COMP201 Java Programming

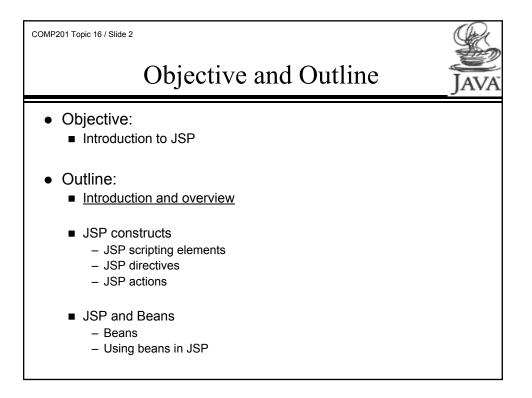
Part III: Advanced Features

Topic 16: JavaServer Pages (JSP)

Servlets and JavaServer Pages (JSP) 1.0:

A Tutorial

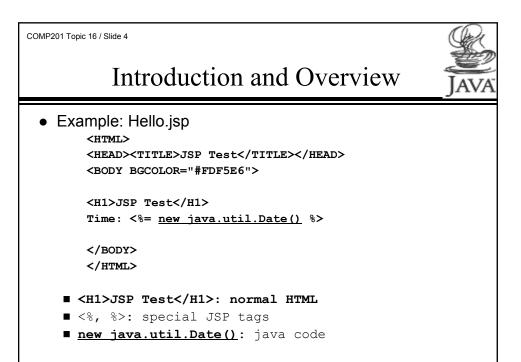
http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/Servlet-Tutorial-Intro.html





Introduction and Overview

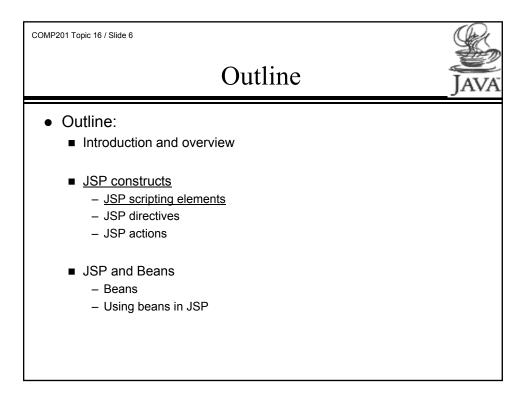
- Server-side java:
 - Scheme 1:
 - HTML files placed at location for web pages
 - Servlets placed at special location for servlets
 - Call servlets from HTML files
 - Scheme 2:
 - JSP: HTML + servlet codes + jsp tags
 - Placed at location for web pages
 - Converted to normal servlets when first accessed
 - Might take some time
 - Good idea for programmer to access it for the first time





Introduction and Overview

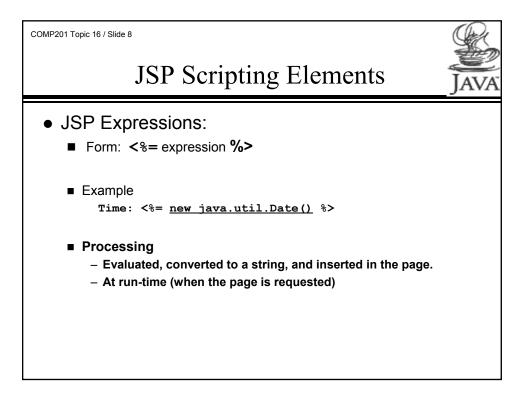
- Ingredients of a JSP
 - Regular HTML
 - Simply "passed through" to the client by the servlet created to handle the page.
 - JSP constructs
 - Scripting elements let you specify Java code that will become part of the resultant servlet,
 - Directives let you control the overall structure of the servlet, and
 - Actions let you specify existing components that should be used, and otherwise control the behavior of the JSP engine
 - JavaBeans: a type of components frequently used in JSP



JSP Scripting Elements



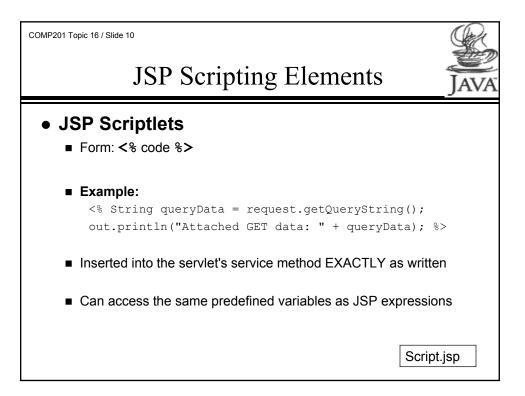
- JSP converted to Servlet at first access
- JSP scripting elements let you insert Java codes into the servlet
 - Expressions:
 - Form <%= expression %>
 - Evaluated and inserted into the output
 - Scriptlets
 - Form <% code %>
 - Inserted into the servlet's service method
 - Declarations:
 - Form <%! code %>
 - Inserted into the body





JSP Scripting Elements

- JSP Expressions:
 - Several variables predefined to simply jsp expressions
 request, the HttpServletRequest;
 - **response**, the HttpServletResponse;
 - **session**, the HttpSession associated with the request (if any);
 - out, the PrintWriter (a buffered version of type JspWriter) used to send output to the client.
- Example: Your hostname: <%= request.getRemoteHost() %>
 Script.jsp





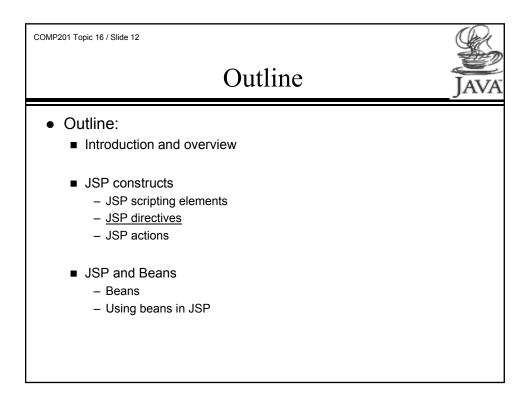
Script.jsp

JSP Scripting Elements

• JSP Declarations:

- Form: <%! code %>
- Example: <%! private int accessCount = 0; %>
- Inserted into the main body of the servlet class (outside of the service method processing the request)
- Normally used in conjunction with JSP expressions or scriptlets.

<%! private int accessCount = 0; %> Example: Accesses to page since server reboot: <%= ++accessCount %>



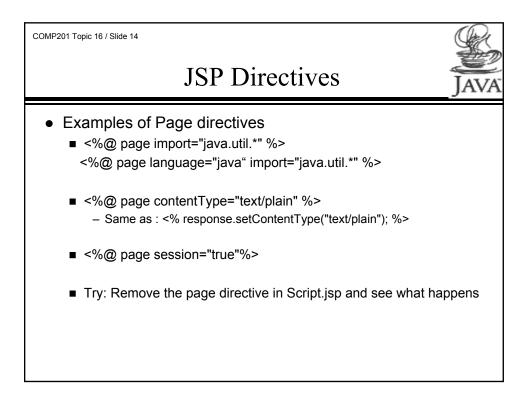
JSP Directives



- Affect the overall structure of the servlet class.
 - Form: <%@ directive attribute1="value1" attribute2="value2"

AttributeN="valueN" %>

- Two commonly used types of directives
 - Page directives <%@ page import="java.util.*" %>
 - Include directives
 <%@ include file="/navbar.html" %>

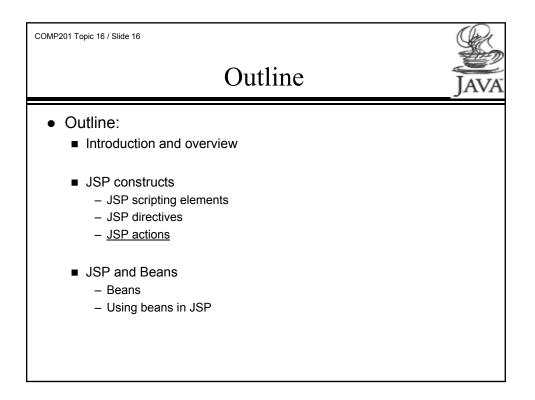


JSP Directives



• Include Directive

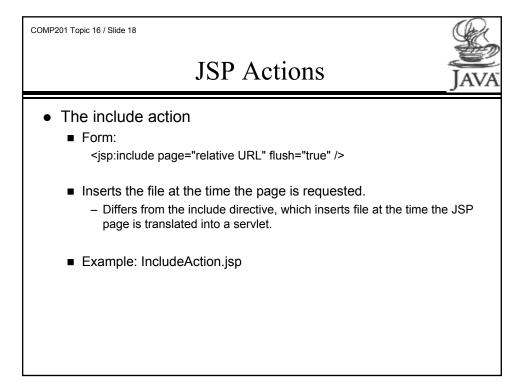
- lets you include files at the time the JSP page is translated into a servlet.
- Form: <%@ include file="relative url" %>
- Example situation where it is useful:
 - Same navigation bar at bottom of many pages.
 - Usage
 - Keep content of the navigation bar in an URL
 - Include the URL in all the pages



JSP Actions

- JAVA
- JSP actions control the behavior of the servlet engine. Let one
 - Dynamically insert a file
 - Forward the user to another page
 - Reuse JavaBeans components

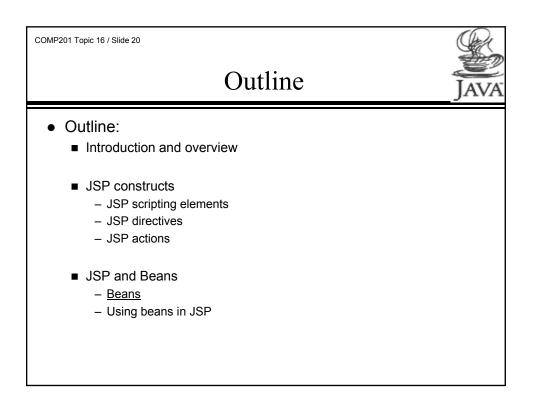
■ ..



JSP Actions



- The forward action:
 - Form: <jsp:forward page="relative URL" /> <jsp:forward page="<%= someJavaExpression %>" />
 - Forward to the page specified.
 - Example: ForwardAction.jsp
- Several actions related to reuse of JavaBeans components
 - Discuss next

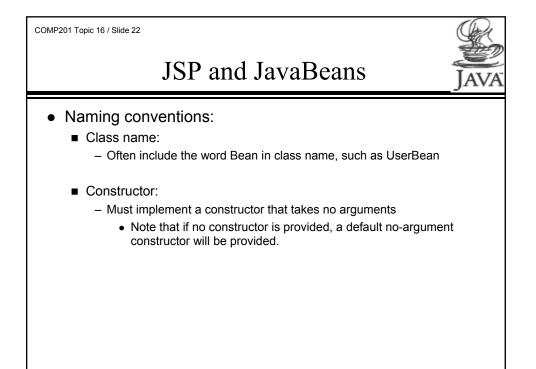


JSP and JavaBeans



• Beans

- Objects of Java classes that follow a set of simple naming and design conventions
 - Outlined by the JavaBeans specification
- Beans are Java objects
 - Other classes can access them and their methods
 - One can access them from jsp using scripting elements.
- Beans are special Java objects
 - Can be accessed using JSP actions.
 - Can be manipulated in a builder tool
 - Why interesting?
 - Programmers provide beans and documentations
 - Users do not have to know Java well to use the beans.



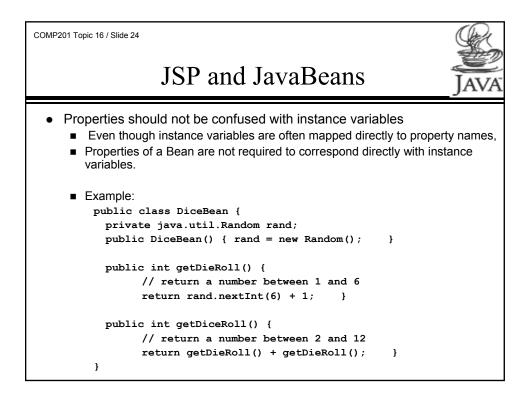
JAVA

Naming conventions: Methods

- Semantically, a bean consists of a collection of properties
- The signature for property access (getter and setter) methods public void setPropertyName(PropertyType value); public PropertyType getProperty

JSP and JavaBeans

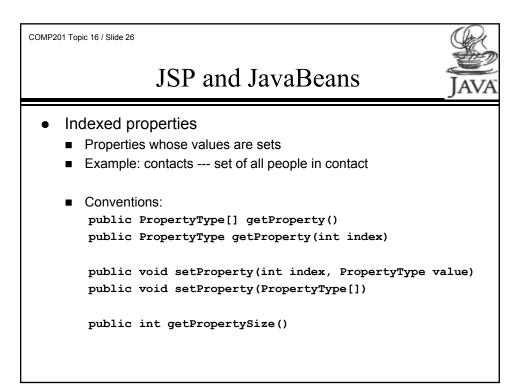
- Example: - Property called rank: public void setRank(String rank); public String getRank();
 - Property called age:
 public void setAge(int age);
 public int getAge();



JSP and JavaBeans



- Property name conventions
 - Begin with a lowercase letter
 - Uppercase the first letter of each word, except the first one, in the property name.
 - Examples: firstName, lastName
- Corresponding setter and getter methods:
 - setFirstName, setLastName
 - getFirstName, getLastName
 - Note the case difference between the property names and their access method

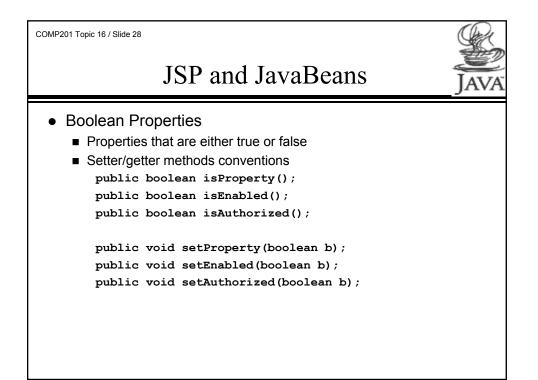




JSP and JavaBeans

```
    Bean with indexed properties

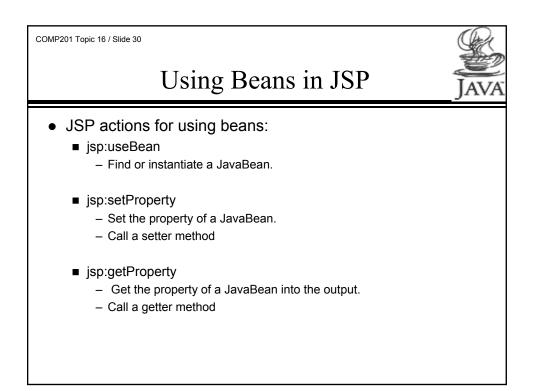
   import java.util.*;
   public class StatBean {
      private double[] numbers;
     public StatBean() {numbers = new double[0];
                                                       ł
     public double getAverage() {..}
      public double getSum() { .. }
     public double[] getNumbers()
      ſ
               return numbers;
                                  }
     public double getNumbers(int index)
              return numbers[index];
      ſ
                                          ł
     public void <u>setNumbers</u>(double[] numbers)
              this.numbers = numbers;
      {
                                          ł
      public void setNumbers(int index, double value)
             numbers[index] = value;
                                          ł
      ſ
      public int getNumbersSize()
              return numbers.length;
                                         }
      ł
   }
```



Outline



- Outline:
 - Introduction and overview
 - JSP constructs
 - JSP scripting elements
 - JSP directives
 - JSP actions
 - JSP and Beans
 - Beans
 - Using beans in JSP



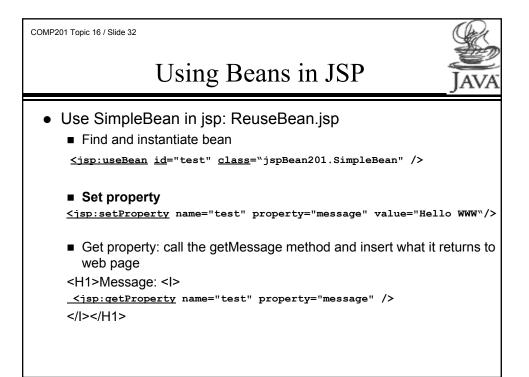
Using Beans in JSP



Example: The bean
package jspBean201;
public class SimpleBean {
 private String message = "No message specified";
 public String getMessage() {
 return(message);
 }
 public void setMessage(String message) {
 this.message = message;
 }
 }

Compile with javac and place in regular classpath

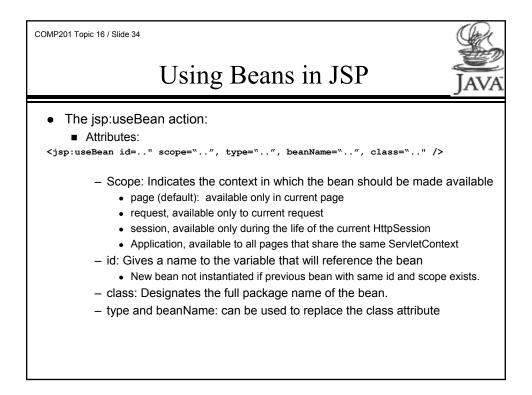
In Tomcat, same location as servlets. (can be different on other web servers)



Using Beans in JSP



- The jsp:useBean action:
 - Format
 - Simple format: <jsp:useBean .../>
 - Container format: body portion executed only when bean first instantiated <jsp:useBean ...> Body
 - </jsp:useBean>



Using Beans in JSP

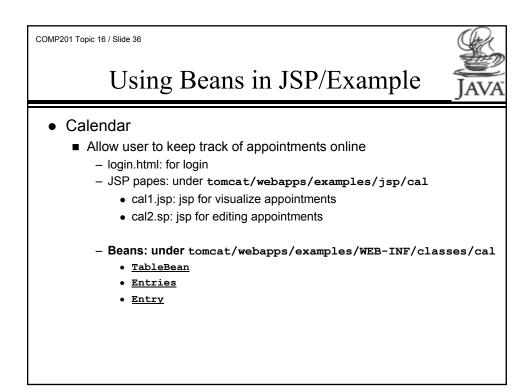


• The jsp:setProperty action:

Forms:

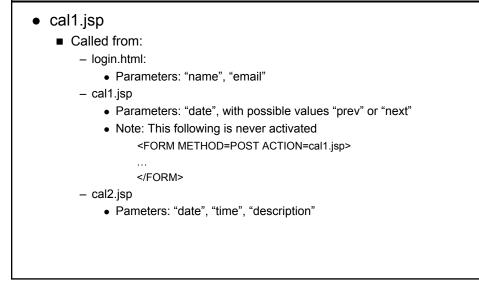
<jsp:setProperty name=".." property=".." value=".." />
<jsp:setProperty name=".." property=".." param=".." />

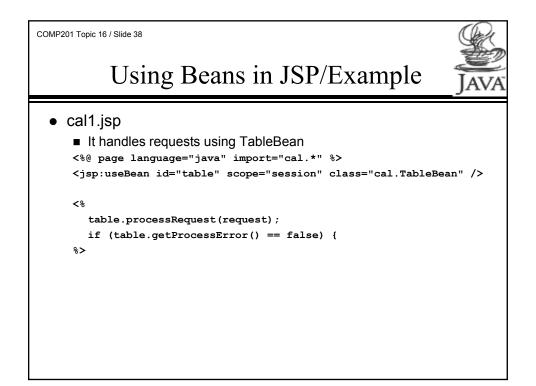
- If the value attribute is used
 - String values are automatically converted to numbers, boolean, Boolean, byte, Byte, char, and Character
- If the param attribute is used
 - No conversion





Using Beans in JSP/Example

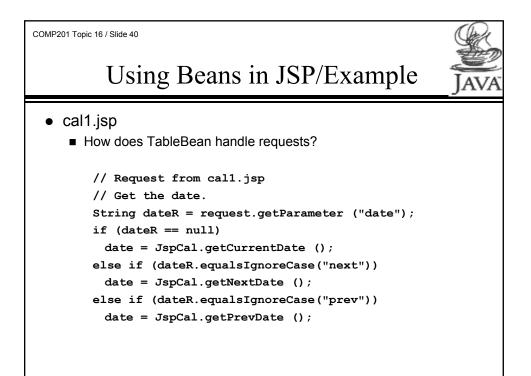






Using Beans in JSP/Example

• cal1.jsp • How does TableBean handle requests? public void processRequest (HttpServletRequest request) { // Request from login.html // Get the name and e-mail. this.processError = false; if (name == null) setName(request.getParameter ("name")); if (email == null) setEmail(request.getParameter ("email")); if (name == null || email == null) { this.processError = true; return; }





Using Beans in JSP/Example

• cal1.jsp

