Problems 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
- Bean
- Custom taglib

Java Beans

- Initially designed for GUI builders
- Beans support
  - Introspection
  - Customization
  - Events
  - Properties
  - Persistence

Java Beans for Dummies

- A zero-argument constructor
- No public class variables
- Properties
  - Properties are defined by getter and/or setters, e.g. getFoo() and setFoo()
  - Properties != Class variables

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

Bean Example

- BGBean.java
  - package!
Bean Tags and Attributes

- `<jsp:useBean>
  - class
  - id
  - scope
    - page (default)
    - request
    - session
    - application
- `<jsp:getProperty`
- `<jsp:setProperty`
  - name
  - property
  - value
  - param

Bean in JSP Example

- `BGColor2.jsp`

```xml
<jsp:useBean id="bg" class="cs320.stu31.BGBean" />
<jsp:setProperty name="bg" property="r" value="255" />
<jsp:setProperty name="bg" property="r" param="r" />
```

More About `<jsp:setProperty>`

- Note the auto-conversion from parameter string to property type
  - e.g. "255" → 255
- What if auto-conversion fails??
- What if a parameter is not specified??

A Few More Ways to setProperty

```xml
<%-- Use JSP expression for the value attribute --%>
<jsp:setProperty name="fb" property="fname"
    value="<%= request.getParameter("fname") %>"/>

<%-- Use <jsp:attribute> for value attribute --%>
<jsp:setProperty name="fb" property="age">
    <jsp:attribute name="value" trim="true">
        <%= request.getParameter("age") %>
    </jsp:attribute>
</jsp:setProperty>

<%-- Set all properties --%>
<jsp:setProperty name="fb" property="*"/>
```

Example: Form Bean

- Set all properties
- And an easier way to "get property":

  ```
  First name: ${fb.fname}
  Last name: ${fb.lname}
  ```

Scopes and Data Sharing

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