Problem with Scripting Elements

- Mixing presentation and processing
  - hard to debug
  - hard to maintain
- No clean and easy way to reuse code
- Solution — separate out Java code
  - Servlets
  - Java Beans
  - Custom tag libraries

Java Beans

- A regular Java object, typically used for modeling data, e.g., `GuestBookEntry`
- A.K.A. POJO (Plain Old Java Object)

Bean Properties

- The properties of a bean are defined by getters and setters
- Properties != Class variables

```java
public class User {
    String firstname;
    String lastname;

    public String getFirstname() { return firstname; }
    public String getLastname() { return lastname; }
    public String getName() { return firstname + " " + lastname; }
}
```

About Bean Properties

- Property naming conventions
  - 1st letter is always in lower case
  - 1st letter must be capitalized in getter (accessor) and/or setter (mutator)
- Property types
  - read-only property: only getter
  - write-only property: only setter
  - read/write property: both

```java
public class Foobar {
    private int a, b, c, d;
    private boolean e;

    public int getA() { return a; }
    public void setA(int a) { this.a = a; }
    public boolean isE() { return e; }
    public void setE(boolean e) { this.e = e; }
}
```
Common Problems with Bean Property ...

```java
public class Foobar {
    private int a, b, c, d;
    public Foobar() { a = b = c = d = 0; }
    public void setD( String s ) { this.d = Integer.parseInt(s); }
    public void setC( int c, int x ) { this.c = c + x; }
    public int getB( int x ) { return b + x; }
}
```

How many properties does Foobar have?

Bean and JSP

Bean Tags and Attributes

```html
<jsp:useBean>
    <jsp:getProperty>
    <jsp:setProperty>
</jsp:useBean>
```

Example: BGColor.jsp

◆ Use a bean BGBean to control the background color of a JSP page

```html
<jsp:useBean id="bg" class="cs320.bean.BGBean" />
<jsp:getProperty name="bg" property="r" />
<jsp:setProperty name="bg" property="r" value="255" />
<jsp:setProperty name="bg" property="r" param="r" />
<jsp:setProperty name="bg" property="r" />
<jsp:setProperty name="bg" property="*" />
```

Example: Counters.jsp

◆ Use Counter bean
◆ How do we increment the counter??
◆ Understand scopes
   • Application
   • Session
   • Request
   • Page
Scopes and Data Sharing

- Application scope – data is valid throughout the life cycle of the web application
- Session scope – data is valid throughout the session
  - redirect, multiple separate requests
- Request scope – data is valid throughout the processing of the request
  - forward
- Page scope – data is valid within current page

Access Scoped Variables in Servlet

- Application scope
  - ServletContext
- Session scope
  - HttpSession
- Request scope
  - HttpServletRequest
- Page scope (in JSP scriptlet)
  - pageContext

Need for EL

- Using `<jsp:getProperty>` to access bean properties is tedious

What is EL?

- Expression Language (EL)
  - Part of the JSP 2.0 Specification
  - A more concise way to access bean properties and write JSP expressions
    - vs. `<jsp:getProperty>`
    - vs. `<%= expression %>`
  - Java’s answer to scripting languages
- Syntax: `${ expression }`

Example: BGColor.jsp Revisited

- Use EL to access the bean properties
  - `${ bean_name.property_name }`

Expression

- Literals
- Operators
- Variables
EL Literals

- true, false
- 23, 0x10, ...
- 7.5, 1.1e13, ...
- "double-quoted", 'single-quoted'
- null

- No char type

EL Operators

- Arithmetic
  - +, -, *, /, %
  - div, mod
- Logical
  - &&, ||, !
  - and, or, not
- Relational
  - ==, !=, <, >, <=, >=
  - eq, ne, lt, gt, le, ge
- Conditional
  - ?:
- empty
  - check whether a value is null or empty
- Other
  - [], \[

EL Evaluation and Auto Type Conversion

<table>
<thead>
<tr>
<th>Expression</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(2+4)/2$</td>
<td>$(empty &quot;&quot;)</td>
</tr>
<tr>
<td>$(2+3)/2$</td>
<td>$(empty param.a)</td>
</tr>
<tr>
<td>$(2+3)/2$</td>
<td>$(empty null)</td>
</tr>
<tr>
<td>$(2*3 \div 2)$</td>
<td>$(empty &quot;null&quot;)</td>
</tr>
<tr>
<td>$(a = 3 \div 2)$</td>
<td>$(&quot;abc&quot; \lt &quot;b&quot;)</td>
</tr>
<tr>
<td>$(null \eq &quot;test&quot;)</td>
<td>$(&quot;ca200&quot; \gt &quot;ca203&quot;)</td>
</tr>
<tr>
<td>$(null eq &quot;null&quot;)</td>
<td>$(empty null)</td>
</tr>
</tbody>
</table>

EL Variables

- You cannot declare new variables using EL (after all, it’s called "expression" language).
- However, you can access beans, implicit objects, and previously defined scoped variables.

Implicit Objects in EL

- pageContext
  - servletContext
  - session
  - request
  - response
- param, paramValues
- header, headerValues
- cookie
- initParam
- pageScope
  - requestScope
- sessionScope
- applicationScope

Example: RequestInfo.jsp

- Display some information about the request
  - Client address ...
  - Cookies and parameters
- Use of implicit objects
  - Find the Java class type for the object
  - Look for getters in the API
    - E.g. ${request.remoteAddr}
  - Access elements in a collection
    - cookie and param
Limitation of EL

��Only expressions, no statements, especially no control-flow statements

JSTL