

Objectives

- ◆ In this session, you will learn to:
 - ◆ Understand JSP technology
 - ◆ Understand JSP lifecycle

Identifying the Components of a JSP Page

- ◆ JSP is a technology that allows developers to easily create dynamic Web pages.
- ◆ A Web page that is developed using the JSP technology has the file extension, .jsp.
- ◆ A JSP page consists of the following components:
 - ◆ JSP comments
 - ◆ JSP directives
 - ◆ JSP declarations
 - ◆ JSP scriptlets
 - ◆ JSP expressions
 - ◆ JSP actions
 - ◆ JSP implicit objects

Identifying the Components of a JSP Page (Contd.)

◆ JSP comments:

- ◆ are used to explain the JSP code written in a JSP page.
- ◆ are not included in the HTTP response.
- ◆ can be added in a JSP page by using any of the following code snippets:
 - ◆ `<%-- comments --%>`
 - ◆ `<% /** this is a comment ... **/ %>`
 - ◆ `<!-- comments ... -->`

◆ JSP directives:

- ◆ provide global information about a particular JSP page.
- ◆ are of the following types:
 - ◆ page directive
 - ◆ taglib directive
 - ◆ include directive

Identifying the Components of a JSP Page (Contd.)

- ◆ The page directive:
 - ◆ is used to define the attributes that notify the Web container about the general settings of a JSP page.
 - ◆ has the following attributes:
 - ◆ `language`: defines the scripting language of the JSP page.
 - ◆ `extends`: defines the parent class that the JSP generated servlet extends.
 - ◆ `import`: imports the list of packages, classes, or interfaces into the generated servlet.
 - ◆ `session`: specifies if the generated servlet can access the session or not.
 - ◆ `buffer`: specifies the size of the out buffer.
 - ◆ `autoFlush`: specifies that the out buffer be flushed automatically if the value is set to `true`.
 - ◆ `isThreadSafe`: specifies whether a JSP page is thread-safe or not.

Identifying the Components of a JSP Page (Contd.)

- ◆ `errorPage`: specifies the URL of a JSP page that will handle any unchecked Java exception.
- ◆ `isErrorPage`: specifies that the current JSP page is an error page, if the attribute value is set to `true`.
- ◆ `isELIgnored`: specifies that the current JSP page will ignore all the EL expressions, if this attribute is set to `true`.
- ◆ `info`: provides a description of a JSP page.
- ◆ `pageEncoding`: specifies the language used by the JSP page to send the response to the Web browser.
- ◆ `contentType`: defines the MIME type for a response.
- ◆ The `taglib` directive:
 - ◆ is used to import a custom tag into the current JSP page.
 - ◆ is added in a JSP page using the following code snippet:

```
<%@ taglib uri="tag_lib_URI" prefix="prefix" %>
```
 - ◆ has the following attributes:
 - ◆ `Uri`: locates the tld file of a custom tag.
 - ◆ `Prefix`: defines a prefix string to be used for distinguishing a custom tag instance.

Identifying the Components of a JSP Page (Contd.)

- ◆ The include directive:
 - ◆ is used to include the output of the file specified using the file attribute in the form of its relative URL in the calling JSP page.
 - ◆ is added in a JSP page using the following code snippet:

```
<%@ include file = "URLname" %>
```
- ◆ JSP declarations:
 - ◆ provide a mechanism to define variables and methods in a JSP page.
 - ◆ are placed within the `<%!` and `%>` symbols and end with a semicolon.

Identifying the Components of a JSP Page (Contd.)

- ◆ The following code snippet uses JSP declarations to define variables and methods:

```
<%!  
int i=5;  
int add()  
{  
i=i+5;  
return i;  
}  
%>
```

Identifying the Components of a JSP Page (Contd.)

- ◆ JSP expressions are:
 - ◆ used to directly insert values into the response output.
 - ◆ evaluated when a user makes an HTTP request.
 - ◆ added in a JSP page using the following code snippet:

```
<%= expression %>
```

- ◆ The following code snippet uses JSP expressions to evaluate the value of an expression:

```
<h1>The product of 5 and 2 is: <%= (2 * 5) %></h1>
```


Identifying the Components of a JSP Page

- ◆ JSP scriptlets:
 - ◆ consist of Java code snippets that are enclosed within the `<%` and `%>` symbols.
 - ◆ are executed at the request time.

Identifying the Components of a JSP Page (Contd.)

- ◆ The following code snippet uses JSP scriptlets to include Java code in a JSP page:

```
<% int i=10;
    if(i>0)
    {
        out.println("i is a positive
number");
    }
    else
    {
        out.println("i is a negative
number");
    }
%>
```

Identifying the Components of a JSP Page (Contd.)

◆ JSP actions:

- ◆ are the tags that are used to perform tasks, such as inserting files and forwarding a user to another page.
- ◆ are added in a JSP page using the following code snippet:

```
<jsp:actionname attribute="">
```

◆ The various JSP action tags are:

- ◆ `<jsp:useBean>`
- ◆ `<jsp:getProperty>`
- ◆ `<jsp:setProperty>`
- ◆ `<jsp:forward>`
- ◆ `<jsp:include>`
- ◆ `<jsp:param>`
- ◆ `<jsp:plugin>`

Identifying the Components of a JSP Page (Contd.)

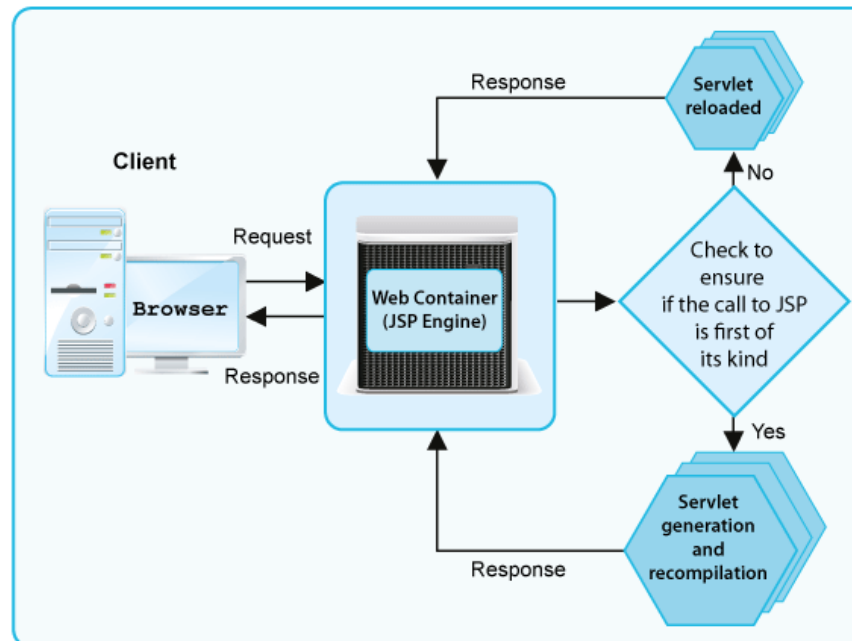
- ◆ JSP implicit objects:
 - ◆ are predefined objects provided by the container that can be included in JSP expressions and scriptlets.
 - ◆ are mapped to the classes and interfaces of the servlet API.
- ◆ The various JSP implicit objects are:
 - ◆ `request`: represents the `HttpServletRequest` object associated with the request.
 - ◆ `response`: represents the `HttpServletResponse` object associated with the response that is sent back to the browser.
 - ◆ `out`: represents the `JspWriter` object associated with the output stream of the response.
 - ◆ `session`: represents the `HttpSession` object associated with the session for the given user of the request.
 - ◆ `application`: represents the `ServletContext` object for the Web application.

Identifying the Components of a JSP Page (Contd.)

- ◆ `config`: represents the `ServletConfig` object associated with the servlet for the JSP page.
- ◆ `page`: represents the current instance of the JSP page that, in turn, is used to refer to the current instance of the generated servlet.
- ◆ `pageContext`: represents the page context for a JSP page.
- ◆ `exception`: represents the `Throwable` exception in a JSP page.

Lifecycle of a JSP Page

- ◆ Whenever the client browser requests for a particular JSP page, the server, in turn, sends a request to the JSP engine.
- ◆ A JSP engine is a part of a Web container that compiles a JSP page to a servlet.
- ◆ The following figure represents the process of the flow of events that occur after a client requests for a JSP page.



Lifecycle of a JSP Page (Contd.)

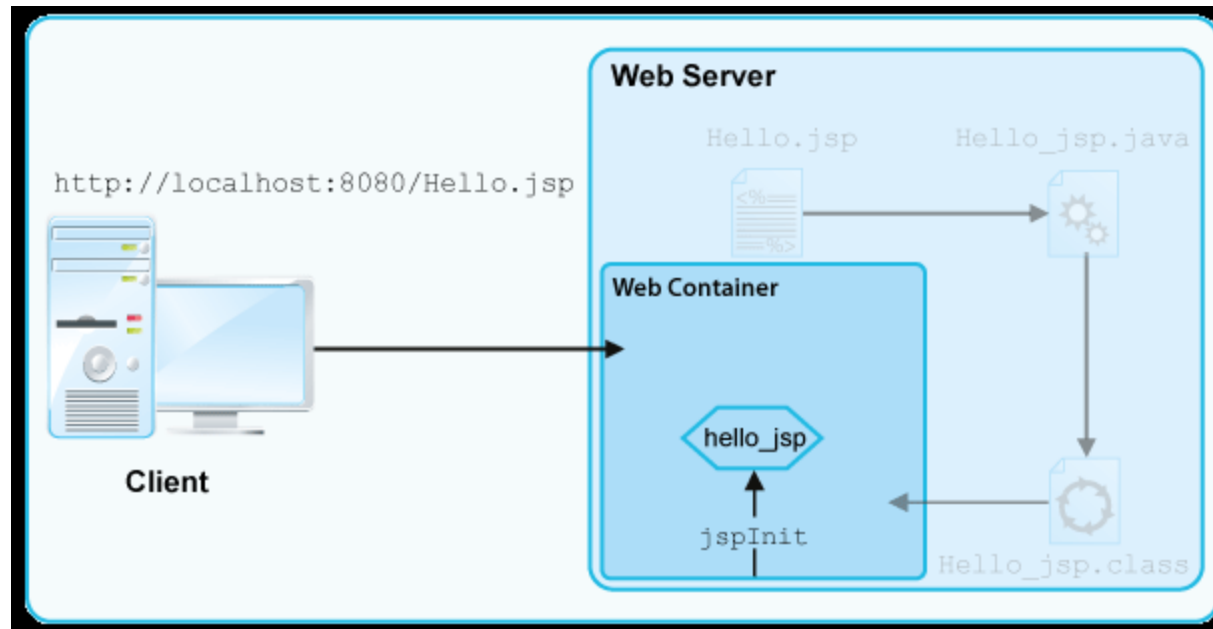
- ◆ The lifecycle of a JSP page is managed using the the following lifecycle methods of the `javax.servlet.jsp.JspPage` interface:
 - ◆ `jspInit()`
 - ◆ `_jspService()`
 - ◆ `jspDestroy()`

Processing of a JSP Page

- ◆ A JSP page needs to be converted to a servlet before it can service a client request.
- ◆ The conversion of a JSP page to servlet is carried out in the following phases:
 - ◆ Transalation
 - ◆ Compilation
 - ◆ Servlet class loading
 - ◆ Servlet instance creation
 - ◆ Servlet initialization
 - ◆ Servicing client requests
 - ◆ Servlet destruction

Processing of a JSP Page (Contd.)

- ◆ The following animation depicts the different phases of the conversion of a JSP page to servlet.



Summary

- ◆ In this session, you learned that:
 - ◆ JSP segregates the work of the Web designer and the Web developer.
 - ◆ The various components of a JSP page are:
 - ◆ JSP comments
 - ◆ JSP directives
 - ◆ JSP declarations
 - ◆ JSP scriptlets
 - ◆ JSP expressions
 - ◆ JSP actions
 - ◆ JSP implicit objects
 - ◆ JSP comments are used to add comments in a JSP page to explain the Java code and make the code more readable.
 - ◆ JSP scriptlet consists of valid Java code snippets that are enclosed within the `<%` and `%>` symbols.
 - ◆ The JSP lifecycle is managed using the following lifecycle methods of the `javax.servlet.jsp.JspPage` interface:
 - ◆ `jspInit()`
 - ◆ `jspService()`
 - ◆ `jspDestroy()`