Chapter 4 STD JSF Tags

1. Styles

➔ One can use styles, either inline(style) or classes(styleClass) to influence how components are rendered.

```
<h:outputText value="#{customer.name}" styleClass="emphasis"/>
<h:outputText value="#{customer.id}" style="border: thin solid blue"/>
```

➔ CSS style attributes can be value expressions that give programmatic control over style.

2. Resources

➔ One can place all style sheets, java script files, images and other files into 'resources' directory in the root of web app.

➔ Sub-directories of 'resources' directories are called libraries. One can create libraries as one like. Ex css, images, java script etc.

➔ To include style sheet use tag

```
<h:outputStylesheet library="css" name="styles.css"/>
```

The tag adds a link of the form

```
<link href="/context-root/faces/javax.faces.resource/styles.css?ln=css" rel="stylesheet"
type="text/css"/>
```

➔ To include java script use tag `<h:outputScript>`

```
<h:outputScript name="jsf.js" library="javascript" target="head"/>
```

➔ Here 'target' could be 'head' or 'body'. The script is appended to the 'head' or 'body' facet of the root component, which means that it appears at the end of the head or body in the generated HTML. If there is no target the script will be inserted in the current location.

➔ To include image library use <h:graphicImage> tag

```
<h:graphicImage name="logo.png" library="images"/>
```

➔ One can provide version for library

Ex.

```
resources/css/1_0_2
resources/css/1_1
```

Then latest version `resources/css/1_1` will be used.

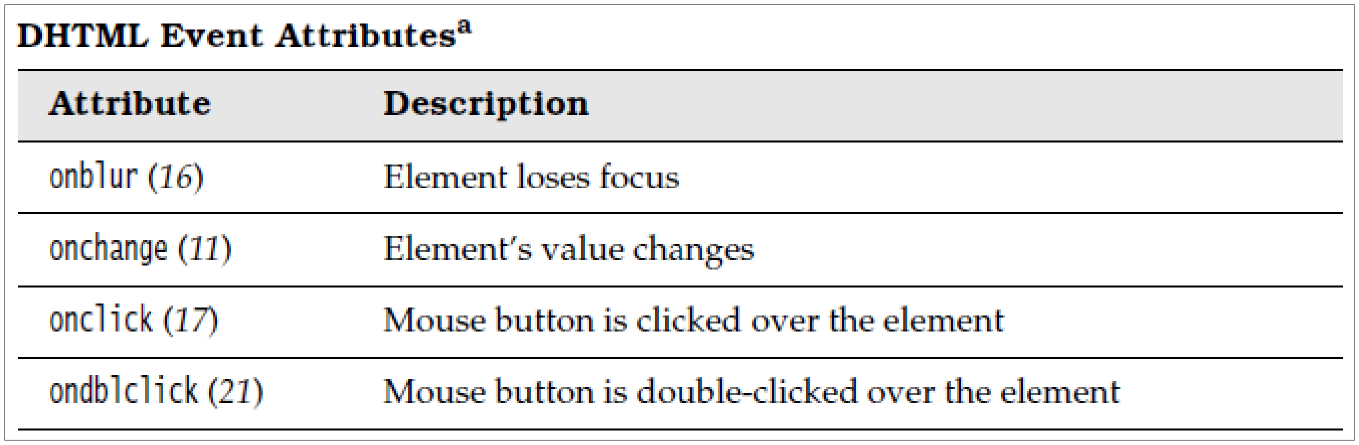
➔ One can even provide version of file also where one need to replace the resource with directory of same name and then use version name as file name. Example is shown below

```
resources/css/styles.css/1_0_2.css
resources/css/styles.css/1_1.css
```

3. DHTML Events

➔ Dynamic HTML is supported by nearly all JSF HTML tags.

➔ Some DHTML attributes are shown below(there are many such attributes).



DHTML Event Attributes[a]

Attribute	Description
onblur *(16)*	Element loses focus
onchange *(11)*	Element's value changes
onclick *(17)*	Mouse button is clicked over the element
ondblclick *(21)*	Mouse button is double-clicked over the element

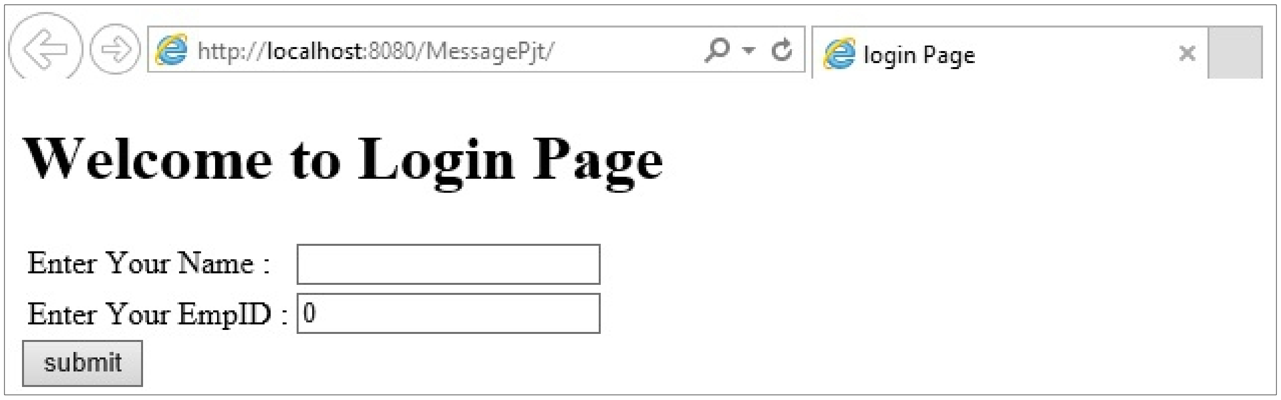
4. **Panels**

➔ We have `<h:panelGrid>` tag which generates HTML markup for laying out components in rows and columns.

➔ If we have 4 components and <h:panelGrid> specified with 2 columns then one will have two rows with 2 components each.

➔ Ex

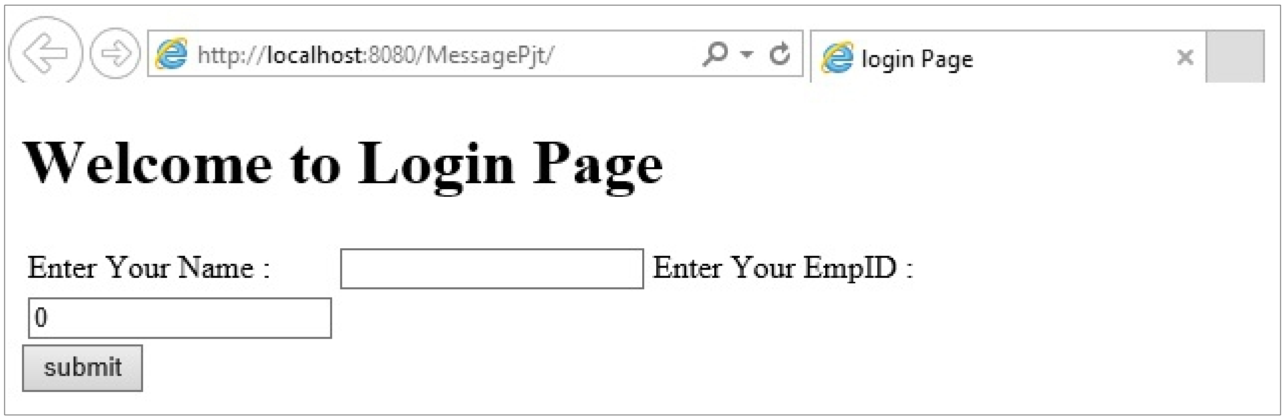
1. <h:panelGrid columns="2">



Welcome to Login Page

Enter Your Name :
Enter Your EmpID : 0

2. <h:panelGrid columns="3">



Welcome to Login Page

Enter Your Name : Enter Your EmpID :
0

➔ <h:panelGrid> is often used with <h:panelGroup> with which two or more components are grouped and then they are treated as one

```
<h:panelGrid columns=”1”>
     <h:panelGroup>
     <h:inputText id='name' value=#{user.name}”>
     <h:message for=”name”/>
     </h:panelGroup>
</h:panelGrid>
```

➔ If no. of columns are not mentioned then by default it is '1'.