



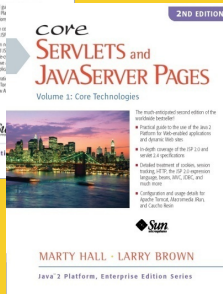
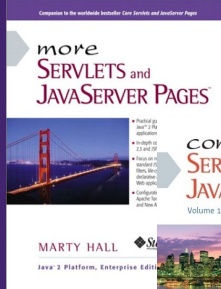
JSF: The "h" Library

Originals of Slides and Source Code for Examples:
<http://www.coreservlets.com/JSF-Tutorial/>

This somewhat old tutorial covers JSF 1, and is left online for those maintaining existing projects. All new projects should use JSF 2, which is both simpler and more powerful. See <http://www.coreservlets.com/JSF-Tutorial/jsf2/>.

Customized Java EE Training: <http://courses.coreservlets.com/>

Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.



For live training on JSF 1 or 2, please see courses at <http://courses.coreservlets.com/>.



Taught by the author of *Core Servlets and JSP*, *More Servlets and JSP*, and this tutorial. Available at public venues, or customized versions can be held on-site at your organization.

- Courses developed and taught by Marty Hall
 - JSF 2, PrimeFaces, servlets/JSP, Ajax, jQuery, Android development, Java 6 or 7 programming, custom mix of topics
 - Ajax courses can concentrate on 1 library (jQuery, Prototype/Scriptaculous, Ext-JS, Dojo, etc.) or survey several
 - Courses developed and taught by coreservlets.com experts (edited by Marty)
 - Spring, Hibernate/JPA, EJB3, GWT, Hadoop, SOAP-based and RESTful Web Services
- Contact hall@coreservlets.com for details**

Topics in This Section

- Most common usage
- Shared attributes
- Implied attributes
- Elements that can invoke action controllers/listeners
- Elements that can invoke value change listeners
- Elements that display lists of items

5

Documentation on h: Library

- Javadoc extensions can now generate online API for JSP tag libraries
 - http://java.sun.com/j2ee/javaserverfaces/1.1_01/docs/tlddocs/



6



Form Basics

Customized Java EE Training: <http://courses.coreservlets.com/>

Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Most Common Elements

- **h:form**
 - No ACTION specified (it is current page automatically)
 - You *must* use POST
- **h:inputText**
 - NAME generated automatically
 - VALUE of form `#{beanName.propertyName}`
- **h:inputSecret**
 - NAME generated automatically
 - VALUE applies only to output, not input
- **h:commandButton**
 - `action` here roughly corresponds to form ACTION
- **h:outputText**
 - Outputs bean properties.

Shared and Expected Attributes

- **Shared attributes**
 - These attributes are available in almost all h:blah elements
 - Will not be repeated in description of each element
- **Most common shared attributes**
 - **id**. Identifier; used to give name to element so it can be referred to later (e.g., from validation message). Often omitted.
 - See validator section
 - **onclick**, **ondblclick**, etc. JavaScript event handlers.
 - Note that case matters, so `onClick` is *not* OK.
 - **styleClass**. CSS style name.
- **Expected attributes**
 - These attributes are the ones that the corresponding HTML element has. E.g. `size` and `maxlength` for textfields
 - Will be omitted in description of each element

9

h:form

- **Corresponding HTML element**
 - `<FORM ...>`
- **Notable features**
 - You do not specify an action or a method
- **Attributes**
 - **enctype**. Encoding type.
 - `application/x-www-form-urlencoded` (default)
 - `multipart/form-data`
 - **target**. Frame cell in which to display results.
 - **onsubmit**. JavaScript code to run before form submission
 - Can perform client-side field validation
- **Examples**
 - See all previous sections

10

h:inputText

- **Corresponding HTML element**
 - `<INPUT TYPE="TEXT" ...>`
- **Attributes**
 - **value**. Corresponding bean value.
 - `value="#{beanName.propertyName}"`
 - Used for both initial value and submitted value
 - **valueChangeListener**. Method to fire when form submitted and content has changed from previous value.
 - Use `onchange="submit()"` and `immediate` with this
 - **onchange**. JavaScript to run when value changes
 - **immediate**. Use to indicate that listener bypasses validation
 - **required**. Flag indicating if user must supply a value.
 - If omitted, form redisplayed and error message stored.
 - **validator**. Method to perform validation.
 - See validation section
- **Examples**
 - See all previous sections

11

h:inputSecret

- **Corresponding HTML element**
 - `<INPUT TYPE="PASSWORD" ...>`
- **Attributes**
 - Same as `h:inputText`, except that `value` applies to submission only
 - Cannot prepopulate password fields
- **Example**
 - See all previous sections

Enter Password:

12



Elements That Can Invoke Action Controllers and Action Listeners

Customized Java EE Training: <http://courses.coreservlets.com/>

Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android. Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

h:commandButton (with value)

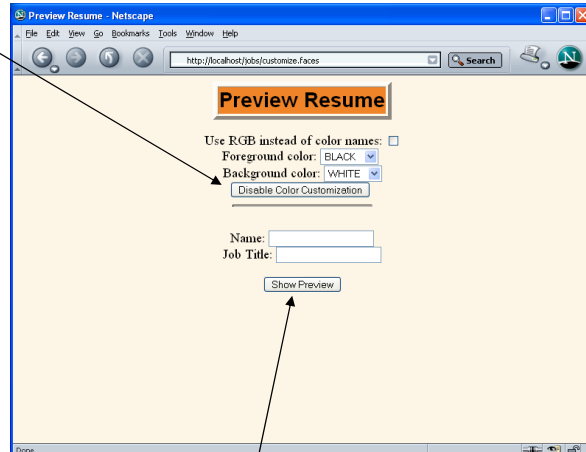
- **Corresponding HTML element**
 - `<INPUT TYPE="SUBMIT" ...>`
- **Attributes**
 - **value**. Button label. A static string is common, but you can also change value dynamically
 - **action**. Action controller method to fire when form submitted. Runs after all listeners have completed.
 - **actionListener**. Listener method to fire when form submitted.
 - Runs method and then redisplay form. No business logic, return value, or navigation rules
 - **immediate**. Indicates that setter methods and validation is bypassed
 - Usually used with actionListener
- **Examples**
 - See next page

h:commandButton (with value) Examples

```
<h:commandButton  
  value="#{resumeBean.colorSupportLabel}"  
  actionListener="#{resumeBean.toggleColorSupport}"  
  immediate="true"/>
```

- Label changes from "Disable Color..." to "Enable Color..." depending on getColorSupportLabel
- Invokes action listener when pressed (runs method and then redisplay form)
- No setter methods or validation

- Invokes action controller when pressed (runs method which returns string that matches from-outcome in navigation rule). Setter methods and validation run first.



```
<h:commandButton value="Show Preview"  
  action="#{resumeBean.showPreview}"/>
```

15

h:commandButton (with image)

- **Corresponding HTML element**
 - `<INPUT TYPE="IMAGE" ...>`
 - Server-side image map
- **Attributes**
 - Same as on previous page, but **actionListener** required if you want **x/y locations**
 - Only listener can access the client ID
 - Client ID needed to obtain x and y locations
 - Locations of the form *ClientID.x* and *ClientID.y*
 - See event handling section for examples

16

h:commandLink

- **Corresponding HTML element**
 - `` with attached JavaScript that submits form when link is pressed
 - JavaScript inserted automatically by JSF
- **Attributes**
 - **value**. Text of the link. A static string is common, but you can also change value dynamically
 - **action**. Action controller method to fire when form submitted.
 - **actionListener**. Listener method to fire when form submitted.
- **Note**
 - Must go inside `h:form` (error in earlier MyFaces releases)

17

Examples: h:commandButton and h:commandLink (Bean)

```
public class RegistrationBean implements Serializable {
    private String email = "";
    private String password = "";
    private String method="";
    // getEmail, setEmail, getPassword, setPassword
    // getMethod, setMethod

    public String register1() {
        setMethod("<CODE>commandButton</CODE> with <CODE>value</CODE>");
        return("success");
    }

    public String register2() {
        setMethod("<CODE>commandButton</CODE> with <CODE>image</CODE>");
        return("success");
    }

    public String register3() {
        setMethod("<CODE>commandLink</CODE>");
        return("success");
    }
}
```

18

h:commandButton and h:commandLink (Form)

```
<h:form>
  Email address:
  <h:inputText value="#{registrationBean.email}"/><BR>
  Password:
  <h:inputSecret value="#{registrationBean.password}"/>
  <P>
  <h:commandButton value="Sign Me Up!"
                    action="#{registrationBean.register1}"/>
  <P>
  <h:commandButton image="sign-me-up.jpg"
                    action="#{registrationBean.register2}"/>
  <P>
  <h:commandLink value="Sign Me Up!"
                 action="#{registrationBean.register3}"/>
</h:form>
```

19

h:commandButton and h:commandLink (Form Result)

- h:commandButton with "value"
- h:command button with "image"
- h:commandLink



20

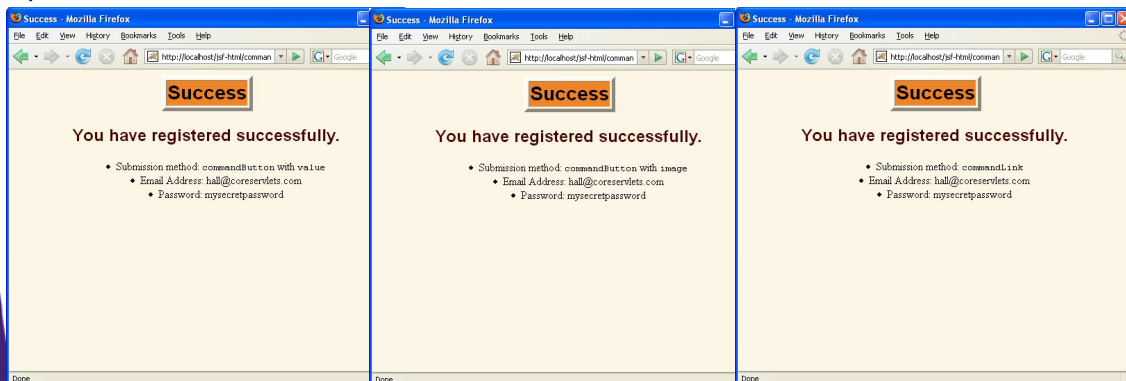
h:commandButton and h:commandLink (faces-config.xml)

```
<faces-config>
  <managed-bean>
    <managed-bean-name>registrationBean</managed-bean-name>
    <managed-bean-class>
      coreservlets.RegistrationBean
    </managed-bean-class>
    <managed-bean-scope>request</managed-bean-scope>
  </managed-bean>
  <navigation-rule>
    <from-view-id>/command-buttons.jsp</from-view-id>
    <navigation-case>
      <from-outcome>success</from-outcome>
      <to-view-id>/WEB-INF/results/success.jsp</to-view-id>
    </navigation-case>
  </navigation-rule>
</faces-config>
```

21

h:commandButton and h:commandLink (Results Page)

```
<UL>
  <LI>Submission method:
    <h:outputText value="#{registrationBean.method}"
      escape="false"/>
  <LI>Email Address:
    <h:outputText value="#{registrationBean.email}"/>
  <LI>Password:
    <h:outputText value="#{registrationBean.password}"/>
</UL>
```



22



Elements That Can Invoke Value Change Listeners

Customized Java EE Training: <http://courses.coreservlets.com/>

Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Most Common Elements with valueChangeListener attributes

- **Check boxes**
 - h:selectBooleanCheckbox
 - Assign value to a boolean bean property. Value always set to true or false (unlike Struts where only true set)
- **Comboboxes**
 - h:selectOneMenu
 - h:selectManyMenu
- **List boxes**
 - h:selectOneListbox
 - h:selectManyListbox
- **Radio Buttons**
 - h:selectOneRadio
 - Same improved behavior as checkboxes
- **Textfields**
 - h:inputText

Comboboxes (h:selectOneMenu) & List boxes (h:selectOneListbox)

- Specifying one item at a time
- Specifying entire list in one fell swoop

```
<h:selectOneMenu ...>  
  <f:selectItem  
    itemValue="..."  
    itemLabel="..." />  
  <f:selectItem  
    itemValue="..."  
    itemLabel="..." />  
  <f:selectItem  
    itemValue="..."  
    itemLabel="..." />  
</h:selectOneMenu>
```

- Note error in *Core JSF* where itemLabel omitted

```
<h:selectOneMenu ...>  
  <f:selectItems  
    value="..." />  
</h:selectOneMenu>
```

25

Comboboxes and List boxes: Getting the Selected Value

- The value attribute of h:selectOneMenu designates where selected value is stored

```
<h:selectOneMenu value="#{myBean.myProperty}">  
  <f:selectItems value="..." />  
</h:selectOneMenu>
```

- The list of items in the combobox is taken from f:selectItems
- When form first displayed, getMyProperty is called, and if it matches any of the list of items, that is the initially selected value
 - Otherwise the first value is the initially selected one
- When form is submitted (e.g. with h:commandButton), setMyProperty is called with the current selection

26

Comboboxes and List boxes: Specifying List of Choices

- The value of `f:selectItems` should be a List or array of `java.faces.model.SelectItem`
 - See example in event-handling section
- **SelectItem** has two main constructors
 - One just specifies a String
 - Identical value displayed and passed to bean setter
 - One specifies a Java object and a String
 - String displayed, but actual Java object is the value supplied to bean setter (i.e., mapping performed automatically by JSF)

```
SelectItem[] listBoxItems =  
    { new SelectItem(realValue1, "DisplayVal1"),  
      new SelectItem(realValue2, "DisplayVal2"),  
      ...  
    }
```

27

Comboboxes and Checkboxes: Example (Bean)

```
public class ColorBean implements Serializable {  
    private String fgColor = "Red";  
    private String bgColor = "Yellow";  
    private SelectItem[] colorNames =  
        { new SelectItem("Red"),  
          new SelectItem("#FFA500", "Orange"),  
          new SelectItem("Yellow"),  
          new SelectItem("Green"),  
          new SelectItem("Blue"),  
          new SelectItem("#4B0082", "Indigo"),  
          new SelectItem("#EE82EE", "Violet") };  
    private boolean isValidatingColors = false;  
  
    // getFgColor, setFgColor, getBgColor, setBgColor,  
    // isValidatingColors, setValidatingColors  
  
    public SelectItem[] getColorNames() {  
        return (colorNames);  
    }  
}
```

Value displayed to user

Value passed to setter methods

28

Comboboxes and Checkboxes (Bean)

```
public String showPreview() {
    if (isValidatingColors && fgColor.equals(bgColor)) {
        return("same-color");
    } else {
        return("show-preview");
    }
}
```

29

Comboboxes and Checkboxes (Form)

```
<h:form>
    Foreground color:
    <h:selectOneMenu value="#{colorBean.fgColor}">
        <f:selectItems value="#{colorBean.colorNames}" />
    </h:selectOneMenu>
    <BR>
    Background color:
    <h:selectOneMenu value="#{colorBean.bgColor}">
        <f:selectItems value="#{colorBean.colorNames}" />
    </h:selectOneMenu><BR>
    Verify that foreground and background are different?
    <h:selectBooleanCheckbox
        value="#{colorBean.validatingColors}" />
    <P>
    <h:commandButton value="Show Preview"
        action="#{colorBean.showPreview}" />
</h:form>
```

Where to store selection (also initial value)

List of choices

30

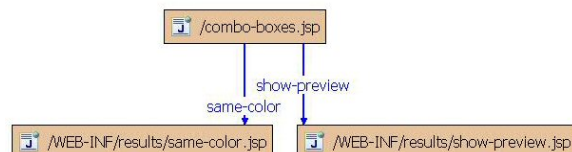
Comboboxes and Checkboxes (Form Result)



31

Comboboxes and Checkboxes (faces-config.xml)

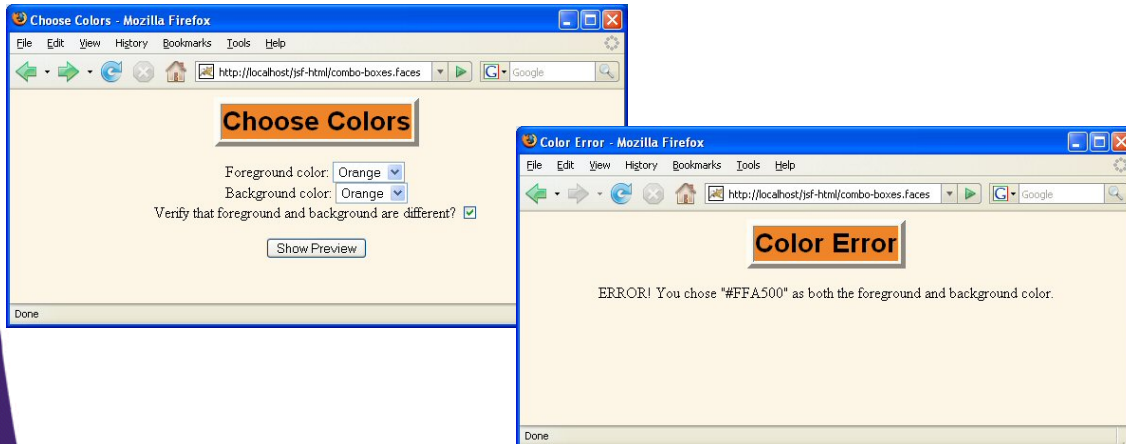
```
<managed-bean>
  <managed-bean-name>colorBean</managed-bean-name>
  <managed-bean-class>
    coreservlets.ColorBean
  </managed-bean-class>
  <managed-bean-scope>request</managed-bean-scope>
</managed-bean>
<navigation-rule>
  <from-view-id>/combo-boxes.jsp</from-view-id>
  <navigation-case>
    <from-outcome>same-color</from-outcome>
    <to-view-id>/WEB-INF/results/same-color.jsp</to-view-id>
  </navigation-case>
  <navigation-case>
    <from-outcome>show-preview</from-outcome>
    <to-view-id>/WEB-INF/results/show-preview.jsp</to-view-id>
  </navigation-case>
</navigation-rule>
```



32

Comboboxes and Checkboxes (Results Pages)

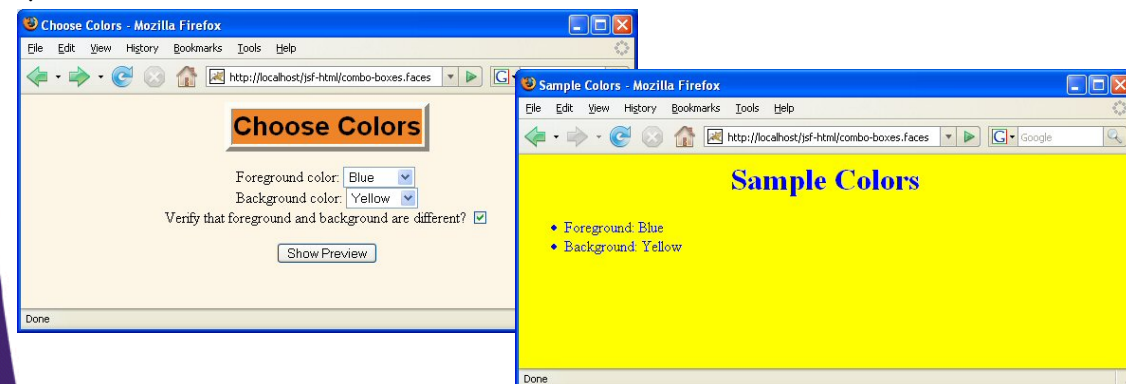
```
<TABLE BORDER=5>
  <TR><TH CLASS="TITLE">Color Error</TH></TR>
</TABLE>
<P>
ERROR! You chose
" <h:outputText value="#{colorBean.fgColor}"/> "
as both the foreground and background color.
```



33

Comboboxes and Checkboxes (Results Pages)

```
<BODY TEXT="<h:outputText value="#{colorBean.fgColor}"/>"
  BGCOLOR="<h:outputText value="#{colorBean.bgColor}"/>">
<H1 ALIGN="CENTER">Sample Colors</H1>
<UL>
  <LI>Foreground:
    <h:outputText value="#{colorBean.fgColor}"/>
  <LI>Background:
    <h:outputText value="#{colorBean.bgColor}"/>
</UL>
```



34



Outputting Values

Customized Java EE Training: <http://courses.coreservlets.com/>

Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.

Output Options

- **h:outputText**
 - Outputs bean property or collection element
 - Values from properties file loaded into a collection via `f:loadBundle`
 - Roughly equivalent to `bean:write` from Struts
 - Attributes
 - `styleClass` or `style`: results in SPAN element instead
 - `escape`: false value means not to filter `<` and `>`
- **h:outputFormat**
 - Outputs parameterized error messages
 - Parameters supplied with `f:param`
 - See section on resource bundles

Summary

- **Most common form**

```
<h:form>  
  <h:inputText value="#{beanName.propertyName}"/>  
  ...  
  <h:commandButton ... action="#{beanName.methodName}"/>  
</h:form>
```

- **Command items**

- h:commandButton (with value or image)
- h:commandLink

- **Value change items**

- h:selectBooleanCheckbox
- h:selectOneMenu
- h:selectManyMenu
- h:selectOneListbox
- h:selectManyListbox
- h:selectOneRadio
- h:inputText

37

© 2012 Marty Hall



Questions?

Customized Java EE Training: <http://courses.coreservlets.com/>

Java, JSF 2, PrimeFaces, Servlets, JSP, Ajax, jQuery, Spring, Hibernate, RESTful Web Services, Hadoop, Android.
Developed and taught by well-known author and developer. At public venues or onsite at *your* location.