, você deve se esforçar para criar um Web Site com recursos de navegação globais e claros, de forma que usuários possam mover-se através do seu site e achar as informações que procuram com facilidade.

A seguir são relacionadas algumas dicas para manter suas páginas Web adequadas:

?? Mantenha as páginas com informações atualizadas.

Atualize sua página Web sempre que as informações nela mudarem. Por exemplo, tenha certeza que os hyperlinks para outras páginas estão atualizados e funcionando.

?? Adicione ponteiros para utilizar recursos de navegadores atualizados;

Isto é especialmente importante se sua página fizer uso de recursos HTML avançados.

?? Informe aos usuários de problemas com sua página.

Se sua página está sendo modificada, adicione um gráfico ou texto indicando onde as modificações se iniciam e terminam.

?? Mantenha hyperlinks para sites antigos.

Se você mover seu site para um novo servidor, mantenha uma página para fornecer um hyperlink do site antigo para o novo, durante pelo menos uns três meses.

?? Defina a Home Page para a página padrão do servidor Web.

Sempre defina o nome da sua Home Page padrão, normalmente default.htm ou index.htm, de forma a estar compatível com o nome atualmente configurado na instalação de seu servidor Web. Isso torna mais fácil para os usuários localizar sua página quando se conectarem a seu servidor.

Código HTML

A Hypertext Markup Language (HTML) é a codificação atrás das páginas Web normais. A HTML usa marcadores ou tags para instruir um navegador Web (*browser*) a como apresentar as informações na página. Por exemplo, existem tags para cabeçalhos, gráficos, parágrafos e hyperlinks.

Quando você usa um software de autoria Web, como o FrontPage da Microsoft, o Composer da Netscape ou outro equivalente para criar páginas Web, ele insere automaticamente as tags HTML apropriadas em seu documento. Porém, você também pode criar arquivos HTML com qualquer editor de textos padrão ASCII, como Microsoft Notepad.

Se você pretende criar páginas Web simples, contendo apenas texto e gráficos estáticos, não é necessário conhecer a sintaxe da linguagem HTML, pois um software de autoria como o Front Page ou o Composer cria o código automaticamente para você. Mas se você quiser criar paginas Web que utilizem recursos avançados, como automação através de scripts escritos em Java (JavaScript ou JScript) ou Visual Basic (VBScript), controles de ActiveX, ou tags HTML avançadas, você precisa usar um editor de texto para programar as tags HTML manualmente.

Atrás de toda página Web está um documento que é formatado com tags HTML. Estas tags dizem ao navegador como exibir o texto, as imagens, os hyperlinks, e todos os outros elementos contidos na página.

Marcadores (TAGS) HTML

Todas páginas Web compartilham várias tags básicas:

?? `<HTML>`

A tag <HTML> indica que o documento é um documento HTML.

?? <HEAD>

A tag <HEAD> define a seção de cabeçalho, que pode conter uma variedade de outras tags, mas a única que é exigida e obrigatória é a tag <TITLE>, que especifica o título da página Web, a ser exibido na barra de título do navegador quando um usuário visualizar o documento.

?? <BODY>

A tag <BODY> especifica o corpo do documento. O corpo contém as tags e o texto que formam o conteúdo principal da página Web.

A ilustração a seguir mostra um exemplo de uma página HTML extremamente simples.

Este é o código HTML por trás desta página:

```
<HTML>
<HEAD><meta name="GENERATOR" content="Microsoft FrontPage 3.0">
<TITLE>ABC Company Home Page</TITLE>
</HEAD>

<BODY>
<!-- Comentário: Esta seção contém as tags e o texto que formam o corpo
da página Web -->
Hello, World!
</BODY>
</HTML>
```

As tags HTML sempre são iniciadas pelo símbolo < , seguido pelo nome da tag, e são finalizadas com o símbolo >. A maioria dos elementos ou conteúdos das páginas Web estão contidos dentro de tags emparelhadas, que marcam seu início e fim (por exemplo, <H1> e </H1>). A tag de fim é idêntica à tag de início, a não ser pelo símbolo / que precede o nome da tag dentro dos parênteses. Tipicamente, você coloca um conteúdo de uma página entre as tags de início e fim.

A tabela a seguir mostra as tags HTML mais utilizadas nas páginas Web:

Característica	TAG Exemplo
Cabeçalho	<H1>...</H1> a <H6>... </H6>
Quebra de linha	
Artigo de lista	<Li>...</LI>
Texto enfatizado	<EM>...</EM>
Imagem	<IMG>
Texto de Pré-formatado	<PRE>...</PRE>
Texto em Negrito	<B>...</B>
Parágrafo	<P>

Nota: as tags quebra de linha
, imagem <IMG> e de parágrafo <P> não têm uma tag de fim (a tag de parágrafo pode ter uma tag de fim </P>, mas é freqüentemente omitida.

Atributos de tags

Dentro de uma tag, você pode fixar atributos para definir características sobre o texto nelas contido. Os atributos formatam o conteúdo da tag, mudando o modo como o elemento aparece. A amostra de código a seguir usa os atributos de FACE e SIZE para definir o estilo das letras do texto:

```
<FONT FACE="Arial" SIZE="2">text in 10pt Arial font.</FONT>
```

Vendo o código fonte HTML no Navegador

Você pode ver o código fonte HTML de qualquer página Web no seu navegador. Estudar o código fonte HTML de várias páginas Web é um bom método para aprender mais sobre como as tags trabalham.

Uma vez que os elementos estruturais do seu arquivo HTML estiverem no lugar, você pode adicionar o conteúdo desejado para o corpo do documento HTML. Apesar de existirem muitas tags para alterar a aparência do conteúdo, a discussão a seguir enfoca os elementos básicos: cabeçalhos, parágrafos e listas.

Cabeçalhos

A linguagem HTML tem seis níveis de cabeçalhos, numerados 1 a 6; H1 é o mais proeminente. As letras dos cabeçalhos são exibidas com fontes maiores e/ou em negrito. As letras do corpo da página são exibidas com fonte normal. A sintaxe da tag de cabeçalho é:

```
<Hx>Texto de cabeçalho </Hx>
```

onde x é um número entre 1 e 6, especificando o nível do cabeçalho.

Parágrafos

Os controles de fim de linha CR não são significantes em um arquivo HTML. A mudança de linha é determinada pelo tamanho da janela do navegador. Considere o código HTML a seguir:

```
Bem-vindo à minha home page.
Aqui está algum conteúdo. <P>E um pouco mais de conteúdo.
Este texto apareceria no espectador de Web como segue:
Bem-vindo para minha home page. Aqui está algum conteúdo.
E um pouco mais de conteúdo.
```

No código fonte, existe um retorno de linha entre as sentenças. Os navegadores ignoram este controle, começando um novo parágrafo só quando alcançam uma tag <P>. Esta tag explicitamente termina o parágrafo e insere uma linha em branco.

Você também pode controlar o comprimento de linha ou a quebra de linha usando a tag
, que indica uma quebra de linha.

Nota: nem a tag <P> ou a tag
 exigem uma tag de fim para exibir corretamente uma página nos navegadores atuais.

Listas

A linguagem HTML suporta marcadores e listas numeradas, referindo-se a eles respectivamente como listas não ordenadas e listas ordenadas.

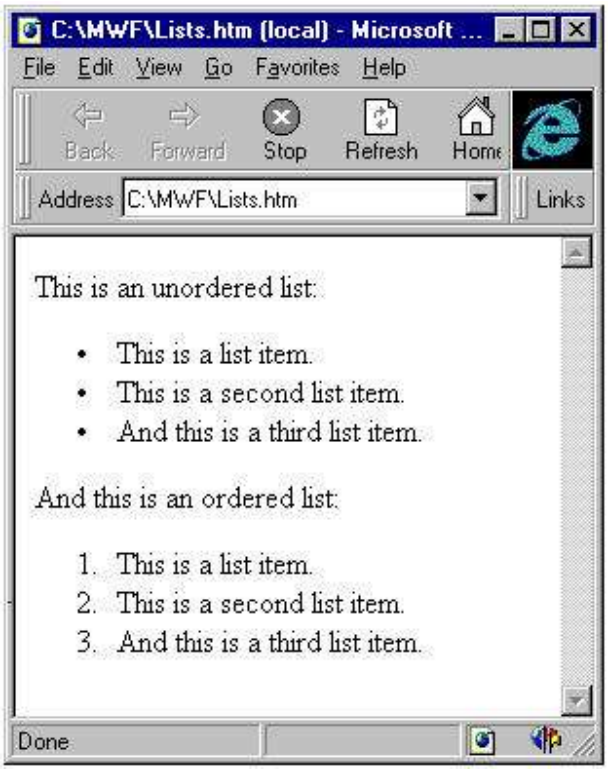
Você deve utilizar a tag <UL> para definir uma lista com marcadores como é exemplificado a seguir:

```
<UL>
<LI>Este é um item da lista.
<LI>Este é um segundo item da lista.
<LI>E isto é um terceiro item da lista.
</UL>
```

Você usa a tag <OL> para definir uma lista numerada como exemplificado a seguir:

```
<OL>
<LI>Este é um item da lista.
<LI>Este é um segundo item da lista.
<LI>E este é um terceiro item da lista.
</OL>
```

A ilustração a seguir mostra como estas listas apareceriam em uma página Web.



Você pode usar o atributo TYPE para controlar o marcador usado em uma lista não ordenada ou o sistema de numeração utilizando uma lista numerada. A sintaxe para uma lista não ordenada é:

`<UL TYPE=formato>`onde a formato é um dos seguintes:

?? `disc`

?? `square`

?? `circle`

A sintaxe para o atributo de TYPE em uma lista ordenada é:

```
<OL TYPE=x>
```

onde x é um dos seguintes:

TIPO	Descrição	Exemplo
1	Números arábicos (padrão)	1, 2, 3, 4, ...
A	Alfanumérico,minúsculas	a, b, c, d, ...
A	Alfanumérico, maiúsculas	A, B, C, D, ...
I	Números romanos, minúsculas	i, ii, iii, iV, ...
I	Números romanos, maiúsculas	I, II, III, IV, ...

Nem todos os navegadores suportam o atributo TYPE.

Hyperlinks

Para criar um hyperlink é utilizada a tag <A>, que funciona como uma âncora. A sintaxe básica para um hyperlink é:

```
<A HREF="Endereço_Destino">texto ou gráfico do hyperlink</A>
```

onde Endereço_Destino é uma URL para o hyperlink e texto ou gráfico do hyperlink é o que o usuário vê e clica para ir para o destino apontado pelo hyperlink.

No exemplo a seguir, o texto "FIA/USP" é um hyperlink para o documento, fia.htm. O documento fia.htm está no mesmo diretório que o documento HTML atual.

```
<A HREF="fia.htm">FIA/USP</A>
```

Tipos de Hyperlinks

A tabela a seguir lista os tipos de hyperlinks que você pode criar e um exemplo da sintaxe HTML de cada um.

Tipo de Hyperlink	Descrição	Sintaxe
URL	Um hyperlink para outra página de Web.	http://xxx/xxx/xxx
Documento não HTML	Um hyperlink para outro tipo de documento, como um documento do Microsoft Word, uma planilha do Microsoft Excel, ou uma imagem.	Arquivo:\\caminho\dir\arquivo
Marcador de páginas	Um hyperlink para uma área nomeada definida dentro de uma página Web.	pagename#bookmarkname
E-mail	Um hyperlink que cria uma nova mensagem endereçada para um destinatário especificado, que é enviada usando o software cliente de correio instalado no equipamento do usuário.	mailto:someone@mycompany.com
Grupo de notícias	Um hyperlink para um grupo de notícias em um site Usenet.	news:xxx.xxx.xxx

Vínculos Relativos e Vínculos Absolutos

Você pode definir hyperlinks em sua página Web com qualquer um de dois tipos de vínculos:

?? **Vínculos Relativos**

Os vínculos relativos descrevem o destino relativamente à localização do documento a partir do qual o usuário está. Você define o caminho do hyperlink relativo removendo a raiz do caminho completo.

Por exemplo, um hyperlink relativo para um documento no mesmo diretório que a página Web atual seria "meu_docto.htm".

```
<A HREF="meu_docto.htm">Texto do Link</A>
```

Você pode especificar também um arquivo contido em outra pasta no mesmo servidor especificando o hyperlink relativo do documento atual até o documento destino como exemplificado a seguir:

```
<A HREF="../minha_pasta/meu_arquivo.htm">Texto do Link</A>
```

Você pode especificar um arquivo em outra pasta no mesmo servidor fornecendo o caminho do diretório raiz da página atual para o documento destino.

```
<A HREF="/diretório/arquivo.htm">Texto do Link</A>
```

?? **Vínculos Absolutos**

Um vínculo absoluto fornece um endereço completo para o hyperlink. Por exemplo, um vínculo absoluto para um documento é:

```
"C:\Meus Documentos Web\meu_docto.htm"
```

O trecho de código a seguir fornece um vínculo absoluto para uma página em outro servidor Web.

```
<A HREF="http://servidor/diretório/arquivo.htm">Texto do Link</A>
```

A vantagem de hyperlinks relativos é que se você mover um grupo de arquivos para outra localização, os endereços de hyperlinks relativos permanecem inalterados.

Porém, você deve usar um hyperlink absoluto para arquivos que:

- Não estão no mesmo computador servidor ou na mesma rede local.
- Não estão no mesmo servidor Web.

Tag Bookmark (Marcador de Páginas)

Um marcador de páginas (*bookmark*) é uma localização em uma página Web que recebe um nome. Um bookmark pode ser um destino de um hyperlink. Quando uma página Web se estender por mais de uma ou duas telas no comprimento, você pode querer inserir bookmarks dentro da página e criar hyperlinks para estes marcadores de páginas, de forma que o usuário possa ir diretamente para a seção de interesse. Este processo é freqüentemente usado para criar um índice simples para uma página Web muito longa, facilitando a navegação do usuário.

Definir um bookmark

?? Use a tag <A> e defina o atributo NAME.

O exemplo a seguir define um marcador de páginas denominado "bookmark1":

```
<A Name="bookmark1">
```

Definir um hyperlink que vai para um bookmark

?? Use a tag <A> e especifique a página destino, seguida por um sinal de libra (#) e o nome do bookmark.

O exemplo a seguir cria um hyperlink para o bookmark "bookmark1" na página "meu_docto.htm".

```
<A HREF="meu_docto.htm#bookmark1">Texto do Link</A>
```

Imagens

Imagens podem ser usadas como ferramentas de navegação. Você pode criar uma imagem com áreas de interesse que são hyperlinks que o usuário pode clicar para descobrir mais sobre um assunto. Estas áreas de interesse são hyperlinks conhecidos como hotspots.

Mapas de imagem

Um mapa de imagem é uma imagem composta de vários hotspots. Os hotspots e hyperlinks que você define para um mapa de imagem são coletivamente chamados de suas informações de mapa ou MAP.

Existem dois caminhos para prover informações MAP:

?? Crie um .arquivo de mapa e o coloque no servidor Web.

Este método produz um mapa de imagem no servidor, porque as informações sobre o tamanho e localização de cada hotspot, e para onde cada hotspot gera um link, residem no servidor.

?? Coloque as informações MAP na página Web propriamente:

Este método produz um mapa de imagem no cliente, porque as informações MAP existem no cliente.

O tópico a seguir aborda as tags HTML usadas para definir mapas de imagem.

Tags de Mapa de Imagem no Servidor

?? Para criar um mapa de imagem no servidor você deve executar estes passos:

1. Construa um hyperlink para o arquivo de mapa de imagem no servidor.
2. Insira o atributo ISMAP para a tag HTML de imagem (<IMG>).

Esta tag notifica o navegador que a imagem é um mapa de imagem no servidor, em lugar de uma imagem regular que liga para uma localização. O navegador consulta o servidor para obter informações sobre como o mapa de imagem funciona.

Esta é a sintaxe HTML para o hyperlink e o atributo ISMAP:

```
<A HREF="/diretorio1/diretorio2/nome.map">
<IMG SRC="/diretorio1/diretorio2/nome.gif" ISMAP>
</A>
```

3. Crie o arquivo de mapa e grave-o no servidor, no mesmo diretório da imagem. O arquivo de mapa é um arquivo texto que você pode criar em qualquer editor de textos ASCII, como o Notepad.

Para cada hotspot na imagem, o arquivo de mapa contém uma linha que especifica a forma, a localização do hotspot e o respectivo hyperlink.

A seguir é exemplificado um arquivo de mapa de imagem.

```
default /default.htm
rect (14,127)(116,229) /miscellaneous.htm
rect (16,48)(134,109) /online.htm
rect (132,5)(205,70) /whatsnew.htm
rect (230,5)(314,86) /toolstech.htm
```

Tags de Mapa de Imagem no Cliente

?? Para criar um mapa de imagem no cliente você deve executar estes passos:

1. No arquivo HTML, insira o atributo USEMAP para a tag HTML de imagem (<IMG>).

   Esta tag notifica o navegador que a imagem é um mapa de imagem no cliente. O navegador consulta a página Web atual para obter informações sobre o mapa de imagem. Esta é a sintaxe HTML para o atributo de USEMAP:

   ```
   <IMG SRC="nome.bmp" USEMAP="#salto">
   ```

   Nota: Com um mapa de imagem no cliente, nenhuma tag de hyperlink é necessária.

   Os caracteres #salto representam o nome provido como referência com a tag <MAP>, prefixada com o símbolo #.

2. No arquivo HTML, use as tags <MAP> e <AREA> para fornecer ao navegador as informações de mapa necessárias: formato dos hotspots, localizações e destino dos hyperlinks.

   O trecho de código exemplificado a seguir mostra o uso das tags <MAP> e <AREA>:

   ```
   <MAP NAME="salto">
         <AREA SHAPE="RECT" COORDS="14,127,116,229"
               HREF="miscellaneous.htm">
         <AREA SHAPE="RECT" COORDS="16,48,134,109"
               HREF="online.htm">
         <AREA SHAPE="RECT" COORDS="132,5,205,70"
               HREF="whatsnew.htm">
         <AREA SHAPE="RECT" COORDS="230,5,314,86"
               HREF="toolstech.htm">
   </MAP>
   ```

Fornecendo Informações para Mapas de Imagem

Uma vez que nem todos os navegadores suportam mapas de imagem no cliente, você deve oferecer as duas opções em sua página Web. Deste modo, se o navegador do usuário não suportar mapas de imagem no cliente, poderá usar a opção de mapa de imagem no servidor.

Para fornecer informações para ambos os mapas de imagem (cliente e servidor), use esta sintaxe:

```
<A HREF="name.map">
<IMG SRC="name.gif" USEMAP="#nome" ISMAP>
</A>
<MAP NAME="nome">

</MAP>
```

À medida que você incorporar imagens em suas páginas Web, mantenha o seguinte em mente:

?? Imagens de não estão automaticamente exibidas em todos os navegadores Web. Alguns navegadores não podem exibir gráficos, enquanto outros fornecem ao usuário a opção de inibir a exibição de imagens. Portanto, projete a sua página Web de forma que o usuário possa ver as informações com ou sem imagens. Um texto alternativo para as imagens ajuda os usuários que não podem ou optem por não vê-las.

?? Imagens podem diminuir significativamente o desempenho de sua página Web, especialmente para usuários com modems mais lentos. Páginas Web com imagens grandes, coloridas podem parecer excelentes, mas se sua página toma muito tempo para ser carregada, seu usuário Web pode perder paciência e abandonar sua página.

?? Você pode diminuir o tempo de carga de sua página Web reduzindo o número de cores em suas imagens ou definindo atributos de altura e largura específicas.

?? A utilização de *thumbnails* aumenta o desempenho das páginas Web. *Thumbnails* são imagens pequenas que você mapeia com hyperlinks para as páginas que contém as versões completas das imagens ou gráficos. Com essa estratégia os usuários têm a opção de ver imagens, mas sua página principal será carregada muito mais rapidamente.

Existem vantagens e desvantagens associadas com mapas de imagem. Ao decidir por usar ou não mapas de imagem, considere as vantagens e desvantagens relacionadas a seguir para cada caso.

Vantagens de Mapas de Imagem no Servidor

?? Cada gráfico em uma página é carregado do servidor como um fluxo separado para o cliente. Por esse motivo, diversas imagens pequenas levarão mais tempo para carregar que uma imagem grande.

?? Com mapas de imagem no servidor, você pode controlar a localização dos hyperlinks sem mudar a página HTML. Se um hyperlink mudar, você só precisa mudar o arquivo de mapa de imagem no servidor. Além disso, caso você inclua o mesmo mapa de imagem em diversas páginas, poderá atualizar todos os hyperlinks em todas as suas páginas mudando apenas o arquivo de mapa no servidor.

Desvantagens de Mapas de Imagem

?? Algum navegadores não suportam mapas de imagem no cliente.

?? Mapas de imagem no servidor exigem que o clique do mouse seja enviado para o servidor, para ser resolvido, portanto o tempo de resposta é mais demorado e o tráfego na rede aumenta.

?? Quando você usar imagens individuais em lugar de mapas de imagem, você tem controle completo acima do plano de sua página Web. Nas páginas diferentes você pode usar os mesmos elementos de gráficos com o mesmo hyperlink em configurações e planos diferentes.

?? Com imagens individuais, quando o usuário desligar gráficos (ou estiver usando um navegador que só aceita textos), o texto alternativo para cada gráfico pode ainda ser visto. Um mapa de imagem mostra ao usuário só um texto alternativo, para o mapa de imagem inteiro.

Tabelas

Tabelas servem para muitos propósitos nas páginas HTML. Não só são usadas da maneira convencional para exibir informações em um formato tabular, mas também são usadas para fazer com que imagens e texto apareçam exatamente na localização desejada na página; ou seja, são essenciais para a formatação e posicionamento de elementos gráficos e textuais numa página Web.

Controlando atributos como a largura e altura das células das tabelas, ou então de suas bordas e o espaço adicional ao redor conteúdo ou entre células, você pode controlar o posicionamento absoluto de qualquer objeto em sua página.

Uma tabela é criada através das tags <TABLE> e </TABLE> em um documento HTML. Quaisquer atributos que se aplicam para uma tabela como um todo são definidos na tag <TABLE>:

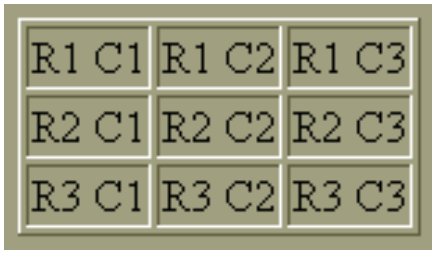
```
<TABLE ALIGN=LEFT BORDER=1 WIDTH=20%>

</TABLE>
```

Cada linha na tabela é criada usando as tags <TR> e </TR>. Como cada linha, colunas são criadas com as tags <TD> e </TD>. Qualquer elemento HTML pode ser colocado dentro das tags <TD> e </TD> (por exemplo, textos, imagens, mapas de imagem, hyperlinks ou formulários). O código exemplificado a seguir cria uma tabela de três linhas por três colunas em uma página:

```
<TABLE ALIGN=LEFT BORDER=1 WIDTH=20%>
<TR><TD>R1 C1</TD><TD>R1 C2</TD><TD>R1 C3</TD></TR>
<TR><TD>R2 C1</TD><TD>R2 C2</TD><TD>R2 C3</TD></TR>
<TR><TD>R3 C1</TD><TD>R3 C2</TD><TD>R3 C3</TD></TR>
</TABLE>
```

A ilustração a seguir mostra a tabela resultante:



Largura e Altura da Tabela

Você pode fixar a largura e a altura das células em uma tabela colocando os atributos WIDTH e HEIGHT dentro da tag <TD>.

Se não for incluído o atributo WIDTH (largura) ou HEIGHT (altura) na tag <TD>, a tabela se expandirá até o tamanho da janela do navegador. A largura (ou altura) de cada coluna (ou linha) será baseada no maior elemento naquela linha ou coluna.

Uma alternativa para isto, é fixar os atributos WIDTH e HEIGHT para um valor porcentual, em lugar de um tamanho fixo. Por exemplo, se você fixar o atributo WIDTH da tag <TABLE> para 50%, a tabela sempre ocupará 50% da largura disponível, determinada pelo tamanho da janela do navegador e se expandirá automaticamente com o navegador. A tabela do exemplo anterior sempre se expandirá ou se reduzirá para ocupar 20% da largura da janela do navegador.

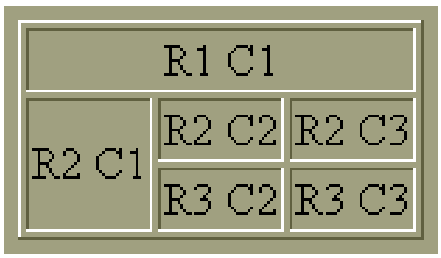
Colunas e Linhas

Você pode usar os atributos de COLSPAN e ROWSPAN das tags <TD> e <TH para estender o conteúdo de uma célula em células adjacentes. Isto é útil se você precisar criar um cabeçalho através de mais de uma coluna.

Os trecho de código do exemplo a seguir expande e centraliza o conteúdo da primeira coluna na primeira linha da tabela. O conteúdo da primeira coluna na segunda linha são expandidos até incluir a terceira linha da tabela:

```
<TABLE ALIGN=LEFT BORDER=1 WIDTH=20%>
<TR><TD COLSPAN=3 ALIGN=CENTER>R1 C1</TD></TR>
<TR><TD ROWSPAN=2>R2 C1</TD><TD>R2 C2</TD><TD>R2 C3</TD></TR>
<TR><TD>R3 C2</TD><TD>R3 C3</TD></TR>
</TABLE>
```

A ilustração a seguir mostra a tabela resultante:



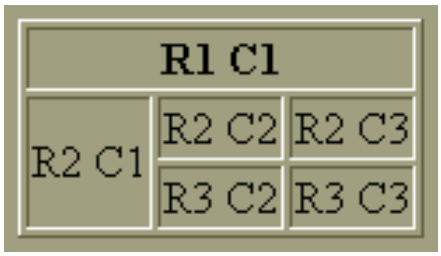
Cabeçalhos de Tabelas

Você pode especificar um cabeçalho (exibido em com texto em negrito) para sua tabela. Qualquer célula que usa as tags <TH> e </TH> em lugar das tags <TD> e </TD> exibirá texto em negrito na tabela.

O trecho de código do exemplo a seguir converte o texto na linha superior para um cabeçalho.

```
<TABLE ALIGN=LEFT BORDER=1 WIDTH=20%>
<TR><TH COLSPAN=3 ALIGN=CENTER>R1 C1</TH></TR>
<TR><TD ROWSPAN=2>R2 C1</TD><TD>R2 C2</TD><TD>R2 C3</TD></TR>
<TR><TD>R3 C2</TD><TD>R3 C3</TD></TR>
</TABLE>
```

A ilustração a seguir mostra a tabela resultante:



Bordas de Tabelas

Você pode usar o atributo de BORDER para criar uma borda para sua tabela. O número que você especifica para o atributo BORDER define a largura da borda. Por exemplo:

```
<TABLE BORDER=4>
```

Você pode usar os atributos de FRAME e RULES da tag <TABLE> para controlar como a borda da tabela será apresentada. O atributo de FRAME especifica a borda para o lados externos da tabela (bordas exteriores). O atributo de RULES especifica as linhas divisórias (bordas internas) da tabela.

O trecho de código exemplificado a seguir exibe uma borda em torno da tabela e bordas horizontais separando as linhas:

```
<TABLE BORDER=2 FRAME=box RULES=rows>
```

A ilustração a seguir mostra a tabela resultante:

Título de Tabelas

Você pode inserir um título para sua tabela usando a tag <CAPTION>. Use o atributo ALIGN para definir o alinhamento horizontal do título. Use o atributo VALIGN para definir a posição vertical do título (TOP ou BOTTOM).

O trecho de código exemplificado a seguir adiciona um título centralizado na parte inferior da tabela:

```
<TABLE ALIGN=LEFT BORDER=1 WIDTH=40%>
<CAPTION ALIGN=CENTER VALIGN=BOTTOM> Minha Tabela </CAPTION>
<TR><TH COLSPAN=3>R1 C1</TH></TR>
<TR><TD ROWSPAN=2>R2 C1</TD><TD>R2 C2</TD><TD>R2 C3</TD></TR>
<TR><TD>R3 C2</TD><TD>R3 C3</TD></TR>
</TABLE>
```

Colunas de Agrupamento

A tag <COLGROUP> fornece um meio para especificar atributos para uma coluna inteira em uma tabela. A ordem das tags <COLGROUP> determina a que coluna uma tag se refere.

As tags no exemplo de código a seguir alinham à esquerda as primeiras duas colunas, e centralizam a terceira coluna.

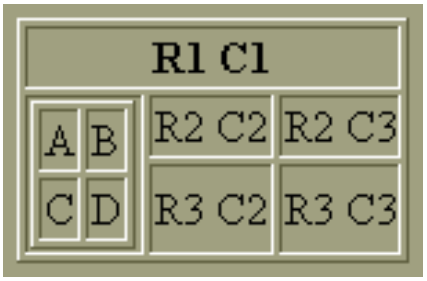
```
<TABLE>
<COLGROUP SPAN=2 ALIGN=LEFT>
<COLGROUP ALIGN=CENTER>
<TBODY>
    <TR>
    <TD>Esta coluna está no primeiro grupo e é alinhado à esquerda.</TD>
    <TD>Esta coluna está no primeiro grupo e é alinhada à esquerda.</TD>
    <TD>Esta coluna está no segundo grupo e é centralizada.</TD>
    </TR>
</TABLE>
```

Tabelas Aninhadas

As tabelas podem ser aninhadas, ou seja, tabelas contendo outras tabelas dentro de suas células, de forma a permitir a criação de tabelas mais complexas. Para aninhar uma tabela dentro de outra, insira outro par de tags <TABLE> e </TABLE> em lugar de texto ou outro conteúdo numa célula. Por exemplo:

```
<TABLE ALIGN=LEFT BORDER=1 WIDTH=50%>
<TR><TH COLSPAN=3 ALIGN=CENTER>R1 C1</TH></TR>
<TR><TD ROWSPAN=2>
      <TABLE BORDER 1>
      <TR><TD>A</TD><TD>B</TD></TR>
      <TR><TD>C</TD><TD>D</TD></TR>
      </TABLE>
</TD><TD>R2 C2</TD><TD>R2 C3</TD></TR>
<TR><TD>R3 C2</TD><TD>R3 C3</TD></TR>
</TABLE>
```

A ilustração a seguir mostra a tabela resultante:



Páginas com Frames

Uma página com *frames* divide a janela de um navegador em seções conhecidas como *frames* ou quadros. Cada frame exibe uma página Web separada.

Os frames permitem criar efeitos visuais interessantes em uma página Web. Por exemplo:

?? Você pode colocar elementos fixos, ou seja, que o usuário sempre veja, como uma barra de título e alguns hyperlinks, em uma frame estática. Estes elementos permanecerão visíveis mesmo quando o usuário rolar a janela do navegador para ver o restante das informações conditas na página.

?? Uma vez que cada frame é definido por um arquivo HTML independente, os frames podem, por exemplo, exibir uma questão em um lado da página e o resultado no outro, permitindo o usuário mover-se entre eles.

As páginas Web que usam frames possuem dois elementos principais:

1. O arquivo HTML principal:

   Este arquivo não tem nenhuma tag <BODY>, mas contém as tags necessárias para implementar cada frame da página.

2. Os arquivos de HTML a serem exibidos em cada frame:

   Cada frame na página tem seu próprio arquivo HTML fonte.

Aqui sugerimos algumas diretrizes gerais para usar frames adequadamente:

?? Utilize frames para exibir informações que devam permanecer estáticas em uma área da página, enquanto nas outras áreas as informações mudam. Os frames são bem aplicados quando um frame contiver itens a serem escolhidos (menu com hyperlinks por exemplo), e os outras frames simultaneamente apresentem o resultado da escolha.

?? Utilize frames sem bordas sempre que possível.

?? Forneça bastante espaço em branco nas páginas que exibem os frames. Uma vez que os frames podem exibir muitas informações simultaneamente, procure inserir espaços entre parágrafos, imagens e outros elementos, de forma que os frames não parecem abarrotados de informações, pois isto faz com que as informações nos frames fiquem mais legíveis.

Os frames são uma característica importante de projeto de uma página Web. Porém, eles têm algumas desvantagens:

?? Frames dividem a janela do navegador em seções pequenas, que podem tornar mais difícil a visualização das informações contidas nas páginas.

?? Frames podem confundir os usuários se não forem adequadamente projetadas para o tipo de conteúdo que exibem.

?? Algum navegadores não suportam frames sem bordas, e desenharão uma borda pouco atrativa ao redor de cada frame, o que prejudica a aparência da página Web.

A tag <FRAMESET>

As páginas com frames usam as tags <FRAMESET> e </FRAMESET> em vez das tags para definir o corpo do documento, <BODY> e </BODY>. A razão para isto é que uma página com frames é um recipiente em que outras páginas Web serão carregadas, do mesmo modo que as tags <BODY> e </BODY> formam um recipiente para as páginas Web que não usam bordas.

As tags <FRAMESET> e </FRAMESET> definem a localização e tamanho dos frames em uma página HTML. A tag <FRAMESET> tem dois atributos: ROWS e COLS.

O atributo ROWS define frames horizontais. É seguido por uma lista delimitada por vírgulas com os tamanhos de cada borda na página. Você pode especificar tamanhos em pixel, porcentagem ou relativos. Os tamanhos em pixel são úteis para bordas estáticas, mas não se ajustam em resposta a variações no tamanho da janela do navegador. Os outros métodos são mais flexíveis.

No trecho de código exemplificado a seguir, o primeiro frame possui 120 pixeis, o terceiro frame 20% da altura total da janela, e o segundo frame recebe o remanescente da altura:

```
<FRAMESET ROWS="120, *, 20%">
```

O trecho de código do exemplo a seguir cria dois frames que dividem o tamanho completo da janela do navegador. O frame superior é sempre duas vezes maior que o frame da parte inferior.

```
<FRAMESET ROWS="2*, *">
```

O atributo de COLS define uma página com frames verticais. Este atributo é especificado do mesmo modo que atributo ROWS.

O trecho de código a seguir exemplifica o uso das tags para a criação de frames numa página Web:

```
<html>
<!O código abaixo cria um conjunto de frames com 2 colunas, uma com
largura de 150 pixeis, e a outra com largura variável, dependendo da área
disponível na tela do navegador>
<frameset cols="150,*">
<!-O código abaixo descreve o comportamento da frame da esquerda->
  <frame name="left" scrolling="no"
src="http://blackbox/mainstre/MSMBODY.HTM">
```

```
<! O código abaixo cria um conjunto de frames com duas linhas: um
ocupando 20% da tela e a outra ocupando o espaço restante (neste caso
80%)>
  <frameset rows="20%,*">
<! O código abaixo define os frames e os respectivos arquivos HTML que
serão apresentados em cada frame horizontal>
    <frame name="rtop" target="rbottom" src="msmjumps.htm">
    <frame name="rbottom" src="http://blackbox/mainstre/CUSTDATA.HTM">
  </frameset>
  <noframes>
  <body bgcolor="#FFFFFF">
  <p>Esta página utiliza frames, mas o seu navegador não os suporta.</p>
  </body>
  </noframes>
</frameset>
</html>
```

A tag <FRAME>

As tags <FRAME> e </FRAME> definem um frame único em uma página com frames. Estas tags definem o arquivo de HTML para o frame, como também a aparência do frame propriamente dito. A tabela seguinte descreve os atributos da tag <FRAME>.

Atributo	Descrição	Exemplo
SRC	Exibe o arquivo para a frame. Se este parâmetro for omitido será criada um frame em branco.	<FRAME SRC="frame1.htm">
NAME	Fornece um nome de destino para o frame.	
<FRAME NAME="topo">		
MARGINWIDTH	Controla a largura da margem para o frame em pixeis.	<FRAME MARGINWIDTH="20">
MARGINHEIGHT	Controla a altura de margem para o frame em pixeis.	<FRAME MARGINHEIGHT="10">
BORDER	Fornece a opção de exibir ou não uma borda para um frame.	<FRAME BORDER="Yes">
SCROLLING	Cria um frame com capacidade para rolar seu conteúdo. Os valores são Yes, No, ou Auto.	<FRAME SCROLLING="Yes">
NORESIZE	Não permite que o usuário redimensione o tamanho do frame.	<FRAME NORESIZE>

Roteiro para criar um Documento de HTML com frames

1. Crie um documento HTML fonte para cada frame em sua página Web. Os documentos fonte podem conter quaisquer tags HTML.
2. Crie um novo documento HTML que contenha as tags <HTML> </HTML> e <HEAD> </HEAD>, mas nenhuma tag <BODY> </BODY>. Isto é, o documento do frame principal que será diretamente aberto pelo navegador.
3. Na área do documento que contém normalmente as tags <BODY>, inclua a tag <FRAMESET>. Defina os atributos para o tamanho e orientação dos frames. O trecho de código exemplificado a seguir define uma página com dois frames horizontais:

```
<HTML>
<HEAD>
<TITLE>O Documento de Frame Principal</TITLE>
</HEAD>

<FRAMESET ROWS="100, *">

</FRAMESET>
</HTML>
```

4. Para cada frame na página, insira uma tag <FRAME>. Configure o atributo SRC da tag <FRAME> para o nome do documento HTML que aparecerá no frame. Defina outros atributos para o frame, como bordas, rolamento, espaçamento e redimensionamento. Por exemplo:

```
<FRAMESET ROWS="100, *">
   <FRAME SRC="FrameDoc1.htm" SCROLLING="No" noresize>
   <FRAME SRC="FrameDoc2.htm" SCROLLING="Sim" noresize>
</FRAMESET>
```

Nota: Neste exemplo, os arquivos FrameDoc1.htm e FrameDoc2.htm precisam estar na mesma pasta do arquivo principal.

Para Navegadores Que Não Suportam Frames

Nem todos os navegadores Web suportam frames. Em consideração aos usuários destes navegadores, você deve fornecer um código HTML alternativo, colocando-o entre as tags <NOFRAMES> e </NOFRAMES> no arquivo HTML do frame principal. Estas tags podem aparecer depois das tags <FRAMESET>.

Formulários

Os formulários fornecem um mecanismo através do qual usuários do seu Web Site podem se comunicar com o servidor Web. Os usuários digitam dados em um formulário e clicam um botão para envia-los ao servidor Web.

Um formulário contém tipicamente:

?? Vários campos de formulário.

?? Botões Submeter (*Submit*) e Limpar (*Reset*).

Normalmente, os usuários preenchem os campos do formulário e clicam o botão Submeter para enviar os dados para o servidor Web. O botão de Limpar (*Reset*) apaga os dados dos campos do formulário.

A ilustração a seguir mostra uma formulário com alguns campos.

Submetendo um Formulário

Quando o usuário clica o botão Submeter ou aperta a tecla Enter, os dados dos campos do formulário são enviados para um programa manipulador de formulários no servidor Web. Este programa sempre é executado quando um usuário submete os dados de um formulário para o servidor. O manipulador de formulários processa os dados submetidos e tipicamente devolve uma página HTML para o usuário com uma confirmação. Você pode configurar as propriedades de um formulário para especificar que programa manipulador de formulários será executado quando o usuário submeter dados através de um formulário para o servidor Web.

O Microsoft FrontPage fornece alguns programas padrão para manipulação de formulários que você pode usar - *FrontPage Server Extensions*. Você também pode construir um programa manipulador personalizado, criando uma página ativa (.ASP) ou uma DLL, que são executas no servidor Web.

Formulários do Microsoft FrontPage

Os programas manipuladores de formulários fornecidos pelo Microsoft FrontPage são conhecidos como Extensões do Servidor e devem ser instalados em seu servidor Web. A grande maioria dos provedores de acesso à Internet que hospedam páginas HTML disponibilizam as Extensões do Servidor do FrontPage para seus clientes.

Você pode criar os seguintes tipos de formulários com o FrontPage:

?? **Formulário para gravação de dados**:

Um formulário para gravação de dados permite um salvar um conjunto de informações digitadas por um usuário e armazena-las em um arquivo no servidor Web ou envia-las para um endereço de e-mail. Você pode configurar as propriedades do formulário para especificar onde as informações serão gravadas ou para onde serão enviadas.

Este é o manipulador de formulário padrão, utilizando pelo FrontPage quando você cria um novo formulário.

?? **Formulário para pesquisa**:

Um formulário para pesquisa aceita critérios de busca de um usuário e procura no Web Site as páginas contenham informações que coincidem com os critérios de busca especificados, retornando uma página HTML que contém hyperlinks para as páginas encontradas. Você pode usar FrontPage para criar um formulário de procura para seu Web Site.

?? **Formulário de discussão**:

Você pode criar um formulário de discussão para usar em um grupo de discussão. Um grupo de discussão é um Web Site que permite a manutenção de uma discussão interativa entre os usuários sobre vários assuntos. Os usuários submetem assuntos digitando texto no formulário de discussão.

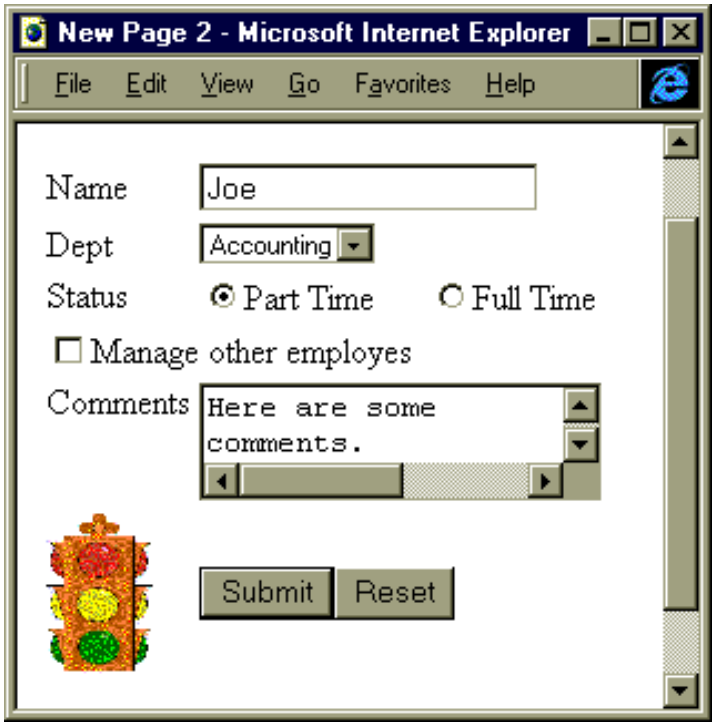
?? **Formulário de inscrição**:

Você pode criar um formulário de inscrição que adiciona pessoas e senhas para formar uma lista de usuários que têm permissão para acessar o seu Web Site. Você pode criar um formulário de inscrição de usuários criando uma nova página Web com o modelo de Inscrição de Usuários do Front Page. Em seguida, você deve configurar permissões em seu Web Site para que só usuários registrados tenham acesso às suas páginas. Uma vez que você definiu estas permissões, quando os usuários tentarem acessar uma página do seu Web Site, terão que preencher o formulário de inscrição de usuários.

Controles de Formulário

Controles HTML padrão, também conhecidos como controles intrínsecos, são controles que você pode inserir em um formulário para exibir ou recuperar informações. Todos os navegadores suportam controles HTML padrão, que normalmente são chamados de campos de formulário. Quaisquer informações digitadas em um controle padrão é enviada para o servidor Web quando o usuário submeter os dados do formulário.

A ilustração a seguir mostra um formulário com vários campos.



Tags para Criação de Formulários

Para criar um formulário você usa as tags <FORM> e </FORM>. Você pode colocar quaisquer elementos HTML entre as tags <FORM> e </FORM>, exceto outros formulários.

O trecho de código HTML a seguir define um formulário que contém vários campos, um botão de submeter e um botão de limpar. Quando o usuário submeter o formulário, a página ativa do servidor (página .ASP) denominada meuexemplo.asp será executada e receberá os dados dos campos do formulário.

```
<FORM ACTION="/scripts/meuexemplo.asp" METHOD=POST>
      Nome de E-mail: <INPUT TYPE=TEXT NAME="txt" VALUE="Meu Nome"><P>
      Confira tudo que se aplica:
      <INPUT TYPE=CHECKBOX NAME="chkUsoProfissional">Uso Profissional
      <INPUT TYPE=CHECKBOX NAME="chkUsoDomestico">Uso Doméstico<P>
      <INPUT TYPE=SUBMIT VALUE="Submeter">
      <INPUT TYPE=RESET VALUE="Limpar">
</FORM>
```

Você determina o comportamento de um formulário através da definição dos atributos ACTION, METHOD e TARGET.

ACTION

O atributo ACTION especifica o manipulador de formulário que deverá ser utilizado pelo servidor para manipular os dados submetidos pelo formulário. Você configura o atributo ACTION para uma URL que especifica o nome do programa ou página ativa (.ASP) que receberá os dados do formulário quando o usuário submete-los.

METHOD

O atributo METHOD indica o tipo de protocolo de manipulação de formulário que será usado para processar os dados submetidos ao servidor. Existem dois valores possíveis para o atributo METHOD:

?? `GET`

Para enviar poucos dados para um servidor Web, configure o atributo METHOD para GET. Quando você usar método GET, os valores dos controles são concatenados para a URL que é indica no manipulador de formulário, e então são enviados para o servidor. Usando este método, você poderá enviar no máximo 1024 bytes de dados.

Tipicamente, você utiliza o método GET para tarefas como enviar uma consulta SQL para um servidor Web a fim de pesquisar informações numa base de dados.

?? `POST`

Se você precisar enviar mais de 1024 bytes de dados para um servidor Web, configure o atributo METHOD para POST. Quando você usar o método POST, os dados do formulário são enviados no corpo do pedido do protocolo HTTP. Não existe nenhum limite para o número de parâmetros ou para o comprimento dos valores que você pode enviar ao usar este método.

Tipicamente você usa o método POST para tarefas como enviar dados para modificar registros numa base de dados.

TARGET

Você pode configurar o atributo TARGET para especificar onde o resultado da submissão do formulário será exibido. O resultado da submissão do formulário retorna ao cliente um novo arquivo HTML que substitui a página que o usuário está vendo atualmente.

Se você estiver usando frames na página do formulário, configure o atributo TARGET para especificar que frame deve apresentar os resultados.

```
<FORM TARGET="Frame1" ...>
```

Controles de Formulário

A maioria dos controles HTML é definida com a tag <INPUT>. A sintaxe para a tag <INPUT> é a seguinte:

```
<INPUT TYPE=tipo_controle NAME=nome_controle VALUE=valor_controle>
```

?? O atributo TYPE especifica o tipo do controle.

?? Com o atributo NAME, você pode designar um nome para o controle que o identificará exclusivamente dentro de uma página HTML.

?? O atributo VALUE é o valor designado para o controle quando o usuário interagir com ele.

Controle de Caixa de texto

Existem três tipos de caixas de texto que pode ser criadas com tags HTML diferentes, como mostra a tabela s seguir:

Tipo de caixa de texto	Tag HTML
Linha única	<INPUT TYPE=TEXT>
Senha	<INPUT TYPE=PASSWORD>
Texto com rolamento	<TEXTAREA> </TEXTAREA>

O exemplo a seguir define um controle de caixa de texto de uma linha:

```
<INPUT TYPE=TEXT NAME=txtUserName VALUE="UserName@Provedor.com.br">
```

O exemplo a seguir define um controle de caixa de texto com rolamento:

```
<TEXTAREA ROWS="2" NAME="txtComments">
      Algum texto na caixa
</TEXTAREA>
```

Botão

Existem três tipos de botões: Normais, Limpar e Submeter. Para inserir estes controles, você usa a tag <INPUT> e configura o atributo TYPE. O atributo VALUE é o texto exibido na face do botão.

A tabela a seguir mostra a tag HTML usada para cada tipo de botão.

Tipo de botão	Tag HTML
Normal	<INPUT TYPE=BUTTON NAME=nome_botão VALUE="valor">
Limpar	<INPUT TYPE=RESET VALUE="valor">
Submeter	<INPUT TYPE=SUBMIT NAME="nome_botão" VALUE="valor">

O exemplo a seguir define um botão Normal denominado btnClickMe:

```
<INPUT TYPE=BUTTON NAME=btnClickMe VALUE="Me clique">
```

O exemplo a seguir define um botão Limpar:

```
<INPUT TYPE=RESET VALUE="Limpar">
```

Este exemplo define um botão Submeter:

```
<INPUT TYPE=SUBMIT NAME="btnSubmit" VALUE="Submeter">
```

Menu em Cascata

Para adicionar um menu em cascata (também chamado de um controle de seleção) em um formulário, use as tags <SELECT> e </SELECT>. Para adicionar itens para as opções do menu, use a tag <OPTION>. Para fazer um dos itens aparecer quando um usuário acessar o menu na página Web, adicione o atributo SELECTED para a tag <OPTION> da opção padrão.

O próximo exemplo exibe um controle de menu em cascata que contém uma lista de preferências. A opção Correio estará pré selecionada.

```
<SELECT NAME=lstPreferencia>
<OPTION SELECTED VALUE="1">Correio
<OPTION VALUE="2">Fax
<OPTION VALUE="3">On-line
</SELECT>
```

Para permitir que o usuário possa selecionar mais de uma opção no menu em cascata, adicione o atributo MULTIPLE para a tag <SELECT> como exemplificado a seguir:

```
<SELECT MULTIPLE=lstPreferencia>
```

Botão de Rádio

Para adicionar um controle de botão de rádio em um formulário, utilize a tag <INPUT> e configure o atributo TYPE para RADIO. Use uma tag <INPUT> separada para cada botão de rádio a ser inserido no formulário.

Para criar um grupo de botões de rádio, designe o mesmo atributo NAME para todos os botões de rádio do grupo.

Para especificar qual opção será pré selecionada quando um usuário acessar a página Web, adicione o atributo CHECKED para a tag <INPUT>.

O exemplo a seguir exibe um grupo de botões de rádio que fornece três opções: Sim, Não, e Talvez. A opção Sim opção é pré selecionada.

```
<INPUT TYPE=RADIO CHECKED NAME=optQuestao VALUE="Sim">Sim
<INPUT TYPE=RADIO NAME=optQuestao VALUE="Não">Não
<INPUT TYPE=RADIO NAME=optQuestao VALUE="Talvez">Talvez
```

Caixa de Verificação

Para adicionar um controle de caixa de verificação em um formulário, use a tag <INPUT> e configure o atributo de TYPE para CHECKBOX.

Para fazer a caixa de verificação aparecer selecionada quando o usuário acessar a página, adicione o atributo CHEKED para a tag <INPUT>.

Para adicionar um texto descritivo para a caixa de verificação, inclua o texto depois da tag <INPUT>.

O exemplo a seguir ilustra a criação de uma caixa de verificação:

```
<INPUT TYPE=CHECKBOX NAME=chkUsoProfissional VALUE="Profissional"
CHEKED>Uso Profissional
<INPUT TYPE=CHECKBOX NAME=chkUsoPessoal VALUE="Pessoal">Uso Pessoal
```

Controle Escondido

Para adicionar um controle escondido em um formulário, use a tag <INPUT> e configure o atributo de TYPE para HIDDEN.

Configure o atributo VALUE para o texto que você deseja que seja enviado com o controle escondido.

O trecho de código a seguir cria um controle escondido denominado "hdnNome". Quando o usuário submeter o formulário, o texto para o atributo VALUE é enviado para o servidor.

```
<INPUT TYPE=HIDDEN NAME=hdnName
VALUE="Informações você não quer que o usuário veja">
```

Para mudar o formato de um formulário, você pode adicionar tags HTML ou mudar os atributos das tags existentes.

Ordem de Tabulação

Para definir a ordem de tabulação dos controles do formulário, configure o atributo TABINDEX de cada controle, como o apresentado no exemplo a seguir. Designe "1" para o controle que estará selecionado o usuário abrir a página; designe "2" para o próximo controle que receberá o foco quando o usuário apertar a tecla de tabulação, e assim por diante.

```
<INPUT TYPE=TEXT NAME=txtName TABINDEX=2>
```

Adicionando um Título

Se você adicionar um título para um controle, o usuário pode selecionar o controle clicando o título ou o controle. Para adicionar um título, primeiro configure o atributo ID do controle para o qual você deseja associar um título. O atributo ID especifica um nome para o controle. Em seguida, adicione a tag <LABEL> e configure o atributo FOR para definir o ID do controle.

O código exemplificado a seguir cria um título para o botão de rádio "optColor1".

```
<INPUT TYPE="RADIO" NAME=optVerde VALUE=1 ID=optColor1>
<LABEL FOR=optColor1> Verde </LABEL>
```

Teclas de Acesso

Para criar uma tecla de acesso para um controle, configure o atributo ACCESSKEY da tag <LABEL> do controle.

O trecho de código a seguir define uma tecla de acesso para a opção Verde. O código sublinha a letra "V" para fornecer uma sugestão visual para o usuário.

```
<INPUT TYPE="RADIO" NAME=optVerde VALUE=1 ID=optColor1>
<LABEL FOR=optColor1 ACCESSKEY="V"><u>V</u>erde </LABEL>
```

Programação Orientada para Objetos com Scripts

Introdução

O desenvolvimento de scripts é baseado no modelo de programação orientada para objetos. A programação orientada para objetos permite que você escreva códigos associados a objetos específicos em seu aplicativo.

Um objeto é uma combinação de código e dados que podem ser tratados como uma unidade. Um objeto pode ser um pedaço de uma página de aplicativo ou Web, como um controle, ou a página inteira propriamente dita. Por exemplo, um botão de comando que você coloca em um formulário em uma página HTML é um objeto. Todos os objetos possuem as seguintes características:

- Propriedades
- Métodos
- Eventos

Propriedades

As propriedades são os atributos de um objeto, como seu tamanho, legenda e cor. Você pode configurar propriedades de um objeto quando adicionar um objeto em uma página Web em tempo de projeto, ou quando escrever scripts para alterar propriedades de um objeto quando um usuário interagir com ele.

Métodos

Os métodos são as ações que um objeto pode executar. Por exemplo, um objeto formulário possui o método submeter. Você pode escrever scripts para invocar os métodos de um objeto. Por exemplo, você pode executar a seguinte linha de código para submeter manualmente o conteúdo de um formulário para o servidor:

```
MeuForm.Submit
```

Eventos

Os eventos são procedimentos que um objeto invoca em resposta a ações de um usuário ou do sistema. Por exemplo, um botão de comando tem um procedimento de evento chamado OnClick que é executado quando um usuário clica o botão.

Você pode adicionar scripts para procedimentos associados a eventos; sempre que o evento ocorrer, o script será executado.

Cada objeto possui um conjunto separado de procedimentos associados a eventos. Se você tiver dois botões de comando em uma página Web, cada botão tem seu próprio evento OnClick. Por exemplo, um botão submeter em um formulário pode executar um código

para submeter os dados do formulário para o servidor, enquanto que um botão Limpar pode apagar todos os campos para que o usuário redigite os dados.

Programação com Scripts

Na Web, a programação no cliente, ou seja, a ser executada pelo navegador, é sempre realizada através de linguagens de script, e dempenha um papel importante na criação de páginas Web com conteúdo ativo. Usando uma linguagem de script, você pode criar páginas Web ativas que:

- Fornecem respostas para perguntas e questões do usuário.
- Validam dados do usuário.
- Calculam expressões.
- Forneçam links para outros aplicativos.
- Manipulam controles ActiveX e Java Applets.

Script no Servidor X Script no Cliente

Você pode escrever scripts que são executados no navegador do cliente ou no servidor Web. Dependendo das necessidades do seu Web Site, você pode usar scripts de cliente, scripts de servidor, ou ambos.

Scripts no Cliente

Para usar scripts no cliente, você embute o código-fonte do script na página HTML como um texto ASCII. Quando a página for carregada do servidor para o cliente, o código do script não é compilado. Conseqüentemente, alguém pode ver ou copiar o seu script, obtendo seu código-fonte HTML, ou seja, o script no cliente torna aberto o código-fonte das suas páginas Web.

Quando um navegador encontra um script, ele chama um interpretador adequado que analisa gramaticalmente e executa o código. Portanto, para que seus usuários possam ter acesso completo às características de uma página Web que contém scripts, eles devem estar utilizando um navegador que os suporta, como o Microsoft Internet Explorer ou o NetScape Navigator, versões 4.0 ou superiores.

Scripts no Servidor

Quando há scripts no servidor, seu código-fonte é executado pelo próprio servidor Web antes da página ser retornada ao usuário. Um script no servidor cria uma página HTML normal de retorno, portanto, o usuário nunca vê o código-fonte do script executado no servidor.

Para usar scripts no servidor, seu servidor Web precisa suportar páginas de servidor ativas (*Active Server Pages* ou ASP), recurso que foi introduzido no servidor Web *Microsoft Internet Information Server* (IIS) a partir da versão 3.0.

?? Para criar uma página Web que contenha script no servidor execute os seguintes passos:

1. Adicione um script no servidor como um texto ASCII para a página.
2. Salve a página como um arquivo ASP (Página Ativa no Servidor).

Quando o usuário solicitar a página com a extensão ASP:

?? O servidor Web inicia os scripts.

?? Processa o código contido nos scripts.

?? Gera o código HTML para retornar o resultado ao usuário através do navegador.

Embora você possa usar até o Notepad para criar páginas ASP, editores como o 1st Page 2000, o Microsoft Visual InterDev ou outros editores de código Web, fornecem uma interface muito mais poderosa.

A decisão de usar script no servidor ou script no cliente depende da funcionalidade que você deseja dar para as suas páginas Web. Por exemplo, se você quiser validar os dados que o usuário digitou em um campo de CPF em um formulário de uma página Web, um script no cliente é, sem dúvida, o método mais apropriado. O código-fonte do script é executado pelo navegador e apenas os dados válidos são submetidos para o servidor.

Porém, se você estiver construindo um Web Site para comércio eletrônico, onde os dados são recuperados e atualizados em uma base de dados on-line, você pode usar scripts no servidor para recuperar os dados da base de dados antes de retornar páginas Web para os seus usuários.

Linguagens de Script

Existem atualmente duas linguagens de script mais utilizadas: *Visual Basic Scripting Edition* (VBScript) e *JavaScript*, sendo que esta última ainda possui uma variação desenvolvida pela Microsoft denominada *JScript*.

Visual Basic Scripting Edition

O *Visual Basic Scripting Edition* é um subconjunto da linguagem Visual Basic da Microsoft para Aplicativos. O VBScript é suportado apenas pelo navegador Microsoft Internet Explorer versão 3.0 e superiores.

JavaScript

O JavaScript é um a linguagem *C-like* baseada em Java, desenvolvida em parceria pela Sun Microsystems Inc. e pela NetScape Communication Corporation. JavaScript é suportada pelo navegador NetScape Navigator versão 2.0 ou superior e pelo navegador Microsoft Internet Explorer versão 3.0 ou superior. Há, porém, algumas pequenas diferenças de sintaxe entre o JavaScript suportado pelo Microsoft Internet Explorer e o pelo NetScape Navigator. Estas diferenças deram origem à uma variação do JavaScript denominada JScript, aceita apenas pelo Microsoft Internet Explorer, e devem ser observadas quando da construção de páginas Web que devem funcionar perfeitamente nos dois navegadores.

O *Visual Basic Scripting Edition* e o *JavaScript* são bem parecidos em termos de funções e recursos. Ambos são orientados para objeto e suportam a criação de objetos, resposta a eventos, funções, métodos e propriedades. A sintaxe é a principal diferença entre os ambos.

A Tag <SCRIPT>

Qualquer código script sempre deve estar contido dentro das tags <SCRIPT> e </SCRIPT>. O atributo LANGUAGE diz ao navegador que interpretador deve ser utilizado quando o código for executado. Para o VBScript, deve-se configurar o atributo LANGUAGE para "VBScript"; para JavaScript, deve-se configurar o atributo LANGUAGE para "JavaScript". Você precisa especificar a linguagem script a ser utilizada porque alguns navegadores, como Microsoft Internet Explorer, podem usar mais de uma linguagem de script e você pode utilizar mais de uma linguagem script numa mesma página Web.

Nota: no Internet Explorer, se você não configurar o atributo LANGUAGE, o navegador assume que você está executando JavaScript. Se seu código for em VBScript, você receberá erros de sintaxe.

Ao escrever scripts, você deve colocar todo o código de cada procedimento dentro da mesma tag <SCRIPT>. Você pode ter vários procedimentos numa mesma tag <SCRIPT>, mas não pode dividir o código de um mesmo procedimento entre duas tags <SCRIPT>.

Embora sua página HTML possa conter mais de uma tag <SCRIPT>, a manutenção do código se tornará muito mais fácil se todo ele estiver contido no mesmo lugar. Você pode colocar a tag <SCRIPT> no <BODY> ou, de preferência, nas seções <HEAD> das páginas HTML.

Usando Visual Basic Scripting

O trecho de código exemplificado a seguir é escrito em VBScript. Será executado quando o usuário clicar um botão denominado btnHello da página Web:

```
<SCRIPT LANGUAGE="VBScript">
<!--
Sub btnHello_OnClick()
      MsgBox "Oi, mundo!"
End Sub
-->
</SCRIPT>
<INPUT TYPE=BUTTOM NAME=btnHello VALUE="Me clique">
```

Nota: navegadores que não entendem a tag <SCRIPT> exibem o código na página HTML como se ele fosse um texto regular. Por esse motivo, sempre coloque o código de scripting entre tags de comentário (<!-- e -->) para evitar que navegadores que não entendem que a tag <SCRIPT> exibam o código-fonte na página HTML como se fosse um texto.

Usando JavaScript

O código exemplificado a seguir é escrito em JavaScript. Ele cria uma função que solicita ao usuário a sua identificação (ID). A tag HTML para o botão cmdTest especifica o procedimento a ser executado quando um usuário clicar o botão.

```
<SCRIPT LANGUAGE=JavaScript>
var ID
function getid() {
      ID = prompt("Entre seu número de identificação:");
}
</SCRIPT>

<INPUT TYPE=buttom NAME=cmdTest OnClick="getid();">
```

Executando Scripts

A localização de um script dentro da seção <SCRIPT> de uma página Web determina quando o script deve ser executado. Em geral, você pode adicionar scripts nas seguintes áreas:

?? Nas linhas de código:

   Se você adicionar scripts fora de um procedimento, o script é executado quando o navegador o encontrar, enquanto estiver fazendo o *download* da página, do servidor para o cliente. Isto é útil se você quiser inicializar dados ou objetos na página.

?? Procedimentos:

   Se você adicionar scripts em um procedimento, o código do script é executado quando o procedimento for explicitamente invocado.

?? Procedimento Associado a Evento:

   Se você adicionar scrips em um procedimento associado a um evento, o script é executado quando o evento ocorrer. Por exemplo, se você criar um procedimento de evento OnClick para um botão, o script é executado quando o usuário clicar o botão.

Componentes Ativos

Os objetos como controles ActiveX e Java Applets desempenham um papel importante na construção de conteúdo de Web interativa. Em uma página HTML, estes tipos de objetos aprimoram a interface de usuário.

Existem centenas de controles e Applets que você pode usar em uma aplicação de Web. Uma vez identificada a funcionalidade que você deseja, você pode selecionar o objeto (um controle ActiveX ou um Java Applet) que fornece aquela funcionalidade, e então simplesmente inserir o objeto na sua página Web.

Segurança

Para assegurar um método seguro e confiável de distribuir software através da Internet, a Microsoft e outras companhias de software desenvolveram um método de assinatura digital do código.

Controles Assinados

Para assinar o código de um controle, o desenvolvedor do controle obtém um certificado digital de uma autoridade de certificação. Um controle com assinatura digital não é necessariamente um controle seguro. O certificado digital só garante que o código sendo carregado foi construído e assinado por um desenvolvedor qualificado, e que ele não foi alterado ou corrompido por terceiros.

Os usuários podem determinar se preferem carregar e executar controles ActiveX ou Java Applets definindo o nível de segurança do navegador e descobrindo se o controle foi assinado para aquele nível com um certificado digital.

Você define opções de segurança na caixa de diálogo Opções de Segurança do Microsoft Internet Explorer. Quando o nível de segurança padrão for selecionado, você pode especificar como quer carregar controles assinados ou não. As seguintes opções estão disponíveis:

Opção	Descrição
Enable	Carregue automaticamente o controle.
Prompt	Avise o usuário antes de carregar.
Disable	Não carregue o controle.

Níveis de Segurança

Como você assinou controles e um nível de segurança do navegador for selecionado, você pode especificar como Internet Explorer trabalhará com o controle. As seguintes opções estão disponíveis.

Opção	Descrição
Enable	Execute/rode automaticamente.
Prompt	Avise o usuário antes de executar/rodar.
Disable	Não execute/rode.

Controles Seguros

Os controles que podem ser inicializados com valores em tags HTML <PARAM> devem ser marcados como seguros por serem inicializados antes do Internet Explorer ler as tags <PARAM>.

Quando Internet Explorer encontrar um controle em uma página Web que tem etiquetas de inicialização, ele questionará o controle para determinar se é seguro para inicialização. O

resultado deste questionamento, combinado com o nível de segurança fixado pelo usuário, determina se o controle será inicializado.

Controles Seguros para Scripting

Quando o desenvolvedor de controles marcar um controle como seguro para scripting, estará garantindo que o modelo de objeto do controle não causará um problema de segurança, quer seja na forma de corrupção de dados ou vazamentos de segurança.

Um controle marcado como seguro para scripting garante que não existem quaisquer propriedades ou métodos disponíveis para uso pelo código de script que pode danificar o computador do usuário. Por exemplo, um controle que permite que informações sejam gravadas no disco rígido não deve ser marcado como seguro para scripting.

Quando o Internet Explorer encontrar um controle em uma página de Web, ele questiona o controle para determinar se é seguro para scripting. O resultado deste questionamento, combinado com o nível de segurança definido pelo usuário, determina se o controle está disponível para ser utilizado no script ou não.

Controles ActiveX

Os controles de ActiveX são objetos ou componente que você pode inserir em uma página Web ou outras aplicações que podem se tornar recipientes ActiveX.

Estes controles:

?? Podem ser construídos com linguagens como C, Visual C++, Visual Basic, Delphi e Java.

?? São tipicamente construídos como *dynamic-link-libraries* (DLLs).

?? São compilados com uma extensão .OCX

A ilustração a seguir mostra como controles ActiveX trabalham em uma página Web.

Como os Controles ActiveX Trabalham

Para executar um controle ActiveX em uma página Web ou em uma aplicação, o computador do usuário já deve ter instalado e registrado o controle. Se o controle exigir funcionalidade adicional de DLLs, as DLLs devem ser também instaladas no computador do usuário.

Quando você cria uma aplicação isolada que usa controles de ActiveX, normalmente será necessário criar um programa de instalação que:

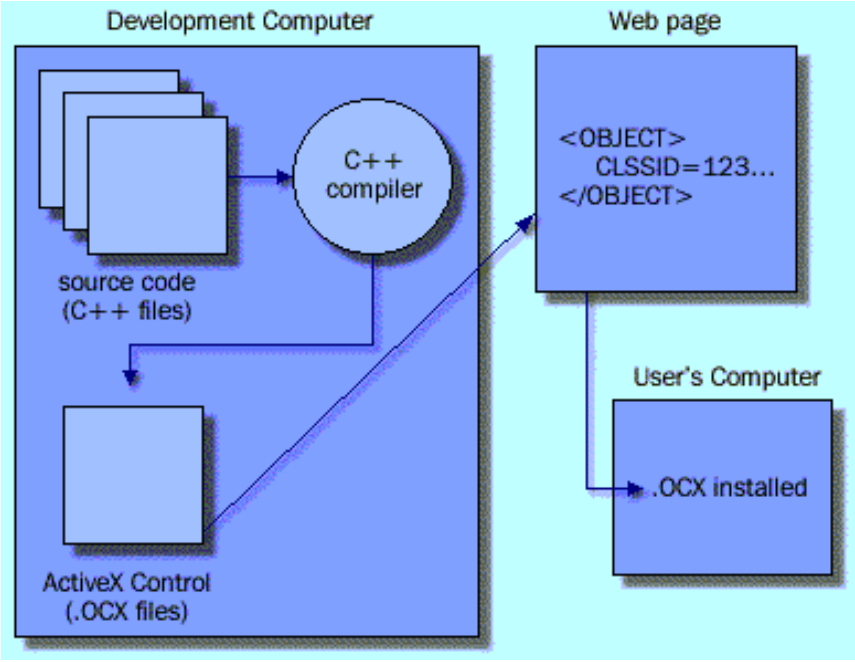
?? Instale a aplicação, os arquivos .OCX, e quaisquer DLLs necessárias no computador do usuário.

?? Registre os controles ActiveX no registro do Windows.

Como resultado, quando a aplicação for executada, os arquivos .OCX devem estar prontos para uso.

As páginas Web não têm programas de instalação, portanto os arquivos .OCX e as DLLs podem ser carregados e instalados quando o usuário acessar a página Web se necessários. O controle deve ser instalado antes que a página Web possa ser executada corretamente.

Para inserir controles em uma página Web e assegurar que eles estão corretamente instalados, você utiliza a tag <OBJECT> e define o atributo CODEBASE.



Download de Componentes via Internet

O Internet Component Download é um componente do Internet Explorer que procura por controles ActiveX e então os carrega se eles não estiverem presentes no computador do usuário.

Como os Componentes são Carregados

Quando Internet Explorer encontrar uma tag <OBJECT>, tenta localizar o controle no computador do usuário o primeiro procurando no registro do Windows. Se o controle não está registrado, o navegador carrega o objeto na localização especificada no atributo CODEBASE.

Se o controle é registrado no computador do usuário, o Internet Explorer tenta carregar (ou instanciar) o objeto. Se esta operação falha, o Internet Explorer tenta carregar o controle como se o objeto não fosse registrado.

A ilustração a seguir mostra a um gráfico que ilustra como o Microsoft Internet Explorer carrega controles ActiveX.

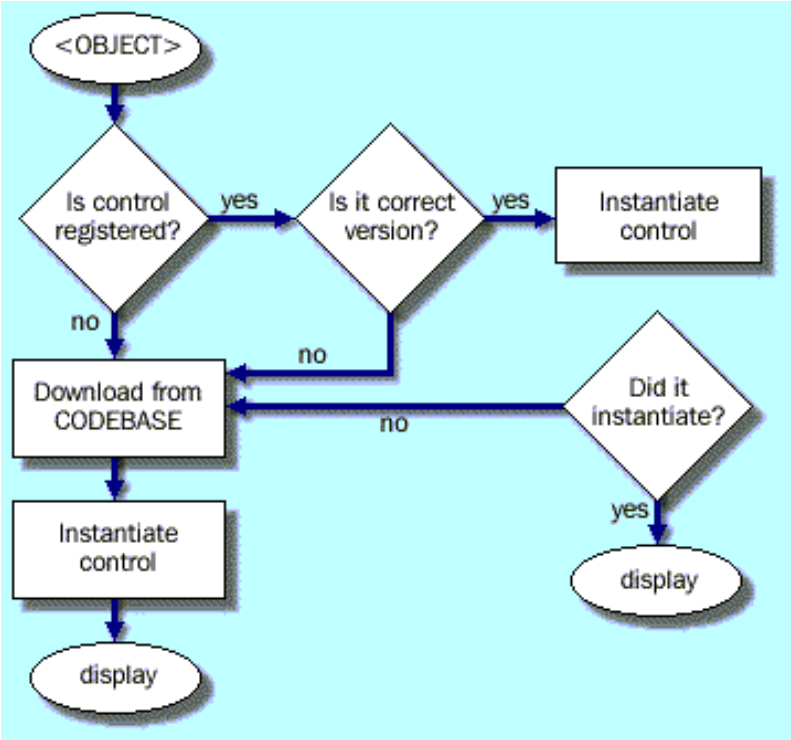
Os controles carregados são instalados na pasta \Windows\Occache. Não são automaticamente removidos quando o usuário esvaziar a pasta de arquivos Internet temporários.

O Caminho de Procura na Internet

Se você não especificar um atributo CODEBASE para um objeto, ou se a localização que você especificar não contiver os arquivos que desejados, o Internet Explorer tenta automaticamente localizar o objeto usando o caminho de procura na Internet.

O caminho de procura na Internet é uma lista de servidores que armazenam objetos e que é pesquisado cada vez que componentes são carregados. Um armazém de objetos no caminho de procura na Internet é um servidor HTTP que processa pedidos para código que pode ser carregado.

Ao procurar um objeto, o Internet Explorer verifica localizações no caminho de procura na Internet. Ele sempre usa a primeira resposta bem sucedida de um servidor e não continua a procurar por versões de componentes mais novas.



Java Applets

A linguagem Java, é uma derivação da linguagem C++, desenvolvida pela Sun Microsystems Inc. para ser uma linguagem robusta, segura e multi-plataforma a ser usada na Internet. Você pode usar Java para criar aplicações isoladas e Java Applets.

Applets são um tipo de objeto que podem ser inseridos em uma página HTML. Os Java Applets são programas pequenos, reutilizáveis que expõem uma interface padrão e são executados em um recipiente como um navegador Web.

Como Java Applets Trabalham nas Páginas Web

Quando arquivos-fonte em Java (.java) são compilados, o compilador Java cria arquivos Java bytecode (.class). Quando estes arquivos de classe são carregados para o computador de um usuário, eles são localmente interpretados pela Máquina Virtual de Java Virtual (JVM - *Java Virtual Machine*). Você pode instalar o JVM como parte de um navegador Web, ou carregar-lo da Internet.

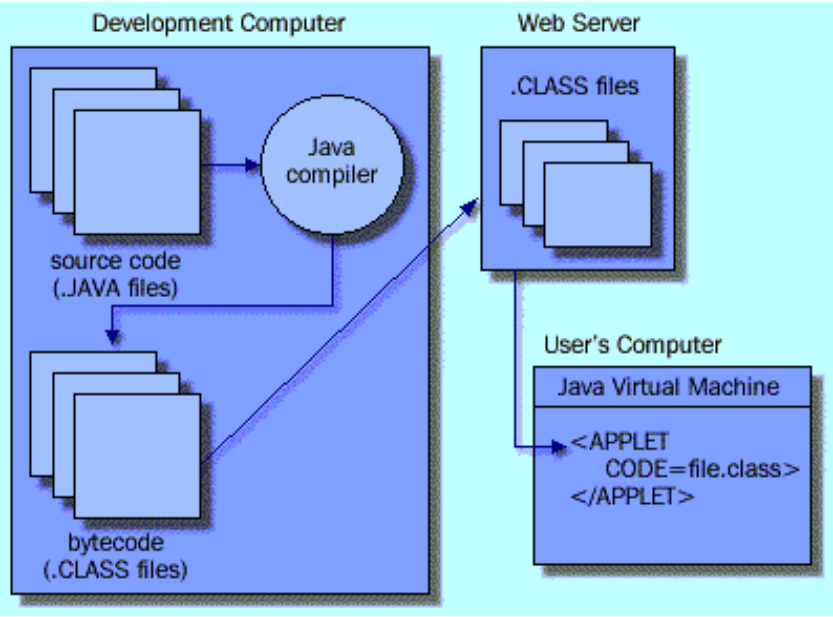
Para adicionar Java Applets em uma página HTML, você usa a tag <APPLET>. O atributo CODE desta tag especifica o arquivo de classe do applet.

Para um applet ser executado em um navegador Web:

?? O computador do usuário deve ter o JVM instalado.

?? A segurança do navegador deve ser configurada de forma a habilitar a execução de Applets.

A ilustração a seguir mostra como arquivos de Java são interpretados.



Controles ActiveX vesus Java Applets

Obviamente, existem vantagens e desvantagens a usar controles ActiveX e Java Applets nas páginas Web.

Ao usar controles ActiveX, você deve considerar os seguintes aspectos:

?? Desempenho:

Os controles de ActiveX são compilados em código nativo, portanto, são executados mais rapidamente do que Java Applets.

?? Familiaridade para o desenvolvedor:

Muitos programadores de Visual Basic e Visual C já usam controles ActiveX por conhecer melhor estas linguagens.

?? Disponibilidade:

Existem muitos controles ActiveX que são incluídos com Visual Basic ou que podem ser adquiridos de um desenvolvedor autônomo. Além disso, vários controles estão livremente disponíveis na Web.

?? Suporte do navegador:

Os controles ActiveX serão executados em qualquer navegador que suporte a tag <OBJETO>. Para navegadores que suportam apenas a tag <EMBED>, um plug-in da Netscape está disponível.

?? Plataforma específica:

Como os controles ActiveX são compilados em código nativo, eles trabalham só nas plataformas para as quais foram construídos, isto é, Windows. Porém, você pode fornecer um arquivo que carrega o controle correto para a plataforma em que o navegador estiver sendo executado.

?? Segurança:

Os controles ActiveX podem ser programados para ser maliciosos e perigosos. Para adicionar segurança para suas páginas Web, você deve sempre usar controles com assinatura. Os usuários do Internet Explorer podem ver o certificado digital de um controle com assinatura antes de decidir carregá-los.

Ao utilizar Java Applets, você deve considerar os seguintes aspectos:

?? Desempenho:

Os Java Applets são executados mais lentamente do que arquivos que são compilados em código nativo. Devido à natureza interpretada da linguagem Java, este aspecto é especialmente percebido em Java Applets grandes e em aplicações Java. Tecnologias recentes, como compiladores just-in-time, resolvem parcialmente este problema.

?? Tamanho reduzido:

O Java Applets tendem a ser menores que outros tipos de arquivos executáveis, porque as funções de biblioteca são residente na Máquina Virtual de Java (JVM) e não são trazidas para o computador do usuário quando os Applets são carregados.

?? Independência de navegador:

O único requisito para um Java Applet poder ser executado em um navegador é que o navegador suporte o JVM. Quando um applet for executado, parecerá que está sendo executado pelo navegador. Porém, o Applet está realmente sendo completamente executado pelo JVM.

Java é uma linguagem em evolução. Portanto, navegadores habilitados em Java e JVM podem se comportar diferentemente para especificações diferentes e executar Java

Applets ligeiramente diferentemente do planejado. A medida que a linguagem Java amadurece, este problema tende a ser reduzido.

?? Independência de plataforma:

A linguagem Java e a arquitetura JVM foram projetados para serem executados em qualquer plataforma.

?? Segurança:

Os Java Applets são implementados com um modelo de segurança rígida. Neste modelo, o Applet é executado em seu próprio espaço, ou "*sandbox*." O modelo *sandbox* não permite a um Applet executar as seguintes tarefas:

- Executar operações de entrada/saída de arquivos ou bancos de dados.
- Executar chamadas para o sistema operacional.
- Iniciar ou se comunicar com outras aplicações fora da página Web atual.
- Abrir um canal de comunicação para um servidor diferente do servidor do qual foi carregado.

Por causa destas restrições, é extremamente difícil para um desenvolvedor de Java Applets intencionalmente ou acidentalmente criar um Applet malicioso ou perigoso.

Nota: o Internet Explorer categoriza classes Java como confiáveis ou não confiáveis. As classes não confiáveis são executadas em seu próprio espaço de aplicação, dentro da *sandbox*, portanto, eles não podem usar serviços COM. As classes confiáveis são as únicas classes que podem usar o Modelo de Objetos de Componentes (COM) e serviços, estando livres das restrições da *sandbox*.

A tag <OBJECT>

A tag <OBJECT> possui atributos que definem o controle ActiveX, sua localização, e como o controle será exibido em uma página HTML. Os parágrafos seguintes descrevem estes atributos.

CLASSID

Quando um controle ActiveX estiver instalado em computador de um usuário, é registrado no registro do sistema com uma classe com um identificador (ID) único. O atributo CLASSID, que contém a identificação da classe (ID), é o único atributo exigido para a tag <OBJECT>.

A sintaxe para o atributo CLASSID é:

```
CLASSID = "clsid:12345678-1234-1234-1234-123456789012"
```

O código exemplificado a seguir mostra a tag <OBJECT> para o um controle de Calendário:

```
<OBJETO
      classid="clsid:8E27C92B-1264-101C-8A2F-040224009C02"
>
</OBJECT>
```

Quando você usa o Microsoft FrontPage para inserir um controle, o FrontPage procura automaticamente o registro para a classe ID do controle selecionado e o grava na tag <OBJECT> gerada. Você pode determinar a classe ID para um controle manualmente usando uma das seguintes ferramentas:

?? Editor de Registro (*RegEdit*):

Esta ferramenta habilita você a mudar definições no registro do seu sistema. Você pode utiliza-lo também para ver e copiar a classe ID de um objeto. Para iniciar o Editor de Registro, execute regedit.exe.

?? Visualizador de OLE (*Ole2View*):

Esta aplicação está disponível no Visual Basic.

ID

O atributo ID o habilita a se referir a um objeto construído com o Visual Basic Scripting Edition (VBScript).

O código exemplificado a seguir mostra como fixar o atributo ID do controle de Calendário:

```
<OBJECT
      classid="clsid:8E27C92B-1264-101C-8A2F-040224009C02"
      id=cldCalendar2
>
</OBJECT>
```

CODEBASE

O atributo CODEBASE é uma URL que aponta para um arquivo contendo a implementação de um objeto.

WIDTH, HEIGHT, ALIGN, HSPACE, VSPACE, e BORDER

Estes atributos afetam a forma com a qual um objeto é posicionado e dimensionado numa página HTML, e se ele conterá ou não uma borda.

Provendo uma Alternativa para a tag <OBJECT>

Para navegadores que não suportam a tag <OBJECT>, ou para usuários que não habilitaram controles ActiveX em seu navegador Web, você deve prover um caminho alternativo para fornecer a mesma funcionalidade nas suas páginas. Se um navegador Web não exibe controles ActiveX, ele exibirá quaisquer tags HTML que forem colocadas entre as tags <OBJETO> e </OBJECT>.

Nota: se um navegador suporta a tag <OBJECT>, mas o usuário não instalou o controle ActiveX, o navegador exibirá um ícone de controle inválido, e não a funcionalidade alternativa.

Por exemplo, para navegadores que não exibem o controle de Calendário, você pode colocar um controle de caixa de texto na página Web para um usuário entrar a data, como exemplificado no código seguinte:

```
<OBJECT
      classid="clsid:8E27C92B-1264-101C-8A2F-040224009C02"
>Melhor visualizado com o Internet Explorer.<P>
Digite uma data aqui:
<INPUT TYPE=TEXT NAME="Date" MAXLENGTH=10 SIZE=10>
</OBJECT>
```

O Atributo CODEBASE

O atributo CODEBASE para a tag <OBJECT> contém uma URL absoluta ou relativa que aponta para o arquivo .OCX do controle especificado.

Se um controle ActiveX não possuir quaisquer outros arquivos diferentes de .OCX, você pode especificar a localização do arquivo .OCX com o atributo CODEBASE.

O código exemplificado a seguir carrega um controle ActiveX fornecendo a referência ao arquivo .OCX:

```
<OBJECT
      CLASSID="clsid:99B42120-6EC7-11CF-A6C7-00AA00A47DD2"
      CODEBASE="http://server/control.ocx">
</OBJECT>
```

Se um controle ActiveX possuir Dlls adicionais para ser executado, você deve prover também:

?? Um arquivo .CAB que contém a relação de todos os arquivos necessários.

?? Um arquivo .INF que especifica onde instalar cada um dos arquivos necessários.

O código exemplificado a seguir carrega e instala um controle ActiveX provendo a referência necessária através de um arquivo .CAB:

```
<OBJECT
      CLASSID="clsid:99B42120-6EC7-11CF-A6C7-00AA00A47DD2
      CODEBASE="http://server/control.cab">
</OBJECT>
```

Especificando um Número de Versão

Com o atributo CODEBASE, você pode incluir também um número da versão para ter certeza que os usuários só carregarão uma versão específica do controle.

No código exemplificado a seguir, o atributo CODEBASE especifica o número de versão 4.70.1165:

```
<OBJECT
      CLASSID="CLSID:7823A620-9DD9-11CF-A662-00AA00C066D2"
      CODEBASE="iemenu.ocx#Versão=4,70,0,1165">
```

O formato do número de versão é N,N,N,N. Se você não especificar um número de versão, o Internet Explorer usa qualquer versão do controle que estiver instalada no computador do usuário.

Para ter certeza de que um controle sempre será carregado para o computador do usuário, especifique o número de versão – 1, – 1, – 1, – 1.

Definindo as Propriedades dos Controles

Os valores iniciais que você estabelece para as propriedades de um controle ActiveX definem como o controle será exibido quando o navegador carregá-lo para a página HTML.

Para definir as propriedades iniciais de objeto para um controle ActiveX, você usa as tags <PARAM>.

A tag <PARAM> usa a seguinte sintaxe:

```
<PARAM NAME="ParameterName" VALUE="Valor">
```

Para cada propriedade do controle, você usa uma tag <PARAM> separada.

O código exemplificado a seguir mostra a tag <OBJECT> para o controle ActiveX Label. O controle tem valores de propriedade iniciais definidas para Ângulo, Alinhamento e Legenda.

```
<OBJECT
      classid="clsid:99B42120-6EC7-11CF-A6C7-00AA00A47DD2"
      id=lblActiveLbl
      width=250
      height=250
>
<PARAM NAME="Angle" VALUE="90">
<PARAM NAME="Alignment" VALUE="2">
<PARAM NAME="Caption" VALUE="Hello, World">
</OBJECT>
```

Nota: para determinar os nomes de propriedade e valores suportados por um objeto, veja o arquivo de Ajuda (Help) do objeto, ou use o Object Navegador do Microsoft Visual Basic da Microsoft.

Usando Java Applets

Para inserir um Java Applet, você adiciona a tag <APPLET> em um arquivo HTML. A tag <APPLET> tem vários atributos que definem o applet, sua localização, e como ele será apresentado na página HTML.

Os seguintes atributos são obrigatórios:

?? CODE

Identifica o nome do arquivo .CLASS a ser carregado (este atributo é sensível a letras maiúsculas e minúsculas).

?? WIDTH

Especifica a largura da área em que o applet será executado.

?? HEIGHT

Especifica a altura da área em que o applet será executado.

O código exemplificado a seguir mostra como usar a tag <APPLET> para Java Applet Outline:

```
<APPLET
      CODE=Outline.class
      HEIGHT=150
      WIDTH=200>
</APPLET>
```

O código a seguir mostra a sintaxe para a definição dos atributos da tag <APPLET>:

```
<APPLET
      ALIGN=LEFT|CENTER|RIGHT|TOP|MIDDLE|BOTTOM
      CODE=appletFile
      CODEBASE=codebaseURL
      HEIGHT=pixels
      HSPACE=pixels
      NAME=appletInstanceName
      VSPACE=pixels
      WIDTH=pixels>
</APPLET>
```

A tabela a seguir define os atributos que você pode configurar para a tag <APPLET>

Atributo	**Definição**	**Exigido**
ALIGN	Especifica o alinhamento do applet em relação ao texto. Os valores possíveis são LEFT, CENTER, RIGHT, TOP, MIDDLE e BOTTOM. O valor padrão é LEFT.	Não
ARCHIVE	Nome do arquivo .ZIP que contém os arquivos .CLASS usados pelo applet.	Não
CODE	Nome do arquivo que contém a subclasse compilada. Este arquivo é relativo para a URL do applet (a pasta em que o applet está localizado). O arquivo não pode ser um valor absoluto.	Sim
CODEBASE	Especifica a URL do applet. Se este atributo não for especificado, a URL da página HTML será usada.	Não
WIDTH e HEIGHT	Especificam a largura e a altura inicial (em pixeis) para a área de exibição do applet. Esta área não inclui quaisquer janelas ou caixas de diálogo invocadas pelo applet.	Sim
VSPACE e HSPACE	VSPACE especifica o número de pixeis acima e abaixo do applet. HSPACE especifica o número de pixeis em cada lado do applet.	Não
NAME	Identifica um applet para outro applets na mesma página HTML.	Não

Provendo uma Alternativa para a tag <APPLET>

Para navegadores que suportam a tag <APPLET>, ou para usuários que não habilitaram Java Applets em seu navegador Web, você deve prover um caminho alternativo para fornecer a mesma funcionalidade na página. Se um navegador Web não exibe Java Applets, ele exibirá quaisquer tags HTML que você colocar entre as tags <APPLET> e </APPLET>.

No código exemplificado a seguir, o HTML alternativo informa aos usuários que o Java Applet não está disponível, e usa hyperlinks para fornecer a mesma funcionalidade que Java Applet Outline:

```
<APPLET
      CODE=Outline.class
      HEIGHT=150
      WIDTH=200>
Você não pode executar o Java applet Outline.
<UL>
<LI><A HREF="default.htm">State University Home Page</A></LI>
<LI><A HREF="majors.htm">State University Majors</A></LI>
</UL>
</APPLET>
```

Definindo as Propriedades de um Java Applet

De forma semelhante a controles ActiveX, valores de propriedade iniciais definem como um Java Applet aparecerá quando o navegador carregar a página HTML.

Para definir as propriedades iniciais de Java Applets, você utilizar as tags PARAM>.

A seguinte sintaxe é utilizada para as tags <PARAM>:

```
<PARAM NAME=appletParameter1  VALUE=value>
```

Por exemplo, para especificar o arquivo que contém informações para o Java Applet Outline, você define o parâmetro TOCFile, como exemplificado no código a seguir:

```
<APPLET
      CODE=Outline.class
      WIDTH=150
      HEIGHT=200 >
      <PARAM NAME=TOCFile VALUE="contents.toc">
</APPLET>
```

Se os arquivos .CLASS para um Java Applet estiverem localizados em uma pasta diferente da que está a página HTML, você deve adicionar o atributo CODEBASE para uma tag <APPLET>. O valor do atributo CODEBASE pode ser uma URL absoluta ou relativa.

O código exemplificado a seguir mostra a tag <APPLET> para o Java Applet Outline contido na pasta relativa applets/javaapps:

```
<APPLET
      CODE=Outline.class
      CODEBASE="applets/javaapps"
>
</APPLET>
```

Distribuição de Java Applets

Por definição, Java Applets são carregados um arquivo .CLASS de cada vez e em um formato não comprimido. À medida que os applets se tornam mais complexos e contam com mais de um arquivo, este método de carga se torna lento e ineficiente.

Usando Arquivos .CAB

A uma maneira de reduzir o tempo de carga (*download*) é para usar um arquivo .CAB para comprimir e distribuir múltiplos arquivos .CLASS. Você pode criar arquivos .CAB usando o utilitário Cabarc, que é incluído com o Kit de Desenvolvimento de Software da Microsoft (SDK) para Java. Este utilitário está também incluído no Internet Client SDK.

Para usar um arquivo .CAB para um Java Applet, você adiciona uma tag <PARAM> com o atributo de NAME definido para cabbase, e o atributo de VALUE definido para o nome do arquivo .CAB, como mostrado no código do exemplo a seguir:

```
<APPLET
      CODE=Outline.class
      WIDTH=150
      HEIGHT=200>
      <PARAM NAME=cabbase VALUE=outline.cab>
</APPLET>
```

Este código informa a Máquina Virtual de Java (JVM) os arquivos .CLASS do Java Applet que devem ser extraídos do arquivo .CAB, em vez de ser carregado um arquivo de .CLASS de cada vez. Um arquivo .CAB deve estar localizado na mesma pasta que a página HTML ou na localização especificada com o atributo CODEBASE.

Usando Arquivos .ZIP

O Internet Explorer é o único navegador que suporta arquivos .CAB. Para carregar Java Applets mais rapidamente em outros navegadores, você pode usar um arquivo compactado .ZIP, com todos os arquivos .CLASS necessários para o seu Java Applet. O JVM lê o arquivo .ZIP extrai os arquivos .CLASS dele.

Para usar um arquivo .ZIP, defina o atributo de ARCHIVE da tag <APPLET> para o nome do arquivo .ZIP.

Você pode definir o atributo ARCHIVE e adicionar o cabbase da tag <PARAM>, como ilustrado no código do exemplo a seguir:

```
<APPLET
      CODE=Outline.class
      WIDTH=150
      HEIGHT=200
      ARCHIVE=outline.zip>
      <PARAM NAME=cabbase VALUE=outline.cab>
</APPLET>
```

HTML Dinâmico - DHTML

Introdução

O HTML Dinâmico é um novo conjunto de recursos suportado pelos navegadores Microsoft Internet Explorer e pelo NetScape Navigator a partir das versões 4.0 e superiores, que permite a criação de páginas Web altamente interativas e intuitivas. Com o HTML Dinâmico você pode:

- ?? Utilizar gráficos e textos animados em uma página Web.
- ?? Tratar conjuntos de elementos de uma página como um grupo, ou seja, como se fosse um único elemento, alterando-os todos de uma só vez.
- ?? Fazer aparecer ou substituir textos a partir de ações do usuário como movimentos ou cliques do mouse.
- ?? Chamar métodos em controles ActiveX e Java Applets.
- ?? Adicionar transições entre as páginas e obter diversos efeitos visuais.

O HTML Dinâmico dá a você o poder para criar documentos HTML com um visual extremamente sofisticado, e que dinamicamente mudam seu conteúdo e interagem com o usuário, sem exigir programas no servidor Web ou conjuntos complexos de páginas HTML para obter efeitos especiais.

Você pode usar HTML Dinâmico para:

- ?? Esconder textos e imagens em seu documento e manter este conteúdo escondido até que um dado tempo decorra ou que o usuário execute uma ação para vê-los.
- ?? Animar texto e imagens em seu documento, movendo independentemente cada elemento de um ponto para outro dentro do documento, seguindo o caminho que você escolher ou que você deixar o usuário escolher.
- ?? Criar um relógio ou um texto que atualiza automaticamente seu conteúdo com as notícias mais recentes, citações, ou outros dados.
- ?? Criar um formulário, e então ler, processar e responder os dados que o usuário digitar, sem ter que retornar para o servidor Web para processamento.

O HTML Dinâmico alcança estes efeitos modificando o conteúdo do documento original, re-formatando automaticamente e reapresentando o documento para mostrar estas mudanças. Não é necessário recarregar o documento, carregar um novo documento, ou depender do servidor Web para gerar um novo conteúdo. O HTML Dinâmico calcula e executa as mudanças no computador do usuário utilizando os recursos disponíveis nas novas versões dos navegadores modernos.

O Modelo de Objetos

Os elementos dinâmicos são tags HTML que o modelo de objetos do HTML Dinâmico define e utiliza. Usando as tags do modelo de objetos, você pode acessar e manipular todos elementos HTML em um documento. Os elementos estão disponíveis como objetos individuais, portanto é possível examinar e modificar um elemento e seus atributos, lendo e modificando suas propriedades, além de chamar seus métodos. Você pode manipular ou até mudar o texto dentro de um elemento através das propriedades e métodos que podem ser configuradas para os elementos.

O exemplo a seguir, em VBScript, torna uma porção definida de texto verde, quando o evento que invoca o procedimento changeMe for disparado:

```
<script language="VBScript">
      Sub changeMe()
            window.document.body.style.color = "green"
      End Sub
</script>
```

O modelo de objetos executa também ações de usuário, como pressionar uma tecla e clicar o mouse, disponíveis como eventos. Você pode interceptar e processar estes e outros eventos criando um manipulador de eventos através de funções e rotinas desenvolvidas utilizando as linguagens de script. O manipulador de eventos recebe o controle quando um dado evento acontece e pode executar qualquer ação apropriada, inclusive usar o modelo de objetos para mudar o documento. O exemplo a seguir cria um manipulador de eventos para interceptar um clique do mouse em qualquer lugar do corpo do documento. Quando um clique acontecer, o procedimento changeMe será executado.

```
<body onclick="ChangeMe">
```

Existem muitos meios para acessar elementos em um documento. Você pode selecionar cabeçalhos, parágrafos, divisões e outros elementos de um documento para controlar efeitos dinâmicos. O caminho mais fácil para controlar elementos em um documento é designar um identificador (ID) para cada elemento.

O código do exemplo a seguir designa o identificador "MyHeading" para a propriedade ID da tag <H3>:

```
<H3 ID="MyHeading">HTML Dinâmico!</H3>
```

Você pode então, se referir a este elemento usando o identificador ou ID que você designou para o texto contido na tag. Você pode, então, acessar todos os elementos em um documento através de uma coleção.

O código exemplificado a seguir muda a cor do elemento para verde:

```
window.document.all.MyHeading.style.color = "green"
```

O Objeto Document

O objeto documento representa um documento HTML em uma janela do navegador. Você pode usar o objeto documento para:

- ?? Recuperar informações sobre o documento.
- ?? Examinar e modificar os elementos HTML e textos contidos no documento.
- ?? Processar eventos.

Você acessa o objeto documento através do objeto janela, entretanto, você pode omitir o objeto janela.

O exemplo a seguir exibe uma mensagem quando o documento mudar o seu estado quando o evento **onreadystatechange** acontecer, isto é, quando o documento for carregado no navegador:

```
<html>
<head>
      <title>Teste</title>
</head>

<script for="document" event="onReadyStateChange" language="VBScript">
      Msgbox "A página " & window.document.title & " foi carregada!"
</script>

<body>

</body>
</html>
```

Você pode acessar e manipular todos os elementos em um documento HTML. Cada elemento tem propriedades, métodos e eventos associados a ele.

Abaixo são relacionados alguns dos elementos comuns que você pode acessar através do modelo de objetos do HTML Dinâmico.

Tag	Objeto	Descrição
<A>	A	Designa o começo ou destino de um hyperlink. O elemento de âncora exige que você especifique o HREF ou a propriedade NAME.
<IMG>	IMG	Embute uma imagem ou um clipe de vídeo no documento.
<BODY>	BODY	Especifica o início e fim do corpo de um documento.
<P>	P	Denota um parágrafo.
<H1>, <H2>, etc.	H1, H2, etc.	Formata texto como um estilo de cabeçalho.

O código a seguir muda a cor de todo o texto no corpo do documento para verde, quando um usuário clicar em qualquer lugar no documento:

```
<html>
<head>
<title>Welcome</title>
<script language="VBScript">
      Sub changeMe()
            document.body.style.color = "green"
      End Sub
</script>
</head>
<body onclick="changeMe()">
<h3>Efeitos em HTML Dinâmico!</h3>
</body>
</html>
```

Elementos de Agrupamento

Usando as tags <SPAN> e <DIV>, você pode interceptar eventos para elementos múltiplos de uma só vez, ou acessar propriedades e métodos para eles.

As tags <SPAN> e <DIV> servem para um propósito semelhante, mas produzem, resultados ligeiramente diferentes:

?? A tag <SPAN> cria um span, que é um agrupamento lógico de elementos em uma página. Esta tag é freqüentemente usada para afetar uma porção de texto em um elemento a nível de bloco, como um parágrafo.

?? A tag <DIV> é usada para criar um recipiente ou container, que você deseja separar do resto do conteúdo de documento e, opcionalmente, configurar para ele propriedades de caixas, como bordas e margens.

Você pode usar as tags <SPAN> e <DIV> para:

- ?? Aplicar estilos.
- ?? Acessar propriedades e métodos.
- ?? Interceptar eventos.

A tag <DIV> define um bloco que consiste em tags de texto e HTML, e separa este bloco do conteúdo circundante por quebras de linha. A tag <SPAN> pode definir um bloco de texto dentro de um parágrafo. O texto não está automaticamente separado por quebras de linha.

Eventos em DHTML

Introdução

Um evento é uma notificação que acontece em resposta a uma ação, como uma mudança de estado, um clique do mouse ou o pressionamento de uma tecla, enquanto o usuário está vendo um documento.

Um manipulador de eventos é um trecho de código, tipicamente uma função ou uma rotina escrita em linguagem script, que recebe o controle quando o evento correspondente acontece. O HTML Dinâmico fornece um conjunto de manipuladores de evento através dos

quais você pode gerar respostas para a maior parte de interações entre o usuário e o documento. O HTML Dinâmico o habilita a determinar sobre que porção do documento o manipulador de eventos irá operar.

Assim que um documento é carregado ou um usuário interage com um documento, o navegador está recebendo muitas notificações de evento. As vezes o navegador age no evento propriamente dito — por exemplo, quando você clica um hyperlink. Em outras situações, o navegador não executa nenhuma ação — por exemplo, quando você clica um texto. Processar estes eventos significa associa-los a blocos de código que você quer que sejam executados quando eles ocorrerem.

Existem três métodos para associar um evento a um bloco de código.

?? Declarar o evento na tag do elemento HTML.

?? Usar o identificador ID do elemento.

?? Declarar um bloco de código em script.

Declarando o Evento na Etiqueta do Elemento

Você pode declarar uma função para a manipulação de eventos e designar uma chamada para ela no atributo de evento apropriado de uma tag HTML.

O trecho de código exemplificado a seguir declara o procedimento **mySub** e associa sua execução a um parágrafo (elemento identificado pela tag <P>), para quando apontador do mouse passar sobre ele, ou seja, quando ocorrer o evento **onmouseover**:

```
<head>
<script language="VBScript">
      Sub mySub()
            'Execute algum código.
      End Sub
</script>
</head>
<body>
<p onmouseover="mySub">HTML Dinâmico!</p>
</body>
```

Usando o ID do Elemento

Declare uma função para manipulação de eventos e associe o manipulador de eventos da função a um determinado evento, dando a este um nome com o seguinte formato: *id_evento*.

O exemplo a seguir declara uma função escrita em VBScript denominada **myParagraph_onmouseover** e associa a função para o elemento denominado myParagraph. Quando um usuário mover o mouse sobre o parágrafo denominado myParagraph, o procedimento **myParagraph_onmouseover** será executado.

```
<html>
<head>
```

```
<script language="VBScript">
      Sub myParagraph_onmouseover()
            Msgbox "Executei algum código!"
      End Sub
</script>
</head>
<body>
<p id="myParagraph">HTML Dinâmico!</p>
</body>
</html>
```

Para declarar um bloco de script para processar eventos, execute os seguintes passos:

1. Declare código de manipulação de evento.
2. Use os atributos FOR e EVENT da tag <SCRIPT> para associar o código para a um evento.

O exemplo a seguir define um código em VBScript e o associa com o evento **onmouseover** do elemento P (parágrafo), tendo o como identificador o nome myParagraph:

```
<script for="MyParagraph" event="onmouseover" language="VBScript">
    'Execute algum código.
</script>
<p id="myParagraph">HTML Dinâmico!</p>
```

Eventos Mais Comuns

Todo conteúdo dinâmico em um documento é executado como resultado de um evento. Um evento pode ser um clique ou movimento do mouse, ou pode ser a própria carga do documento no navegador. A seguir são relacionados alguns dos eventos mais comuns que você pode utilizar para iniciar a execução de um script.

Os eventos de mouse acontecem quando o usuário move o mouse ou clica o seu botão. A tabela a seguir descreve quando cada evento de mouse acontece.

Evento de mouse	Ação do usuário que dispara o evento
onclick	Pressionar e largar o botão do mouse, ou pressionar teclas como ENTER e ESC, em um formulário.
ondblclick	Clicar duas vezes um objeto.
ondragstart	Começar a arrastar uma seleção ou selecionar um elemento.
onmousedown	Pressionar um botão em um dispositivo de apontamento, como o mouse.
onmousemove	Mover o mouse.
onmouseout	Mover o apontador do mouse para fora de um elemento.
onmouseover	Mover o apontador do mouse para um elemento. O evento acontece quando o apontador entra pela primeira vez no elemento, e não se repete a menos que o usuário mova o apontador para fora do elemento e então novamente para dentro dele.
onmouseup	Soltar o botão de mouse.

Quando um evento de mouse acontecer, suas propriedades definem o evento como a seguir:

?? A propriedade **Buttom** identifica que botão do mouse (se algum) foi pressionado.

?? As propriedades **x** e **y** especificam a localização do apontador do mouse no momento do evento.

?? Para os eventos **onmouseover** e **onmouseout**, as propriedades **toElement** e **fromElement** especificam entre que elementos o mouse está se movendo (de um para outro).

Característica de Rollover

Você pode usar os eventos **onmouseover** e **onmouseout** para animar objetos com script que é comum a estes eventos. Geralmente, você usa o evento **onmouseover** para mudar o estado do objeto e **onmouseout** para retornar ao estado original.

Use os eventos **onmouseover** e **onmouseout** do documento em conjunto com um nome de classe dada aos vários elementos, para evitar ter que escrever um manipulador de eventos separado para cada item. Isto permite a você criar tantos itens quanto desejar, para os quais será executada a mesma ação, desde que você dê o mesmo nome de classe para eles.

O exemplo a seguir usa os eventos **onmouseover** e **onmouseout** para mudar a cor de um texto específico em um documento quando o usuário passar o mouse por cima dele:

```
<html>
<head>
<title>Rollover</title>
<script language="VBScript">
      Sub document_onmouseover()
            If window.event.srcElement.className = "myH3" Then
                  window.event.srcElement.style.color = "Red"
            End if
      End Sub
      Sub document_onmouseout()
            If window.event.srcElement.className = "myH3" Then
                  window.event.srcElement.style.color = "Black"
            End if
      End Sub
</script>
</head>
<body>
<H3 class="myH3">HTML Dinâmico!</H3>
<H3 class="myH3">É Legal!</H3>
</body>
</html>
```

Movendo Elementos

Você pode utilizar os eventos **onmousedown**, **onmouseup** e **onmousemove** arrastar e mover elementos sobre o seu documento Web. Por exemplo:

```
<html>
<head>
<title>Mover Texto</title>
<script language="VBScript">
Dim DragElement
Sub test_onmousedown
      DragElement = True
End Sub
Sub document_onmousemove
      If DragElement then
            document.all.test.Style.marginleft = window.event.clientX70
            document.all.test.Style.margintop = window.event.clientY20
      End If
End Sub
Sub document_onmouseup
      DragElement = False
End Sub
</script>
</head>
<body>
<p id="test"><big><big><big><strong>Arraste
me</strong></big></big></big></p>
</body>
</html>
```

Eventos onload e onunload

Você pode utilizar os eventos onload e onunload para executar um script quanto uma página for carregada ou descarregada, respectivamente.

?? O evento **onload** ocorre depois que o documento é carregado e que todos os elementos sobre a página tiverem sido completamente descarregados pelo navegador.

?? O evento **onunload** ocorre imediatamente após o documento ser fechado.

Estes eventos operam sobre o objeto janela, mas você pode utiliza-los de duas maneiras diferentes:

?? Em VBScript, por exemplo, crie procedimentos para a manipulação dos eventos **window_onload** e **window_onunload**.

Este método é o mais usado entre os dois. O exemplo de código a seguir mostra a declaração para o procedimento **window_onload**.

```
Sub window_onload
      'execute script.
End Sub
```

?? Associar os eventos na tag <BODY>, como no exemplo a seguir:

```
<BODY onload="loadfunction" onunload="unloadfunction">
```

O Objeto Event

O **objeto evento** representa o estado de um evento, como a seguir:

- ?? O elemento em que o evento aconteceu;
- ?? O estado das teclas do teclado;
- ?? A localização do mouse;
- ?? O estado dos botões do mouse.

O **objeto evento** está disponível somente durante um evento. Isto é, você pode usa-lo em funções e rotinas manipuladoras de evento mas não em outro código. Você recupera o **objeto evento** aplicando o a palavra-chave **event** para o **objeto janela**.

O código exemplificado a seguir usa o **objeto evento** para exibir as coordenadas X e Y na janela de status:

```
Sub document_onmousemove()
      window.status = "X=" & window.event.x & " Y=" & window.event.y
End Sub
```

Embora todas propriedades de evento estejam disponíveis para todos objetos evento, algumas propriedades podem não ter valores significativos durante alguns eventos. Por exemplo, as propriedades **fromElement** e **toElement** são significativas somente quando forem processados os eventos **onmouseover** e **onmouseout**.

Nota: em VBScript, você não pode usar a palavra-chave **event** sem aplica-la para a palavra-chave **window** ou uma expressão que avalia um **objeto janela**.

Concatenação de Eventos (Bubbling)

A concatenação de eventos (*event bubbling*) assegura que os manipuladores de eventos, para todos os elementos em que um evento acontece, tem uma oportunidade para responder ao evento. Considere o exemplo a seguir:

```
<P onclick="myEvent()">
Pule para um documento <B>exemplo</B>.
</P>
```

O elemento B elemento não tem um evento associado a ele. Porém, está contido dentro do elemento P, que tem um evento associado a ele. Quando o usuário clicar a palavra "exemplo", o elemento B passa para o evento do elemento imediatamente acima, na hierarquia do documento, isto é chamado de "concatenação para cima". O elemento P pode então responder ao evento.

Se você estiver concatenando eventos em muitos elementos aninhados, provavelmente desejará evitar a concatenação dos eventos para cima na hierarquia de eventos do documento. Você pode fazer isso utilizando a propriedade **cancelBubble** do objeto evento para **True**. Quando a propriedade **cancelBubble** for configurada para **True**, o processo de concatenação de eventos é interrompido.

Os trecho de código a seguir captura eventos para o corpo do documento, do parágrafo e do tipo negrito. Uma caixa de mensagem aparece, indicando que evento foi disparado, dando ao usuário a opção de cancelar a concatenação de eventos.

```
<html>
<head>
<title>Concatenação de Eventos</title>
<script language="VBScript">

Sub MyBody_onclick()
      msgbox "O evendo do corpo do documento foi disparado", vbOkOnly
End Sub
Sub MyParagraph_onclick()
      If msgbox("O evento do parágrafo foi disparado.Cancelar a
borbulhação?",vbYesNo) = vbYes Then
            window.event.cancelBubble = True
      End If
End Sub
Sub MyBold_onclick()
      If msgbox("O evento negrito foi disparado. Cancelar a
borbulhação?",vbYesNo) = vbYes Then
            window.event.cancelBubble = True
      End If
End Sub
</script>
</head>
<body id="MyBody">

<p id="MyParagraph">Pular para o documento <b id="MyBold">exemplo</b>.
</p>
</body>
</html>
```

Usando Coleções

Uma coleção é um grupo de itens relacionados. Em HTML Dinâmico, você usa coleções para manter itens ligados como hyperlinks em um documento ou células em uma tabela. As coleções podem ter propriedades e métodos, e podem também conter outras coleções.

Toda coleção em HTML Dinâmico possui duas propriedades:

- **length** (comprimento)
- **item**

A propriedade **length** contém o número de elementos de uma coleção.

O exemplo a seguir utiliza a propriedade **length** da coleção **links** para contar o número de hyperlinks em um documento:

```
<html>
<head>
```

```
<script language="VBScript">
      sub myBody_onclick()
            msgbox "Esta página web possui " & document.links.length & "
links."
      end sub
</script>
</head>
<body id="myBody">
<p>Clique Me</p>
<p><a id="myhref1" href="http://www.fea.usp/">Home page da
FEA/USP</a></p>
<p><a id="myhref2" href="http://www.fea.usp/fia/">Página da FIA</a></p>
</body>
</html>
```

Com a propriedade **item**, você pode recuperar um elemento ou uma coleção dentro uma outra coleção utilizando:

- O número índice do elemento, ou
- O identificador (ID) do elemento.

Você usa o índice do elemento para recuperar um elemento ou coleção se:

- Você não designou um ID para um elemento particular que você quer acessar.
- Você quer enumerar a coleção com um contador.

O exemplo de código a seguir usa a propriedade **href** para recuperar informações de **href** para o primeiro hyperlink do documento:

```
msgbox "O primeiro link deste documento destinase a " &
document.links.item(0).href & "."
```

Você pode omitir a propriedade **item,** que é usada no exemplo anterior, porque **item** é a propriedade padrão. Por exemplo:

```
msgbox "O primeiro link nesta página destinase a " &
document.links(0).href & "."
```

Nota: as coleções de itens em HTML Dinâmico são iniciadas em 0 (zero). Em Visual Basic, as coleções podem ser iniciadas em 0 (zero) ou em 1 (um). A maioria das coleções mais recentes são iniciadas em 1 (um).

Se você souber o ID (identificador) do elemento que você deseja acessar, pode usar-lo em lugar do número índice. Para fazer isto, inclua o ID em citações duplas.

O exemplo a seguir recupera o **href** para o item com o ID **myhref1**:

```
msgbox "O primeiro link nesta página destinase a " &
document.links.item("myhref1").href & "."
```

Você pode omitir o método **item** que é usado no exemplo anterior porque o **item** é o método padrão. Por exemplo:

```
msgbox "O primeiro link nesta página destinase a " &
document.links("myhref1").href & "."
```

Usando a declaração **For...Next** , você pode enumerar uma coleção de elementos.

O bloco **For...Next** é executado uma vez para cada elemento na coleção. A declaração **Exit For** pode ser usada dentro de um bloco **For...Next** para finalizar o elo de repetição. Você pode usar qualquer número de declarações **Exit For** em qualquer lugar no elo de repetição. A declaração **Exit For** é freqüentemente usada para avaliar uma condição e sair o elo de repetição mais cedo, se necessário.

O exemplo de código a seguir ilustra elos de repetição através da coleção de links do objeto documento:

```
Sub myBody_onclick()
      Dim mycol
      Dim mylinks
'Todo laço através da coleção define mycol para o próximo índice na
coleção document.links. Isto é equivalente a utilizar o comando set desta
forma: Set mycol = document.links.
      For Each mycol in document.links
      Set mycol = document.links.
      For Each mycol in document.links
            mylinks = mylinks & " " & mycol.href
      Next
      Msgbox "Os links nesta página incluem " & mylinks & "."
End Sub
```

Coleção Global (all)

A coleção global **all** representa todos os elementos de um documento HTML. Cada elemento é representado como um objeto programável identificado dentro da coleção por sua localização (do topo para parte inferior) na página HTML. Você pode acessar objetos de elementos individuais usando as propriedades **index** ou **ID**. Por exemplo, se você tiver uma página com três parágrafos H1:

```
<H1 id=head1>Heading text</H1>
<H1 id=head2>Heading text</H1>
<H1 id=head3>Heading text</H1>
```

O script a seguir acessaria o terceiro parágrafo H1 por seu ID:

```
document.all.head3
```

Devido ao fato de que cada item da coleção global **all** é um objeto, você pode aplicar propriedades e métodos para estes itens. No exemplo a seguir, a propriedade **tagName** recupera o nome da tag HTML do elemento. De forma semelhante, você pode usar outras propriedades para modificar os elementos.

No código de exemplo a seguir, o script exibe a lista de todos os elementos do documento obtidos através da coleção **all** quando o documento é carregado:

```
<html>
<head>
<title>Elementos: Coleção</title>
```

```
<script language="VBScript">
Sub showElements()
      Dim tagNames
      For each thisTagName in document.all
            tagNames = tagNames & thisTagName.tagname& ", "
      Next
   msgbox("Este documento contém as tags: " & tagNames & " e só.")
End Sub
</script>
</head>
<body onload="showElements()">
<h1>Bem-vindo!</h1>
<p>Este documento é <b>muito</b> curto.</p>
</body>
</html>
```

A coleção **all** tem as seguintes características:

1. É automaticamente atualizada:

   A coleção **all** é automaticamente atualizada para refletir quaisquer mudanças feitas no documento. Por exemplo, se você recuperar a coleção e a utilizar para apagar um elemento, a coleção não incluirá mais aquele elemento. Da mesma forma, se você adicionar um elemento, a coleção incluirá o novo elemento.

2. Pode incluir elementos que não estão na página Web:

   Em alguns casos, a coleção **all** pode conter mais elementos do que os que estão realmente no arquivo do documento. Em particular, a coleção sempre contém os elementos HTML, HEAD, TITLE, BODY, ainda que estes não estejam explicitamente presentes no documento fonte. Da mesma forma, a coleção sempre contém um elemento de TBODY para cada tabela.

3. Inclui comentários e tags desconhecidas ou inválidas:

   A coleção **all** inclui também comentários (<!-- -->) e tags desconhecidas ou inválidas. O propósito é dar a você um retrato preciso do conteúdo do documento. As tags desconhecidas ou inválidas estão tipicamente mal escritas ou são tags perdidas ou desemparelhadas. Sabendo quais são e onde elas estão, fornece a você uma oportunidade para elimina-las ou substitui-las por tags válidas.

O Método Tag

O **método tag** fornece uma coleção de elementos que possuem o nome da tag especificada. Você aplica o método para uma coleção existente e provê o nome da tag que deseja recuperar. O método procura a coleção existente e retorna uma nova coleção.

Este método procura qualquer nome de tag, até nomes que não são válidos em HTML. Se ele achar um ou mais elementos tendo aquele nome, será retornada uma coleção. Se não for achado nenhum elemento tendo o nome especificado, será retornada uma coleção vazia.

Por exemplo, o script a seguir recupera uma coleção de todos os parágrafos H1 em um documento HTML:

```
document.all.tags("H1")
```

Você usa a propriedade **length** (comprimento) para determinar quantos elementos a coleção contém. A coleção está vazia se seu comprimento é zero.

O código em VBScript exemplificado a seguir:

- ?? Aplica o método **tag** para a coleção **all** para recuperar uma nova coleção contendo só os elementos TABLE existentes no documento.
- ?? Determina se a coleção está vazia, isto é, com comprimento zero.
- ?? Enumera a coleção com **FOR EACH...NEXT** para aplicar uma borda para cada tabela.

```
<html>
<head>
<title>Elementos</title>
<script language="VBScript">
Sub addBorderToTable()
      Set colTables = document.all.tags("TABLE")
      If colTables.length > 0 Then
            For Each thisTable in colTables
                  thisTable.border = 1
            Next
      End If
End Sub
</script>
</head>
<body onclick="addBorderToTable()">
<table border="0" cellpadding="0" cellspacing="0" width="40%">
  <tr>
    <td width="33%">Tabela 1 Item 1</td>
  </tr>
  <tr>
    <td width="33%">Tabela 1 Item 2</td>
  </tr>
  <tr>
    <td width="33%">Tabela 1 Item 3</td>
  </tr>
</table>
<p>Clique Aqui!</p>
<table border="0" cellpadding="0" cellspacing="0" width="40%">
  <tr>
    <td width="33%">Tabela 1 Item 1</td>
  </tr>
  <tr>
    <td width="33%">Tabela 1 Item 2</td>
  </tr>
  <tr>
    <td width="33%">Tabela 1 Item 3</td>
  </tr>
</table>
</body>
</html>
```

Manipulando Textos e Código HTML

Introdução

O HTML Dinâmico fornece propriedades e métodos para manipular textos e HTML em seu documento. Você pode usar estas propriedades e métodos para:

?? Examinar textos em um elemento ou na tag do elemento.

?? Substituir textos e elementos.

Substituindo, Excluindo e Inserindo Textos

Em qualquer elemento que pode conter texto, você pode rapidamente examinar e substitui o texto com qualquer uma das seguintes propriedades:

?? **innerTex**t

?? **outerText**

A propriedade **innerText** é uma string e especifica todo o texto contido no elemento e quaisquer outros elementos (como o I, ou elemento itálico, ou o B, ou elemento negrito) contidos naquele elemento.

Designando uma nova string para esta propriedade é possível substituir seu conteúdo anterior, inclusive quaisquer elementos. Por exemplo, você pode apagar tudo em um elemento designando uma string vazia para a propriedade.

A propriedade **outerText** substitui o texto ou HTML fora da tag, substituindo eficazmente o texto e a tag.

O exemplo a seguir demonstra a substituição de texto em um documento:

```
<html>
<head>
<title>Subistituindo Texto e HTML</title>
<script language="VBScript">
      Sub item1_onclick()
window.event.srcElement.innerText = "innerText apenas substituição de
texto."
      End Sub
      Sub item2_onclick()
window.event.srcElement.outerText = "outerText texto substituído e
remoção da tag H3."
      End Sub
</script>
</head>
<body>
<h3 id="Item1">Clique Me innerText</h3>
<h3 id="Item2">Clique Me outerText</h3>
</body>
</html>
```

O método **insertAdjacentText** especifica o texto a ser inserido num elemento em um determinado lugar. O método insere o texto como texto simples.

A sintaxe do método **insertAdjacentText** é a seguinte:

objeto.**insertAdjacentText**(*onde*, *texto*)

onde

É uma string especificando onde inserir o texto. Pode ser uma das seguintes opções:

?? **BeforeBegin**

Insere o texto imediatamente antes do elemento.

?? **AfterBegin**

Insere o texto depois do início do elemento mas antes do outro conteúdo no elemento.

?? **BeforeEnd**

Insere o texto imediatamente antes do fim do elemento mas ao final do outro conteúdo no elemento.

?? **AfterEnd**

Insere o texto logo depois do fim do elemento.

texto

Especifica o texto a ser inserido.

O exemplo a seguir insere um texto quando o documento é clicado:

```
<html>
<head>
<title>Bem-vindo</title>
<script language="VBScript">
      Sub myH3_onClick()
            document.all.myH3.insertAdjacentText "beforeEnd","Mais Texto"
      End Sub
</script>
</head>
<body>
<h3 id="myH3">Algum texto</h3>
</body>
</html>
```

Substituindo, Excluindo e Inserindo Elementos

Da mesma maneira que você pode examinar e substituir texto, você também pode examinar e substituir elementos usando as propriedades **innerHTML** e **outerHTML**. Usando estas propriedades em conjunto com innerText e outerText, você pode examinar e substituir textos e tags HTML num documento.

O exemplo a seguir demonstra a substituição de tags HTML em um documento:

```
<html>
<head>
<title>Substituindo Texto e HTML</title>
```

```
<script language="VBScript">
      Sub item1_onclick()
window.event.srcElement.innerHTML = "<i>innerHTML substituição de texto e
adição da tag itálico.</i>"
      End Sub
      Sub item2_onclick()
window.event.srcElement.outerHTML = "<i>propriedade outerHTML Texto
substituído, tag H3 removida, tag itálico inserida.</i>"
      End Sub

</script>
</head>
<body>
<h3 id="Item1">Clique Me innerHTML</h3>
<h3 id="Item2">Clique Me outerHTML</h3>
</body>
</html>
```

O método **insertAdjacentHTML** insere o texto HTML especificado no elemento no lugar determinado. Enquanto o método **insertAdjacentText** só deixa que você inserir texto, o método **insertAdjacentHTML** deixa também inserir tags. Se o texto contém tags HTML, o método analisa gramaticalmente e formata o texto à medida que ele o insere. Por exemplo, se você inserir texto HTML delimitado pela tag <STRONG>, o texto adicionado apresentado como negrito na página Web.

A sintaxe do método **insertAdjacentHTML** é a seguinte:

object.***insertAdjacentHTML***(*onde*, *HTML*)

onde

É uma string especificando onde inserir o texto HTML. Pode ser uma das seguintes opções:

- **BeforeBegin**

  Insere o texto imediatamente antes do elemento.

- **AfterBegin**

  Insere o texto depois do começo do elemento mas antes de outro conteúdo no elemento.

- **BeforeEnd**

  Insere o texto imediatamente antes do fim do elemento mas ao final do outro conteúdo no elemento.

- **AfterEnd**

  Insere o texto logo depois do fim do elemento.

HTML

É uma string especificando o texto HTML a ser inserido. A string pode ser uma combinação de tags de texto e HTML. Esta string deve estar bem formatada em HTML válido, caso contrário este método falhará.

O exemplo a seguir adiciona HTML para o texto para myH3. Usando o método **insertAdjacentHTML** você pode embutir tags HTML no texto inserido.

```
<html>
<head>
<title>Bem-vindo</title>
<script language="VBScript">
      Sub myH3_onClick()
document.all.myH3.insertAdjacentHTML "beforeEnd","<Big>-Mais Texto</Big>"
      End Sub
</script>
</head>
<body>
<h3 id="myH3">Algum texto</h3>
</body>
</html>
```

Rolagem de Elementos

O método **scrollIntoView** rola o elemento em uma visão dentro da janela, colocando-o no topo ou na parte inferior da janela. O método é útil para mostrar imediatamente o resultado de alguma ação sem exigir que o usuário tenha que rolar manualmente o documento para achar o resultado. O exemplo a seguir sublinha o conteúdo do 16° parágrafo e o rola em uma visão:

```
<html>
<head>
<title>Rola o Item 16 numa Visão</title>
<script language="VBScript">
Sub scrollToView()
      myP1.scrollIntoView(true)
      myP1.style.textDecorationUnderline = True
End Sub
</script>
</head>
<body onload="scrollToView">
<p>Item 1</p>
<p>Item 2</p>
<p>Item 3</p>
<p>Item 4</p>
<p>Item 5</p>
<p>Item 6</p>
<p>Item 7</p>
<p>Item 8</p>
<p>Item 9</p>
<p>Item 10</p>
<p>Item 11</p>
<p>Item 12</p>
<p>Item 13</p>
<p>Item 14</p>
<p>Item 15</p>
<p id = "myP1">Item 16</p>
</body>
</html>
```

Estilos Dinâmicos

Introdução

Você pode dinamicamente mudar o estilo de qualquer elemento HTML em um documento: cor, fonte, espaçamento, indentação, posição e até a visibilidade do texto. O modelo de objetos do HTML Dinâmico torna todos os elemento e atributos de HTML acessíveis, permitindo a utilização de scripts para dinamicamente ler e mudar estilos.

Folhas de estilo (também chamadas de folhas de estilo em cascata, ou *cascading style sheets* CSS) são uma nova característica suportada e disponível nos navegadores mais modernos.

De forma semelhante aos estilos do processador de textos Microsoft Word, as folhas de estilo permitem que você defina um conjunto de estilos que podem ser aplicados em documentos HTML. As folhas de estilo oferecem os seguintes benefícios para os autores páginas Web:

?? Opções de formatação adicionais:

Através do uso de folhas de estilo, mais estilos estão disponíveis, sem a necessidade para criar tags HTML adicionais.

?? Melhor controle:

Em lugar de editar muitas instâncias de uma tag para mudar uma formatação, você pode fazer a mudança para um estilo em um lugar e ver esta mudança ser refletida em todos lugares onde o estilo é usado.

?? Personalização mais fácil:

Você pode projetar seu Web Site baseado em uma folha de estilos e, tornando a folha de estilos disponível para outros autores, assegurar uma aparência consistente através de todos os documentos HTML do seu Web Site.

Implementando Folhas de Estilo

Você pode implementar folhas de estilo em três modos:

?? **Ligadas**: as definições de estilo são armazenadas em um documento separado das páginas HTML às quais ele é aplicado. Uma folha de estilo única pode ser ligada a várias páginas HTML.

?? **Embutidas**: as definições de estilo são armazenadas em um documento HTML. Uma definição de estilo é aplicada para todas as instâncias daquele estilo dentro da página HTML.

?? **Inline**: as definições de estilo são criadas para tags ou blocos únicos de tags em uma página HTML.

Folhas de Estilo Ligadas

As folhas de estilo ligadas são a forma de implementação mais poderosa de folhas de estilo. Você pode criar um documento de folha de estilo único, e ligar este documento a qualquer

número de páginas Web em seu Web Site. Para mudar um estilo, simplesmente faça a mudança no documento de folha de estilo, e ele será automaticamente refletido em todas as páginas Web que são ligadas àquela folha de estilo.

Para criar uma folha de estilo ligada execute os seguintes passos:

1. Usando o Notepad ou outro editor de textos simples, crie um arquivo que contém as definições de estilo. Uma definição de estilo consiste no nome da tag, seguidas pelas propriedades e os valores que você deseja configurar. O exemplo a seguir define os estilos H1, H2, e P (a definição de estilo para P fixa formato padrão para todo texto no documento que não possuir tags próprias):

```
H1 {font: 17pt "Arial";
   font-weight: bold;
   color: teal}
H2 {font: 13pt "Arial";
   font-weight: bold;
   color: purple}
P  {font: 10pt "Arial";
   color: white}
```

2. Salve o arquivo no servidor Web com a extensão **.CSS** (por exemplo, Meusestilos.css).

3. Ligue o arquivo .CSS a qualquer documento HTML colocando a tag <LINK> na seção de cabeçalho <HEAD> do documento HTML. Por exemplo:

```
<HEAD>
<LINK REL=STYLE TYPE="text/css"
   SRC="http://server/docs/meusestilos.css">
</HEAD>
```

Folhas de Estilo Embutidas

Quando uma folha de estilo for embutida em um documento HTML, os estilos que você define são aplicados para o documento inteiro. Você embute uma folha de estilos em um documento HTML usando as tags <STYLE> e </STYLE>. A tag <STYLE> tem um parâmetro, TYPE. Este parâmetro especifica o tipo de mídia Internet como "text/css", que permite a navegadores que não suportam este tipo ignorar a folhas de estilos.

Qualquer número de estilos pode ser definido dentro das tags <STYLE> usando o mesmo formato como o que foi usado no arquivo .CSS descrito anteriormente. Por exemplo:

```
<HEAD>
<STYLE TYPE="text/css">
      H1    {font: 17pt "Arial";
             font-weight: bold;
             color: teal}
      H2    {font: 13pt "Arial";
             font-weight: bold;
             color: purple}
      P     {font: 10pt "Arial";
             color: white}
</STYLE>
</HEAD>
```

Nota: ao utilizar estilos embutidos, você precisa prevenir suas definições de estilo de serem exibidas como texto regular em navegadores que não suportam folhas de estilos. A maioria dos navegadores que não suporta folhas de estilo ignoram as tags <STYLE> e </STYLE>, mas interpretem as definições de estilo entre as tags como texto regular. Para evitar este problema, embuta seu bloco de estilo <STYLE> e </STYLE> dentro de um comentário, como segue:

```
<STYLE TYPE="text/css">
      <!--
      H1    {font: 17pt "Arial";
            font-weight: bold;
            color: teal}
      -->
</STYLE>
```

Usando Estilos Inline

Você insere estilos inline para tags HTML usando o parâmetro de STYLE. Por exemplo, você pode configurar uma propriedade de estilo da tag <P> como exemplificado a seguir:

```
<P STYLE="margin-left: 1.0in; color: red">
Este texto aparece indentado na cor vermelha.
<P>
```

Você pode aplicar também um estilo para um bloco de tags usando a tag <SPAN>. Quaisquer outras tags que aparecerem entre as tags <SPAN> e </SPAN> conterão a formatação que você especificar. O código exemplificado a seguir aplica uma fonte de 14 pontos para o texto e para a lista que o segue:

```
<SPAN STYLE="font-size: 14pt; color: red">
```

A especificação de estilo afeta todo o conteúdo desta página até a tag </SPAN> que fecha a tag <SPAN>.

```
<UL>
<LI>List item A
<LI>List item B
</UL>
</SPAN>
```

Estilos em Cascata

Um documento HTML pode conter mais de uma implementação de folha de estilo. Em geral, as folhas de estilo definidas pelo autor das páginas sobrepõem-se sobre os valores padrão do navegador. Se você usar todos os três métodos listados anteriormente, os estilos inline terão precedência sobre o bloco de estilos embutido, que terão precedência sobre a folha de estilos ligada.

Criando Novos Estilos

Você cria uma nova classe de estilos precedendo o nome do estilo com um ponto. Estes estilos estarão disponíveis para scripts e na lista de estilos do FrontPage, na barra de ferramentas de Formato.

O exemplo a seguir usa a tag <STYLE> para definir os estilos `.vermelhao` e `.verdinho`.

```
<style>
      .vermelhao {font-size:24pt; color:red}
      .verdinho {font-size:10pt; color:green}
</style>
```

Mudança Dinâmica de Estilos

Você pode alterar o estilo de um elemento de dois modos:

1. Mudando as propriedades do objeto de estilo do elemento.
2. Mudando o nome de classe do elemento.

Se você usar folhas de estilo em seu documento HTML, poderá mudar o nome de classe do elemento para quaisquer uma das classes definidas por uma folha de estilo.

Atributos e Propriedades de Folhas de Estilo

Cada atributo de folha de estilo (.CSS) possui uma propriedade correspondente, o que permite a você examinar e alterar qualquer estilo.

Nomes de propriedades e nomes de atributos podem diferir ligeiramente, porém são os nomes de propriedades que devem ser utilizados na especificação de estilos em linguagens de script. Em particular, hífens (-) são removidos de nomes compostos, e as primeiras letras na segunda e nas palavras subseqüentes do nome são capitalizadas. Por exemplo, a cor de fundo **background-color** do atributo CSS se torna a propriedade **backgroundColor**.

Usando Propriedades de Estilo para mudar um Estilo

Você pode usar quaisquer uma das propriedades do objeto estilo para afetar diretamente um elemento em sua página Web.

O código exemplificado a seguir mostra como você pode usar um script para mudar o objeto estilo de um elemento HTML:

```
<script language="VBScript">
      Sub changeStyle()
            myh1.style.color = "red"
            myh1.style.fontsize = "48"
            myh1.style.cursor = "wait"
      End sub
</script>
<h1 id="myh1" onclick="changestyle">Meu cabeçalho</h1>
```

Usando o Atributo ClassName para mudar um Estilo

Você pode definir um estilo que fixa vários atributos e, então construir um script que define o atributo ClassName de seu elemento HTML para o nome do estilo. Por exemplo:

```
<style>
      .vermelhao {font-size:24pt; color:red}
      .verdinho {font-size:10pt; color:green}
</style>
</head>
<body>
<script language="vbscript">
sub changeColor()
  if myh1.classname = "verdinho" then
     myh1.classname = "vermelhao"
  else
     myh1.classname = "verdinho"
  end if
end sub
</script>
<h1 id="myh1" onclick="changeColor">HTML Dinâmico</h1>
</body>
</html>
```

Nota: quando você mudar um estilo, o Microsoft Internet Explorer 4.0 ou superior automaticamente refaz só as seções da página afetadas pela mudança. Por exemplo, se o tamanho da letra de um parágrafo for alterado, o texto no parágrafo será redistribuído conforme necessário. Por outro lado, se um item em uma lista for escondido, os itens subseqüentes são redistribuídos e apropriadamente renumerados para preencher o lugar do item que foi escondido.

Transições e Efeitos Visuais

Introdução

Filtros visuais são atributos de folhas de estilo em cascata (CSS) que permitem a adição de recursos visuais como sombras, brilho, iluminação e outros efeitos para:

- Textos
- Imagens
- Qualquer objeto não contido em uma janela

O Microsoft Internet Explorer versão 4.0 ou superior, suporta filtros e efeitos visuais de transição através de folhas de estilo em cascata (CSS). As transições são filtros que criam uma transposição de um estado visual até outro com uma duração especificada. Por exemplo, um estado visual no começo da transição pode ter um bloco de texto com uma fonte em cor azul. O estado visual no fim da transição pode ter a cor da fonte do texto em vermelho.

Usando Filtros Visuais

Um filtro visual é um efeito que você pode adicionar para elementos e objetos em sua página Web para mudar dinamicamente a apresentação da página. Você pode usar filtros visuais para inserir:

- ?? Animações
- ?? Iluminação
- ?? Sombras
- ?? Brilho e outros efeitos

Os efeitos de filtro visual estão disponíveis através do atributo de filtro CSS, seguido pelo nome do filtro e quaisquer parâmetros para o filtro, entre parênteses. Você pode especificar filtros através de folhas de estilo completas ou estilos inline, e então associa-los a eventos para fornecer efeitos conforme o usuário interagir com o documento.

Parâmetros de Filtro

Cada filtro pode ter um conjunto de parâmetros opcionais que definem a natureza precisa do efeito. Alguns filtros, como **FlipV** e **FlipH** para imagens espelho vertical e horizontal, não possuem nenhum parâmetro.

O código exemplificado a seguir adiciona o filtro de sombra **Shadow** para um texto, quando este for clicado. O filtro de sombra possui dois parâmetros, cor e direção.

```
<html>
<head>
<title>Sombra Simples</title>
<script language="vbscript">
      Sub myText_onClick
            myText.style.filter="shadow(color=0080FF, direction=315)"
      End Sub
</script>
</head>
<body>
<p><span id="myText" style="height:330"><big><big><big><big><big>Clique
Me</big></big></big></big></big>
</span></p>
</body>
</html>
```

Filtros Múltiplos

Você pode aplicar filtros múltiplos para um elemento ou objeto. O código exemplificado a seguir combina os filtros de sombra e obscurecimento:

```
<html>
<head>
<title>Múltiplos Filtros</title>
```

```
<script language="vbscript">
      Sub myText_onClick
myText.style.filter="shadow(color=0080FF, direction=315) blur(add=false,
direction=260, strength=8)"
      End Sub
</script>
</head>
<body>
<p><span id="myText" style="height:330"><big><big><big><big><big>Clique
Me</big></big></big></big></big>
</span></p>
</body>
</html>
```

Coleção de Filtros

Você pode acessar filtros múltiplos em um elemento ou objeto através de uma coleção de filtros. Cada filtro que você aplica para um elemento ou objeto é automaticamente adicionado a uma coleção de filtros para aquele elemento ou objeto.

Você pode acessar os filtros na coleção usando o nome ou o índice do item na coleção.

- **Índice**: a ordem dos filtros na coleção é a ordem em que os filtros foram aplicados para o objeto.
- **Nome**: o nome é automaticamente definido para o nome do filtro.

O código exemplificado a seguir usa o índice e o nome de um item para mudar os parâmetros de filtro:

```
<html>
<head>
<title>Coleção de Filtros</title>
<script language="vbscript">
      Sub myText_onClick
            'Usando o nome do elemento.
            mytext.filters.item("shadow").color=008000
'Usando o índice do elemento. Isto modificará o atributo do filtro
obscurecer que é o segundo elemento ou índice 1 desde que é baseado em 0
            mytext.filters.item(1).strength=20
      End Sub
</script>
</head>
<body>
<p><span id="myText"
style="height:330; filter:shadow(color=0080FF, direction=315)
blur(add=false, direction=260,
strength=8)"><big><big><big><big><big>Clique Me
</big></big></big></big></big> </span></p>
</body>
</html>
```

Usando Transições

As transições permitem que você adicione efeitos de animação para elementos em seus documentos. As transições podem produzir:

?? Um efeito animado entre dois elementos;

?? Um efeito de enfraquecer ou desaparecer em um elemento;

Dois filtros de transição são fornecidos através das extensões CSS:

?? **revealTrans**

Especifica efeitos múltiplos como caixa, cortinas horizontais e assim por diante.

?? **blendTrans**

Fornece um simples enfraquecer ou desaparecer com uma duração especificada.

Para aplicar transições a um elemento você deve executar os seguintes procedimentos:

1. Definir a transição de filtro a ser utilizada. Por exemplo:

```
myText.style.filter="blendTrans(Duration=1)"
```

2. Chamar o método **Apply** para aplicar a transição ao elemento. Por exemplo:

```
myText.filters.item(0).Apply
```

3. Mudar o estado do objeto, e então chamar o método **Play** para iniciar a transição do estado original para o novo. Por exemplo:

```
myText.style.color="purple"
myText.filters.item(0).Play
```

4. Verificar o evento **onfilterchange** para determinar quando a transição for completada. Por exemplo:

```
Sub myText_onfilterchange
      'Este código será executado quando a transição for
completada.
End Sub
```

O código exemplificado a seguir executa uma transição simples:

```
<html>
<head>
<title>Mudança de Cor</title>
```

```
<script language="vbscript">
      Sub myText_onClick

            'Define a cor inicial.
            myText.style.color="black"

            myText.style.filter="blendTrans(Duration=1)"

            'Aplica a transição.
            myText.filters.item(0).Apply

            'Define a nova cor.
            myText.style.color="purple"

            'Executa a transição
            myText.filters.item(0).Play

      End Sub
</script>
</head>
<body>
<p><span id="myText" style="height:330"><big><big><big><big><big>Clique
Me</big></big></big></big></big>
</span></p>
</body>
</html>
```

O código HTML a seguir é um exemplo de uma transição simples entre duas imagens em uma tag <IMG>:

```
<html>
<head>
<title>Transição Simples</title>
<script language="vbscript">
      Sub doTrans
            theImg.filters(0).Apply
            theImg.src = "circles.bmp"
            theImg.filters(0).Play
      End Sub
</script>
</head>
<body>
<p><img id="theImg" width="130" height="130" src="triangles.jpg"
style="filter:revealTrans(Duration=3.3, Transition=23)"> <br>
<input type="button" value="Iniciar transição" onclick="doTrans()"> </p>
</body>
</html>
```

Transição entre Páginas

As transições entre páginas tornam possível fornecer efeitos multimídia na carga e descarga de páginas Web. Da mesma maneira que o Microsoft PowerPoint torna possível a transições entre slides, com HTML Dinâmico você pode fornecer vários tipos de transições entre as páginas.

As transições entre páginas são as mesmas transições que você usa em elementos e objetos, exceto que a transição afeta a página Web inteira, em lugar de afetar apenas um elemento ou objeto em particular.

Você cria transições de página com a tag <META> na seção de cabeçalho <HEAD> de uma página Web especificando:

?? Quando uma transição deve acontecer: como durante as cargas da página ou à medida que ela é descarregada.

?? O tipo de transição, sua duração e outros detalhes.

A sintaxe para transições é composta de três partes:

1. Quando a transição deve ser executada.
2. A duração da transição.
3. O efeito de transição a ser utilizado.

Os dois exemplos a seguir mostram como configurar transições quando uma página for carregada ou descarregada.

A primeira tag <META> causa a transição 2, a ser executada quando o usuário carregar a página, com duração de 4 segundos. A segunda tag <META> causa execução da transição 3, com duração de 2 segundos, para ser executada quando o usuário descarregar ou deixar a página.

```
<meta http-equiv="Page-Enter"
content="RevealTrans(Duration=4.0,Transition=2);">
<meta http-equiv="Page-Exit"
content="RevealTrans(Duration=2.1,Transition=3)">
```

Nota: para estas transições funcionarem, você precisa mover-se de uma página para outra.

Posicionamento

Introdução

O navegador Microsoft Internet Explorer 4.0 ou superior suporta elementos HTML de posicionamento através das coordenadas x e y e do plano z. Com isso, você pode posicionar elementos exatamente no lugar onde deseja que eles apareçam na página. É possível colocar também elementos em diferentes planos z, para sobrepor elementos e especificar que elemento deve estar em cima de outro.

Manipulando estas coordenadas e outros estilos dinâmicos em scripts, você pode mover elementos ao redor de uma página para anima-la. A combinação de estilos e posicionamentos dinâmicos fornecem um conjunto muito rico de opções para animação.

Os atributos CSS que descrevem a posição do objeto e os elementos de objetos que estão sendo posicionados, estão disponíveis para o modelo de scripting. Através de scripts, você pode dinamicamente mudar os atributos CSS que descrevem a posição do objeto. Quando você mudar a posição de elementos na página, depois da página ser carregada, o navegador recalcula e redesenha os elementos sem precisar recarregar a página a partir do servidor Web.

Na verdade, a posição de um elemento consiste em mais do que só suas coordenas x e y de posição na janela do navegador ou em uma impressa página. Os atributos de posicionamento incluem:

?? As coordenadas x, y, e z de posição do elemento.

?? A largura e a altura da região retangular que um elemento pode ocupar.

?? As dimensões da região de recorte na região retangular que o elemento ocupa.

?? As características quando porções do elemento ultrapassam a região retangular disponível.

?? Se o elemento é ou não visível.

Posicionamento Absoluto

A definição de uma posição absoluta para um elemento coloca-o fora do fluxo padrão do documento (de cima para baixo) e o posiciona independente de como os elementos circundantes a ele obedecem o plano do HTML padrão. Se outros elementos já ocupam a posição definida, eles não afetam o elemento posicionado, ou vice-versa. Ao invés disso, todos os elementos são apresentados no mesmo lugar, causando potencialmente que texto e outros objetos sejam sobrepostos.

Você pode controlar esta sobreposição usando o atributo indexador Z-INDEX para especificar a ordem em que elementos são empilhados na mesma localização. Por definição, um elemento posicionado sempre possui um valor mais alto para Z-INDEX que seu elemento de pai.

Se você explicitamente não definir os atributos LEFT (esquerda), TOP (topo), WIDTH (largura), ou HEIGHT (altura) para um elemento posicionado absolutamente, a posição e as dimensões serão definidas para valores padrão. Por exemplo, se você não definir uma largura, a área padrão se estenderá até a extremidade direita do elemento pai. Se você não definir uma altura, a área será alta o suficiente para acomodar todo o seu conteúdo.

Você define a posição de um elemento através de uma localização absoluta de pixel com o atributo STYLE. Defina a posição para "absoluta" e as posições esquerda e de topo para valores de pixel, como exemplificado no código a seguir:

```
<html>
<head>
<title>Posicionamento Absoluto Simples</title>
</head>
```

```
<body>
<div style="position:absolute;left:150;top:150;height:100;width:100">
<p>Esta &lt;DIV&gt; tag é posicionada na página utilizando o recurso de
posicionamento absoluto</p>
</div>
</body>
</html>
```

A posição absoluta é sempre relativa para qualquer elemento pai posicionado, ou se não existe um, ao elemento BODY (corpo). Os valores para a posição esquerda (LEFT) e de topo (TOP) são relativos para o canto superior esquerdo do próximo elemento posicionado dentro da hierarquia. Por exemplo, para colocar uma imagem no canto superior esquerdo do documento, use o seguinte código HTML:

```
<IMG SRC=" WB01541.gif" STYLE="position:absolute; left:0; top:0">
```

Para ver como um elemento pai posicionado afeta a posição absoluta de um elemento, considere o exemplo a seguir. Este exemplo coloca um elemento de imagem (IMG) no canto superior esquerdo do elemento DIV, que é ele próprio posicionado na página e coloca a imagem atrás do texto:

```
<html>
<head>
<title>Posicionamento Absoluto</title>
</head>
<body>
<div style="position:relative;left:50;top:30;height:100;width:100">
<p> O texto dentro do DIV será visível, já que a imagem é positionada
atrás dele.<img src="imagem.gif"
style="position:absolute; left:0; top:0; z-index:-1" width="40"
height="41"> </p>
</div>
</body>
</html>
```

Definindo uma Posição Relativa

Ao definir o atributo POSITION da folha de estilo para *relative*, você posiciona o elemento de acordo com o fluxo natural de construção do documento HTML, mas a posição do elemento será deslocada, de acordo com os conteúdos que o precederem. Por exemplo, ao colocar um trecho de texto dentro de um parágrafo com posicionamento relativo, este trecho será posicionado relativamente ao texto do parágrafo que o precede.

O exemplo a seguir posiciona um trecho de texto em um parágrafo relativamente ao restante do texto do parágrafo:

```
<html>
<head>
<title>Exemplo de Posicionamento Relativo</title>
</head>
```

```
<body>
<p>O sobrescrito neste nome <span style="position: relative; top:
3px">xyz</span> é
&amp;quot;xyz&amp;quot;. </p>
</body>
</html>
```

O valor **– 3px** fixa o texto "xyz" três pixeis acima do alinhamento natural do texto. Você pode configurar o atributo LEFT de forma semelhante para mudar o espaçamento horizontal entre "nome" e "xyz".

Se o conteúdo do elemento SPAN estivesse posicionado de forma absoluta, "xyz" seria posicionado relativamente ao elemento BODY (ou ao próximo elemento posicionado na hierarquia) — neste caso, "xyz" seria apenas visível no canto superior esquerdo da janela do navegador do cliente.

Textos e elementos que seguem um elemento com posicionamento relativo ocupam seu próprio espaço e não sobrepõem o espaço natural usado para este elemento. Em contraste, textos e elementos que seguem um elemento com posicionamento absoluto ocupam o espaço que teria sido natural para este elemento, antes dele ser retirado do fluxo.

Elementos relativamente posicionados podem sobrepor outros objetos e elementos na página. Da mesma forma como com o posicionamento absoluto, você pode especificar o atributo Z-INDEX para definir o índice z do elemento relativamente a outros elementos que podem ocupar a mesma área. Por definição, um elemento posicionado sempre tem uma coordenada z maior que seu elemento de pai, ou seja, ele sempre estará num plano acima de seu elemento de pai.

Posicionamento e Dimensões

Existem várias propriedades disponíveis para definição da localização e dimensões de um elemento numa página Web. A possibilidade de mudar o valor destas propriedades através do script apropriado revela uma das características mais poderosas do HTML Dinâmico.

Você pode redimensionar ou posicionar um elemento na página usando as propriedades **pixelLeft**, **pixelTop**, **pixelWidth** e **pixelHeight**. Estas propriedades numéricas definem ou fornecem as coordenadas físicas e as dimensões do elemento, conforme descritas na tabela a seguir.

Propriedade	Descrição
pixelLeft	Define ou retorna a posição esquerda do elemento, em pixeis.
pixelTop	Define ou retorna a posição superior do elemento, em pixeis.
pixelWidth	Define ou retorna a largura do elemento, em pixeis.
pixelHeight	Define ou retorna a altura do elemento, em pixeis.

Nota: dependendo do elemento, antes de poder usar uma das propriedades de posicionamento, você pode ter que definir sua **propriedade de posição** para "relativa" ou "absoluta". Por exemplo, **style**="**position:relative**."

Você pode examinar a localização, largura e altura de um elemento usando as propriedades **offsetLeft**, **offsetTop**, **offsetHeight** e **offsetWidth**. Estas propriedades numéricas fornecem

as coordenadas físicas e as dimensões do elemento relativas ao seu elemento pai (do qual o elemento em questão é derivado), como descrito na tabela a seguir.

Propriedade	**Descrição**
offsetLeft	Retorna a posição esquerda calculada, em pixeis, baseada na janela.
offsetTop	Retorna a posição superior do elemento, calculada relativamente ao sistema de coordenadas do elemento pai.
offsetWidth	Retorna a largura do elemento, relativamente ao canto superior esquerdo do documento.
offsetHeight	Retorna a altura do elemento, em pixeis, calculada relativamente ao sistema de coordenadas do elemento pai.

O exemplo a seguir usa o posicionamento dinâmico para fazer um texto saltar de um lado para outro em uma página Web:

```
<html>
<head>
<title>Animando Elementos</title>
<script language="VBScript"><!--
Dim id
Dim theDirection
Sub StartBounce()
    id = window.setInterval("Bounce()",10)
End Sub
Sub Bounce()
      If theDirection=0 then
            Banner.style.pixelLeft=Banner.style.pixelLeft-10
            If Banner.style.pixelLeft<=0 Then
                  Banner.style.pixelLeft=0
                  theDirection=1
            End If
      Else
            Banner.style.pixelLeft = Banner.style.pixelLeft+10
If Banner.style.pixelLeft>=document.body.OffsetWidth-Banner.OffsetWidth-
20 Then
Banner.style.pixelLeft=document.body.OffsetWidth-Banner.OffsetWidth-20
                  theDirection=0
            End If
      End If
End Sub
--></script>
</head>
<body onload="StartBounce()">
<span id="Banner" style="position:relative">
<p>HTML Dinâmico</p>
</span>
</body>
</html>
```

Controlando a Visibilidade

Existem muitas alternativas para você controlar dinamicamente a visibilidade de um elemento. Por exemplo, você pode usar a propriedade **visibility** (visibilidade) para mostrar ou esconder um texto em uma página Web, baseando-se em uma interação com usuário. Se sua página Web contém grandes trechos de texto, os usuários podem selecionar o texto que desejam ver, enquanto o resto permanece escondido.

O trecho de código a seguir configura a propriedade **visibility** para "*visible*" (visível) e "*hidden*" (escondido) para um determinado elemento em uma página Web:

```
document.all.Banner.style.visibility = "visible"
document.all.Banner.style.visibility = "hidden"
```

Scriptlets

Introdução

Um scriptlet é uma página Web baseada em HTML Dinâmico que você pode usar como um controle ou componente em qualquer aplicativo que suporta controles. O scriptlet é uma página HTML completa (arquivo .htm), mas inclui informações que permitem que você trabalhe com ele como um controle, ou seja, você pode recuperar e fixar suas propriedades, chamar seus métodos e assim por diante, da mesma forma que você utiliza um controle ActiveX ou um componente numa linguagem orientada a objetos.

Scriptlets são pequenos, eficientes e fáceis de serem criados e mantidos, além de fornecerem recursos para:

- Autores de páginas Web criarem componentes de interface de usuário reutilizáveis, sem ter que, para isso, utilizar o poder completo de linguagens complexas como C, C++, ou outros ambientes de construção de controles.
- Desenvolvedores que utilizam o Microsoft Visual Basic, Microsoft Visual InterDev, o Borland/Inprise Delphi ou outros ambientes que suportam o desenvolvimento de controles, para usar as características contidas nos controles na construção de páginas Web.

Com scriptlets, você pode:

- Usar as capacidades gráficas e de hipertexto das páginas Web para criar uma interface visualmente rica para seu aplicativo.
- Alterar a aparência e o comportamento das páginas Web dentro de um determinado ambiente hospedeiro. Por exemplo, você pode usar um aplicativo escrito em Visual Basic para ler informações de arquivos de dados e então escreve-las em um scriptlet.
- Prototipar controles que você pretende escrever para outros ambientes. Por ser fácil e rápida a criação de um scriptlet, você pode testar suas idéias. Quando você completar seu projeto, pode implementar o controle em outro ambiente, como C++, Visual Basic ou Visual J++ ou Delphi, caso você necessite desempenho maior ou um meio diferente de empacotar (*packaging*) seu controle.

Usando Scriptlets

Você usa scriptlets como se fossem controles normais. Em um scriptlet, você cria as propriedades, métodos ou os eventos que você desejar através da construção de scripts em qualquer linguagem de scripting como **JavaScript** ou **VBScript** (*Visual Basic Scripting Edition*). Os scripts contam com os recursos do HTML Dinâmico, e fornecem a você um modelo de objetos completo para o controle de elementos no scriptlet.

Por exemplo, um scriptlet poderia ser uma página Web que contém animação baseada em HTML Dinâmico, que move e redimensiona textos de acordo com o tamanho da página. Você pode escrever scripts para expor propriedades que permitem outro aplicativo configurar o texto, a velocidade e a direção da animação, além de métodos que permitem outro aplicativo para iniciar, parar e pausar a animação.

O suporte para scriptlets é construído nos navegadores, nos quais você pode usar a tag <OBJECT> para inserir e trabalhar com scriptlets, como você faz qualquer outro objeto em uma página Web.

Para inserir scriptlets em aplicativos que suportam controles ActiveX, você utiliza um controle chamado objeto scriptlet container, que conterá o scriptlet. O aplicativo usa o objeto scriptlet container para criar uma janela para o scriptlet e fornecer um modo para o aplicativo que o contém especificar o scriptlet a ser usado, onde ele deve ser exibido, com que tamanho, e assim por diante. O objeto scriptlet container fornece também a interface para que você para configurar e definir as propriedades do scriptlet, chamar seus métodos e responder a seus eventos.

Vantagens de Scriptlets

Scriptlets fornecem vários recursos chave para provedores de conteúdo. Para reutilizar códigos escritos em HTML e scripts, os autores de conteúdo freqüentemente os cortam e colam, para então personalizar código existente. Construindo scriptlets, os desenvolvedores de conteúdo podem criar seu conteúdo uma única vez, usando HTML e scripts, e armazena-los dentro de scriplets. Com isso, outros autores rapidamente poderão reutilizar os scriptlets desenvolvidos dentro de páginas Web e aplicativos, seguindo convenções simples para expor interfaces bem definidas. Através destas interfaces, outros autores podem personalizar o conteúdo de um scriptlet existente, sem ter que entender os detalhes complexos de implementação.

A incorporação de scriptlets em páginas Web pode também melhorar o seu desempenho. Uma vez que scriptlets são compostos apenas por código HTML e scripts, eles são extremamente leves e rápidos para *download* e execução para usuários finais. Uma vez que o scriptlet for carregado para o cliente, ele é armazenado (*cached*) para uso futuro.

Criando um Scriplet Simples

Um scriptlet é uma página escrita com HTML Dinâmico que usa certas convenções para definir suas propriedades, métodos e eventos. Um desenvolvedor Web pode então usar estas propriedades, métodos e eventos para acessar recursos do scriptlet que você definiu para expor como interface. Existem várias formas para definir propriedades, métodos e eventos para scriptlets.

Uma vez que nosso objetivo é apenas fornecer uma visão geral sobre scriplets, nossa discussão será limitada a criar uma propriedade exposta, para leitura e escrita.

Uma maneira simples para criar uma propriedade que estará disponível externamente ao documento do scriptlet é através da criação de uma função com uma determinada convenção de nomenclatura, especificada a seguir:

?? **Propriedade de leitura**

- Para expor uma propriedade, o nome da função deve começar com a palavra **Public**.
- Para que ela seja uma propriedade de leitura, **Public** deve ser seguida por um sublinhado, e então a palavra **Get**.
- A última parte do nome de função é o nome da propriedade que será exposta para leitura.

```
Function Public_Get_MyProperty()
'O valor de retorno é tipicamente um script ou uma variável a nível
de documento.
      Public_Get_MyProperty=myLocalVariable
End Function
```

?? **Propriedade de Escrita**

- Para expor uma propriedade, o nome da função deve começar com a palavra **Public**.
- Para que ela seja uma propriedade de leitura, **Public** deve ser seguida por um sublinhado, e então a palavra **Put**.
- A última parte do nome de função é o nome da propriedade que será exposta.

```
Function Public_Put_MyProperty(argument)
'Tipicamente aqui é definido um script ou uma variável a nível de
documento.
      myLocalVariable=argument
End Function
```

Nota: você deve usar ambas as funções **Get** e **Put** para criar propriedades que possam ser lidas e escritas.

O código exemplificado a seguir é um documento em HTML Dinâmico projetado como um scriptlet. Ele apresenta uma mensagem definida pelo usuário em sua janela. Existem duas propriedades apenas de leitura expostas: **Message** e **Interval**. A propriedade **Message** é o texto que está apresentado, e a propriedade **Interval** controla o tempo de sua apresentação.

```
<head>
<title>Mensagem na Tela</title>
<script language="VBScript"><!--
Dim id
Dim myInterval
Dim theDirection
```

```
Sub StartBounce()
    id = window.setInterval("Bounce()",myInterval)
End Sub

Sub Bounce()
      If theDirection=0 then
            Banner.style.pixelLeft=Banner.style.pixelLeft10
            If Banner.style.pixelLeft<=0 Then
                  Banner.style.pixelLeft=0
                  theDirection=1
            End If
      Else
            Banner.style.pixelLeft = Banner.style.pixelLeft+10
If Banner.style.pixelLeft>=document.body.OffsetWidthBanner.OffsetWidth-
20 Then
Banner.style.pixelLeft=document.body.OffsetWidth-Banner.OffsetWidth-20
                  theDirection=0
            End If
      End If
End Sub

'Criação da propriedade pública apenas de escrita chamada Interval.
Function Public_Put_Interval(theInterval)
      myInterval=theInterval
      window.clearInterval(id)
      StartBounce()
End Function

'Criação da propriedade pública apenas de leitura chamada Message.
Function Public_Put_Message(theMessage)
      'Alterar o banner para refletir a nova mensagem
      Banner.innerText=theMessage
End Function
--></script>
</head>

<body onload="Public_Put_Interval(10)">
<span id="Banner" style="position:relative">

<p>Mensagem Padrão</p>
</span>
</body>
</html>
```

Adicionando Scriptles a uma Página Web

Adicionar um scriptlet em uma página Web é semelhante a inserir um controle ActiveX. A diferença principal é que você não usa a classe ID do objeto. Ao invés disso, você usa o atributo **Type** para definir o objeto. O navegador reconhecerá este tipo e inserirá o scriptlet.

?? Para inserir um scriptlet em uma página Web execute o seguinte:

1. Defina o scriptlet usando uma tag <OBJECT>, definindo seu atributo TYPE para "**text/x-scriptlet**".
2. Especifique a URL do scriptlet definindo o atributo DATA na tag <OBJECT>, como no exemplo a seguir:

```
<OBJECT ID="Scriptlet1" TYPE="text/x-scriptlet" WIDTH=300
HEIGHT=200>
   <PARAM NAME="url" VALUE="http://myserver/sample.htm"
</OBJECT>
```

O próximo exemplo usa o scriptlet definido na seção anterior. No evento **window_onload** as propriedades **Message** e **Interval** do scriplet são configuradas.

```
<html>
<head>
<title>Scriptlet Container</title>
<script language="VBScript">
Sub   window_onload
      BouncingBanner.Message = "Mensagem na Tela"
      BouncingBanner.Interval = 30
End sub
</script>
</head>
<body>
<p>
<object id="BouncingBanner" type="text/x-scriptlet" width="300"
height="40">
  <param name="url" value="Scriptlet.htm">
</object>
</p>
</body>
</html>
```

Bibliografia

Mastering Web Site Fundamentals - Microsoft Press1998.

Mastering JavaScript and Jscript – James Jaworski – Sybex 2000.