1.264 – Aula 15

Ambientes de desenvolvimento da rede:
Java Script
Java Applets
Java Servlets
Páginas ativas de servidor

Ambientes de Desenvolvimento

- XML e WSDL são documentos
- SOAP é uma extensão http
- UDDI é um diretório/registro de serviços e sites
  - Permite localizar servidores e serviços
- Applets de navegadores e de clientes podem manusear esses protocolos novos
  - Menor necessidade de download de conteúdo dinâmico
- Servidores devem gerar e receber XML, SOAP,...
  - Extensões diretas para tipos http e MIME
- Os ambientes de servidor que os gerenciam são Java (J2EE) e páginas ativas de servidor (ASP.NET)
  - Java (J2EE) e C##, C++ e Visual Basic (ASP) são linguagens de programação utilizadas por servidores
- São utilizados scripts e applets pelo lado do cliente, porém com importância relativamente menor
  - JavaScript, VBScript e Java Applets são utilizados para clientes

Java, JavaScript, Java Beans...

Rodam no servidor

Rodam no cliente

- Desenvolvidos de forma independente pela Sun (Java e Java Beans) e Netscape (JavaScript)
  - Java e JavaScript por acaso liberados para o navegador Netscape 2.0
- JavaScript, JavaScriptBeans (componentes JS):
  - Linguagem de scripts para páginas HTML, utilizando diversas etiquetas novas
  - Pode rodar em qualquer navegador (do cliente) ou do servidor
- Java:
  - Linguagem totalmente programável: programação da rede e em geral
  - Java applets: programas Java limitados que rodam no navegador (do cliente)
  - Java servlets: programas Java completos que rodam no servidor da rede
- Java Beans: componentes de linguagem de programação Java
  - Modelo de componentes com interface padrão (no servidor)
  - Paradigma PME (propriedades, métodos, eventos)
- Java Beans de empresas: componente Java orientado conforme a base de dados
  - Componentes da transação da base de dados: equilíbrio de cargas, retrocessos,
  - Componentes da sessão: manutenção de condição
- Páginas de servidor Java:
  - Forma mais simples de escrever Java servlets no servidor da Rede

Scripts Basics (Java Script, VB Script)

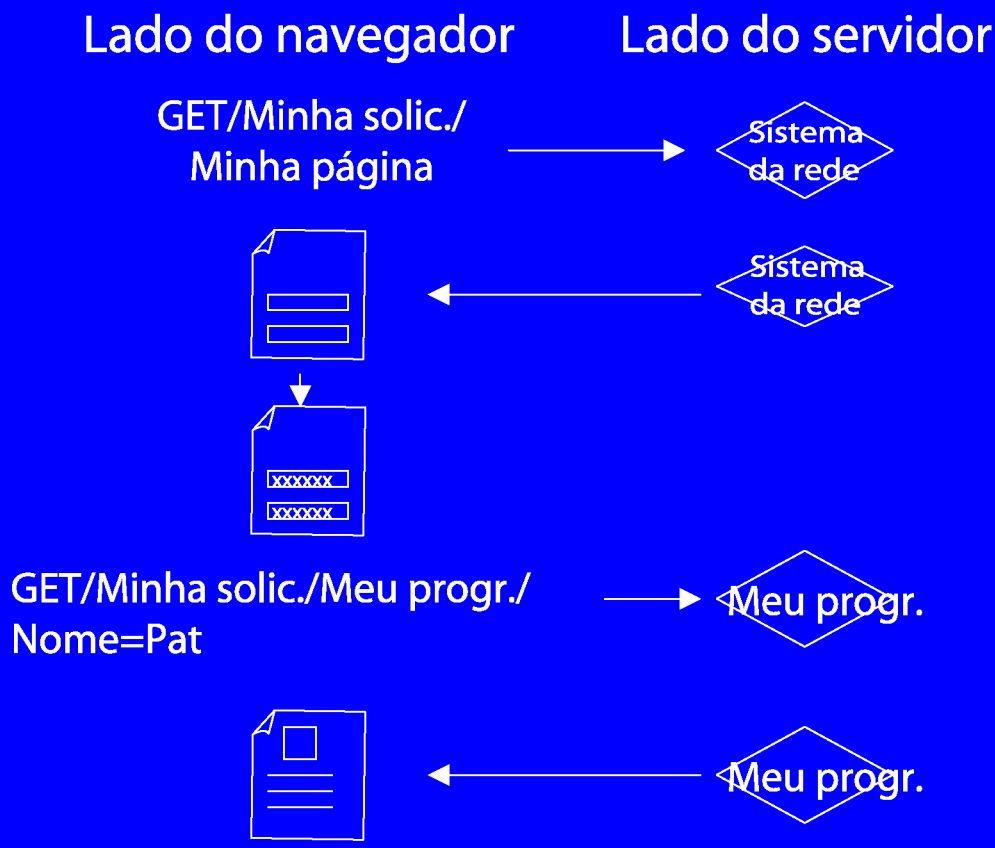
- Script é um programa que roda no cliente ou no servidor em resposta à solicitação pelo navegador
  - Scripts antigos (Interface de portal comum ou CGI) rodam somente no servidor
  - Linguagens de scripts modernos (VBScripts, JavaScripts) rodam tanto no cliente como no servidor
    - JavaScripts e VBScripts são pelo menos três vezes mais rápidos no servidor do que CGI
  - Páginas ativas de servidor especifica scripts que rodam no cliente e/ou no servidor
- Scripts da rede podem ser escritos em muitas linguagens
  - Perl e C são típicas versões do passado (e atuais) somente para servidores
  - Atualmente, JavaScript e VBScript são as maiores opções para clientes
  - "Incluído no lado do servidor" é uma capacidade muito simples do script
    - Inclui variáveis do servidor, arquivos e saídas de comandos em páginas html enviadas para o navegador. SSI roda somente no servidor.
- A primeira parte desta apresentação se concentra em scripts que rodam principalmente no navegador
  - Esses scripts de clientes são denominados frouxamente "HTML dinâmica" ou DHTML
  - DHTML's da Microsoft e Netscape são diferentes e incompatíveis, apesar de ambas suportarem JavaScript (o MS Jscript é basicamente idêntico)
  - Scripts pelo lado do servidor são utilizados da mesma forma que programas do lado do servidor, porém somente para operações simples

Motivação para scripts

- Caso programas sofisticados (Java, etc.) sejam executados somente no servidor, haverá alguns problemas:
  - Enviar e receber html do navegador, que é tratado como terminal mudo
  - Interatividade não muito boa: longa demora em circulação na exibição e obtenção de formulários com qualquer lógica ou operação interna
  - Característica econômica (ainda não solucionada): clientes médios e servidores caros e carregados
- A solução é instalar programas pequenos na máquina do cliente para execução local (apesar de não ser a solução ideal)
  - Formulários com controle integrado, animações, ativação de janelas de popups, ... (XML e Formulários X são melhores)
  - Java, JavaScripts e VBScripts são diferentes e equivocadamente denominados de linguagens que recorrem a essas características
- JavaScript atualmente é considerado inadequado para:
  - Características de segurança com carregamento de scripts desconhecidos no navegador
  - Capacidade limitada do JavaScript; pode ser substituído pelo servidor

Interação Navegador – Servidor

- **Passo 1: Navegador solicita uma página sem qualquer parâmetro**
- **Passo 2: Servidor procura a página e carrega um formulário sem entradas vazio**
- **Passo 3: Usuário preenche o formulário de entradas e pressiona a tecla "Submeter"**
- **Passo 4: Navegador compacta o formulário em forma de corrente de consulta e envia ao servidor**
- **Passo 5: Programa sintetiza um documento em Resposta e o envia de volta**



Qualquer validação dinâmica e preenchimento requerem a intervenção do servidor.

No caso de opções posteriores dependerem de opções anteriores, etc.

Pensou-se que JavaScript fosse a solução, atualmente a solução é XML, XSLT

As páginas podem ser XML ou HTML

JavaScript (ou VBScript) no navegador

- Pode ler e escrever partes de seus próprios documentos e de outros documentos que o navegador tenha aberto
- Acesso limitado a recursos de navegação (p.ex., lista de histórico, teclas de retorno e de avanço)
- Pode ler e ajustar conteúdos de formulários preenchidos e abrir novas janelas, criar novos documentos (acesso a pastas locais!)
- Biblioteca de funções matemáticas e de manipulação de textos
- Comunicações com a rede e criação de gráficos e janelas limitadas
- Você deverá utilizar Java, não o JavaScript, caso necessite ir além dos limites de um documento html ou de uma simples característica, tais como:
  - Animação, simulação, comunicações com a rede, módulos
  - JavaScript não possui módulos e se torna muito pesado 300 – 400 linhas
    - Utilize Incluído no lado do servidor para enganar uma biblioteca de JavaScript
  - O código do JavaScript não pode ser mantido privado; Java pode ser compilado

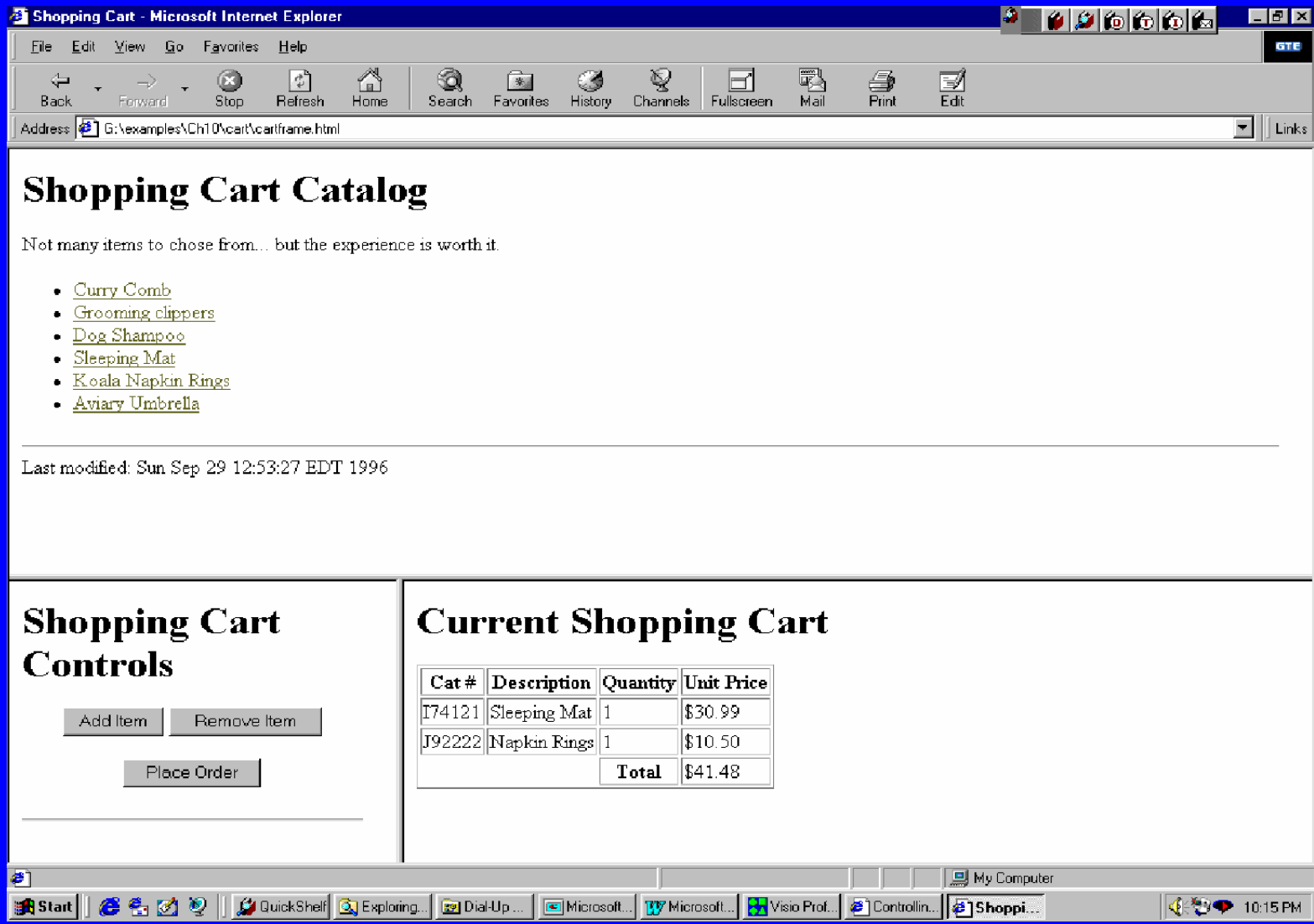
Aplicações JavaScript comuns

- Criar botões “Avançar” e “Retornar” utilizando a lista de histórico
  - É difícil saber como um usuário chega a uma página html estática sem o JavaScript
- Criar menus que selecionam URL’s
  - Economizar espaço utilizando caixas de texto ao invés de listas de textos
- Criar cabeçalhos ou marcadores de rolagem ou ativos
  - Utilizar a função alternar encadeia um caráter por vez utilizando setTimeout()
- Criar barras de navegação em pacotes
  - Possibilita o menu em um pacote para controlar a exibição em outro pacote
  - Extensão de menus que selecionam URL’s
- Repetindo, são suficientemente simples para que o XML e o XSLT possam realizar esses recursos sem carregar conteúdo ativo para o navegador a partir de um servidor

Aplicações JavaScript, continuação

- Validando formulário preenchido antes de submetê-lo
  - A rotina roda para verificar campos necessários e formatos válidos
  - Caso incorreto, exibe caixa de alerta e se recusa a submeter o formulário
- Criando listas de escolha para campos de texto utilizando menus de popups
- Criando carrinho de compras (apesar de não ser melhor forma de fazê-lo)
  - Aplicação em e-commerce para acrescentar itens selecionados a um pedido
  - Podem ser utilizados pacotes para catálogos pequenos; janelas para catálogos extensos
  - Devem ser utilizados cookies para manter a condição real da ficha de compras
    - O navegador recarrega o JavaScript a cada vez que o usuário redimensionar as janelas ou pacotes
    - Utilize armazenamento em disco local com cookies

Exemplo de carrinho de compras JavaScript



Java applets (no navegador)

- Navegadores habilitados para Java podem carregar e executar Java applets (pequenos programas Java)
- Java applets são muito mais poderosos do que o JavaScript
  - Abrem suas próprias janelas
  - Criam botões
  - Criam animações facilmente, vistas em modelos de 3D, calculadoras, etc.
  - Abrem conexões de rede para servidores domésticos, etc.
- Bibliotecas para redes, gráficos, multimídia, controles (widgets)
- Independe de plataforma: UNIX, Windows, Mac,…
- Recursos de segurança (muito melhor do que scripts ou ActiveX)
- Entretanto, os applets ainda são muito limitados pois rodam no navegador:
  - Sem acesso direto à base de dados, pequeno acesso ao hardware do cliente, menor acesso aos objetos internos do navegador que o JavaScript (em razão das preocupações com a segurança)

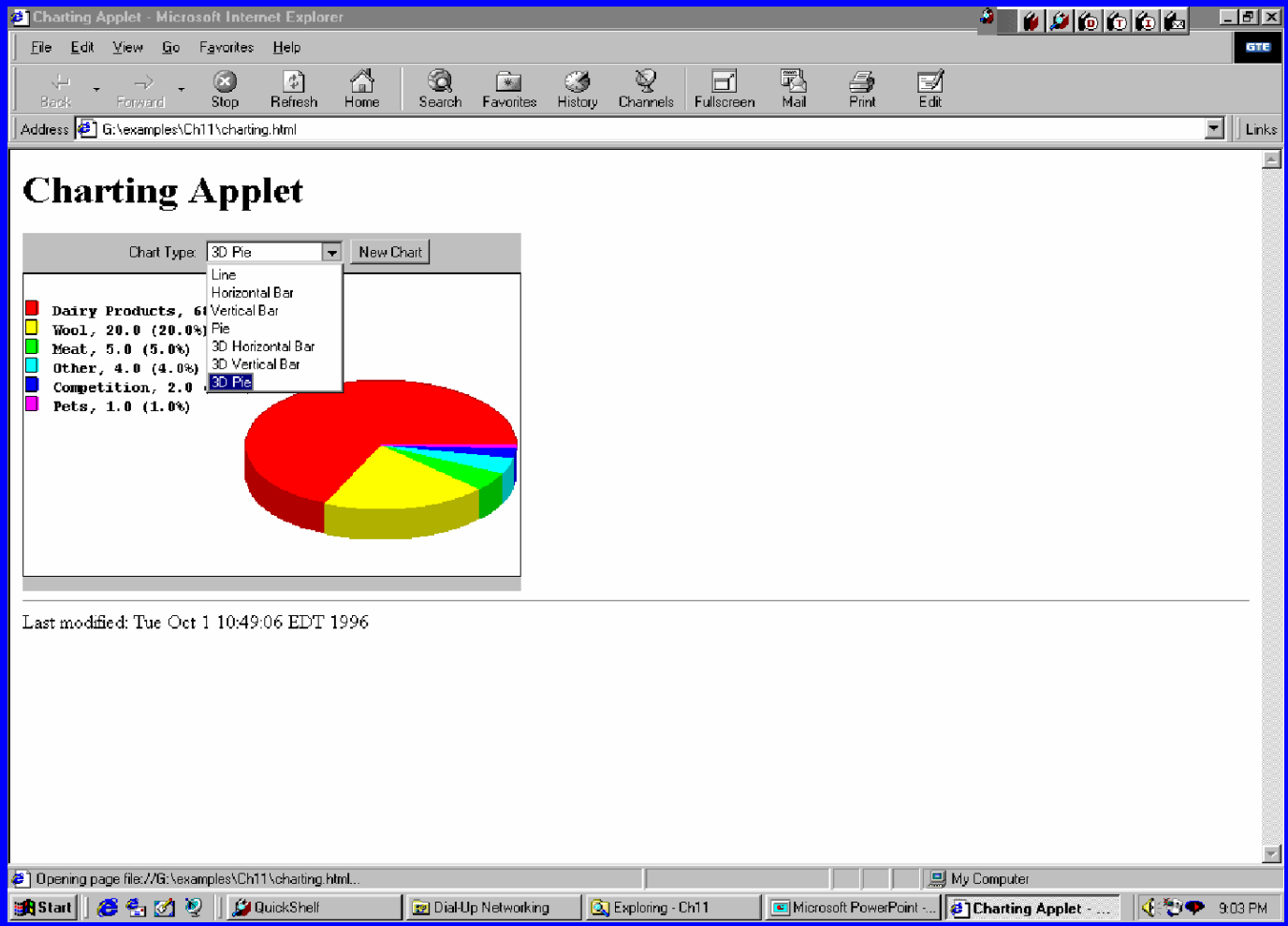
Java applets

- Applets estão incorporados a documentos HTML (como imagens)
  - <APPLET>
- O navegador examina a etiqueta para encontrar a localização do código Java
- O navegador carrega o código, cria espaço na página e o executa
  - Normalmente a partir do mesmo servidor como HTML, porém não sempre
- O Applet faz o resto:
  - Desenha textos e gráficos em seu espaço
  - Interage com o usuário com seus próprios botões
- A maioria dos applets roda na mesma janela, entretanto alguns abrem uma nova
- O applet roda até que o usuário feche a janela do navegador ou mude para outra página

Etiqueta <APPLET>

```
<APPLET
   CODE= Nome do arquivo compilado do applet (arquivo .class)
   WIDTH= Largura do applet (pixels)
   HEIGHT= Altura do applet (pixels)
   CODEBASE= URL do applet
   ALT= Alterna texto para exibição
   NAME= Nome do applet
   ALIGN= Alinhamento
   VSPACE= Espaço extra em branco acima e abaixo do applet
   HSPACE= Espaço extra em branco à esquerda e à direita do
   applet
>
<PARAM NAME="Primeiro parâmetro" VALUE="Primeiro valor">
<PARAM NAME="Segundo parâmetro" VALUE="Segundo valor">
...
Texto HTML (ignorado pelo applet; exibido por navegadores
   não-habilitados para Java)
</APPLET>
```

Java applet (roda no cliente, não no servidor)



Java applet HTML

```
<html> <head><title>Diagrama
    Applet</title></head>

<body><h1>Applet de diagrama<h1>
<APPLET CODEBASE="./applets" CODE="ChartUI"
    WIDTH= 400     HEIGTH= 300
<PARAM NAME="Tipo de localização" VALUE=URL>
<PARAM NAME="Localização"
    VALUE="Applets/goatd.txt">
<PARAM NAME="Estilo"    VALUE="3D Pie">
</APPLET>
```

Exemplos típicos atuais de Java applets

- Decoração de páginas
  - Marcadores, banners, etc.
- Animação (instalação de arquivos .gif que realizam loops)
- Botões de multimídia
- Mapas de imagens (do lado do cliente)
- Gráficos (diagramas, gráficos)
- Exibição de slides
- Aplicações reais implementadas como applets são raras...
  - O JavaScript pode agora transmitir parâmetros ao applet
  - Applets assinados obtêm mais permissões do que anônimos
  - CORBA ORB's possibilitam ainda mais permissões a applets (próxima apresentação)
- Isto ainda não se tornou muito útil de forma semelhante ao JavaScript (ou VBScript)

Java em servidores

- Java servlets. Implemente os seguintes passos:
  - Leia os dados enviados pelo navegador
    - Normalmente colocados pelo usuário, porém poderiam ser pelo applet ou pelo JavaScript
  - Examine outras informações da solicitação http
    - Capacidades do navegador, cookies, host, etc. dos cabeçalhos http
  - Gere resultados
    - Acesse a base de dados, execute o programa (possivelmente por meio de CORDA/COM), etc.
  - Formate os resultados como HTML, XML ou outro
    - Servlets podem gerar GIF, gzip e muitos tipos de MIME
  - Coloque os parâmetros de resposta http nos cabeçalhos
  - Envie o documento de volta ao navegador
- A máquina virtual Java (JVM) roda os servlets no servidor
- Estes evoluíram de applets (inúteis) para servlets (extremamente úteis, seguros, padrão)

Exemplo de servlet

```
import java.io.*;
import     javax.servlet.*;
import     javax.servlet.http.*;

public class Hello1264 extends                    HttpServlet        {
    public void         doGet(HttpServletRequest              request,
                                HttpServletResponse           response)
          throws     ServletException,          IOException       {
       response.setContentType("text/html");
       PrintWriter          out = response.getWriter();
       String      docType    =
          "<!DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.0 "                    +
          "Transitional//EN\">\n";
       out.println(docType            +
                          "<HTML>\n"     +
                          "<HEAD><TITLE>Hello 1.264</TITLE></HEAD>\n"              +
                          "<BODY>\n"     +
                          "<H1>Hello 1.264</H1>\n"          +
                          "</BODY></HTML>");}}
```

Servidor Java de páginas

- Possibilita que mixemos HTML estáticos com conteúdos gerados dinamicamente por servlets
  - Muitas páginas da rede são basicamente estáticas
  - Partes modificáveis estão somente em poucos locais da página
- Um servlet requer que geremos a página inteira de forma dinâmica a cada vez
- O JSP possibilita que o designer de rede escreve HTML estáticos e que deixe módulos referenciadores para conteúdos dinâmicos
- Programadores Java podem então preencher esses módulos
- Páginas ativas de servidores é muito parecido (lição de casa 6)
  - Você não precisará escrever nenhum código!

Exemplo JSP

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
   Transitional//EN">
<HTML> <HEAD>
<TITLE>JSP Example</TITLE>
<LINK REL=STYLESHEET
      HREF="JSP-Styles.css"
      TYPE="text/css">
</HEAD>
<BODY>
<H2>JSP Expressions</H2>
<UL>
  <LI>Current time: <%= new java.util.Date() %>
  <LI>Your hostname: <%= request.getRemoteHost() %>
  <LI>Your session ID: <%= session.getId() %>
  <LI>The <CODE>testParam</CODE> form parameter:
      <%= request.getParameter("testParam") %>
</UL>
</BODY>
</HTML>
```

Exemplo 2 JSP

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD> <TITLE>Color Testing</TITLE> </HEAD>
<%
String bgColor = request.getParameter("bgColor");
boolean hasExplicitColor;
if (bgColor != null) {
  hasExplicitColor = true;
} else {
  hasExplicitColor = false;
  bgColor = "WHITE";}
%>
```

Exemplo 2 JSP, continuação

```
<BODY BGCOLOR="<%= bgColor %>">
<H2 ALIGN="CENTER">Color Testing</H2>
<%
if (hasExplicitColor) {
  out.println("You supplied an explicit background color of " +
   bgColor + ".");
} else {
  out.println("Using default background color of WHITE. " +
              "Supply the bgColor request attribute to try " +
              "a standard color, an RRGGBB value, or to see " +
              "if your browser supports X11 color names.");
}
%>
</BODY>
</HTML>
```

Exemplo de páginas ativas de servidor ou de página de JavaScript pelo lado do servidor

```
<HTML>
   <HEAD>
   …                 'Normal HTML page sections
   </HEAD>
   <BODY>
        <% … %>      'Script that runs on server as pg created
        <SCRIPT LANGUAGE="JavaScript">
        …            'Script that runs in browser as pg interpreted
        </SCRIPT>
        <SERVER>
        …            'Script executed on server, as pg created
        </SERVER>
        <!--#include … --!>  'Server-side include, e.g., script or
                             'HTML stored in files
        <TABLE>      'Normal HTML code
        ...
        </TABLE>
   </BODY>
</HTML>
```

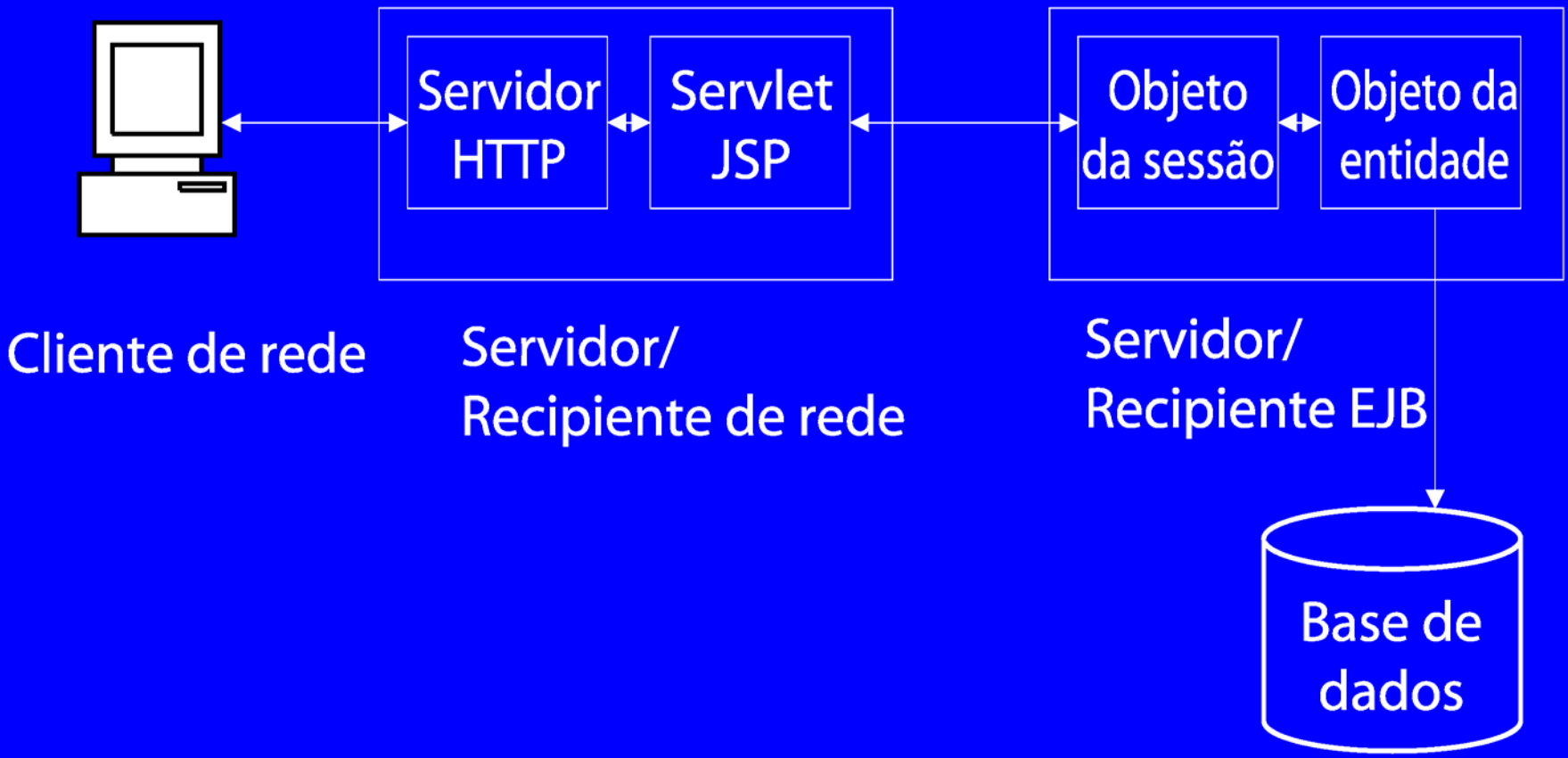
Coordenar scripts no servidor e no cliente

- Navegadores suportam VBScript, JavaScript e Java (JVM)
  - O servidor poderá executar alguns scripts ou programas em suas própria máquina e retornar outros para o navegador para execução no cliente
  - Java applets, JavaScript pelo lado do cliente e ActiveX são sempre executados no cliente:
    - Applets são transmitidos entre etiquetas <APPLET> e </APPLET>
    - Controles ActiveX são transmitidos entre etiquetas <OBJETO> e </OBJETO>
    - JavaScript é transmitido entre etiquetas <SCRIPT> e </SCRIPT>
  - Scripts a serem executados no servidor:
    - <%...%> ou <SERVIDOR> incluem scripts a serem executados no servidor
      - Este código é executado à medida que a página está sendo gerada/interpretada no servidor

JavaBeans

- Modelo de componentes para Java Servlets e páginas Java de servidor
  - Suporta componentes de aplicativo reutilizável
  - Componentes são partes pré-desenvolvidas de códigos de aplicativos que podem ser montadas e integradas em aplicativos operacionais
  - Beans são classes encapsuladas cujo código não pode ser alterado
    - Todos os dados (propriedades) possuem métodos SetXxx e GetXxx
    - Isto permite a exibição de uma caixa com propriedades visuais e coloca todos os dados
    - (Beans também necessitam de um construtor de argumento 0 e não possuem dados públicos)
  - Normalmente Beans são utilizados em páginas Java de servidores que proporcionam o melhor 'ambiente de componentes'

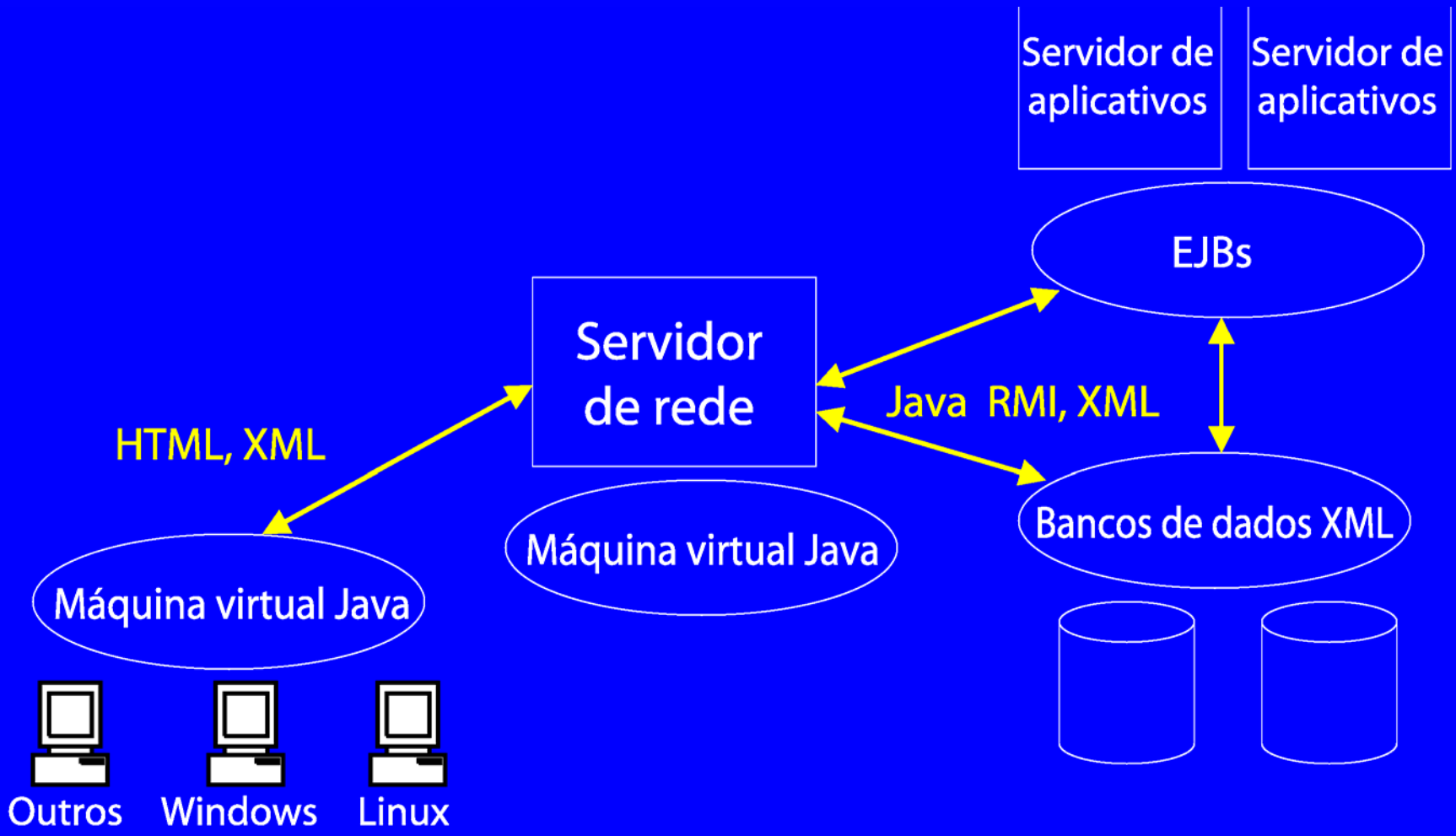
Beans Java de empresas (EJB's)



EJB's, continuação

- Beans da sessão
  - Representa o cliente (navegador, sempre um cliente único
  - Conserva a 'condição': a condição e a representação atuais da sessão (pode também ser sem condição)
  - Cria sessão bean 'just in time' quando necessário para maior eficiência e equilíbrio do carregamento
- Beans de entidade
  - Representações de dados de retaguarda: quase sempre um banco de dados
  - Persistente, representa um objeto (freqüentemente uma tabela)
  - Gerencia transações, incluindo retrocessos em caso de falhas
    - Exposição em duas etapas, exatamente como o banco de dados utiliza
- Ambos os EJB's são altamente reutilizáveis e portáteis
  - Reduzem amplamente o esforço de programação
  - Podem ser agregados e configurados de muitas formas por não-programadores

Resumo: Ambiente Java (J2EE)



Arquitetura de páginas ativas de servidor