**JavaServer Pages(JSP)**

JavaServer Pages was introduced by Sun Microsystems and was released in 1999. It is similar to ASP and PHP but it differs in the sense that it uses the Java language.

* JavaServer Pages is a technology which supports the dynamic as well as static contents of a web page basing on HTML and XML. It deals with the interface of a java web application. Codes written in Java are inserted into an HTML page with the use of JSP tags(<% … %>).
* While Servlet is HTML codes inside Java, JSP is Java codes inside HTML .
* JSP’s are based on Servlets. At runtime, JSP is translated to servlets and will be stored and reused until the original JSP is modified.
* JSP runs inside a HTTP server supporting java. An official reference implementation for this is the Apache Tomcat Server or can be Jetty. This provides an environment for Java HTTP and implements the specifications for the JavaServer Pages. At first request of a JSP page, the server will translate it into a servlet, will be compiled, loads it and then execute.

Just like servlet, JSP has its methods that corresponds to servlet’s init(), destroy(), and service().

Superinterface: javax.servlet.Servlet

Subinterface: HttpJspPage

**Methods:**

**\_jsInit()** – a method invoked only when the JSP page was initialized. It can be overridden by the page author for initializing resources such as the database.

**\_jspDestroy()** – a method called whenever JSP page is to be destroyed and is only called when the jsp container is shut down. It can be overridden by the author of the page just like the jspInit method, to release or close resource connections and open files.

**\_jspService()** – this methods corresponds to the body of the JSP page and is automatically defined by the JSP container. It generates a response for a certain request and for the HTTP methods(i.e GET, POST, DELETE). It has two parameters: the **HttpServletRequest** and the **HttpServletResponse.**

**JSP Implicit Objects**

|  |  |
| --- | --- |
| **request** | a HttpServletRequest object that specifies what data are to be included in the request. It takes value from client browser to be passed on to the response. |
| **Response** | a HTTPServletResponse object responsible for the response being sent for the client request |
| **out** | The object that sends output to the client. |
| **session** | It is an object that stores user data and making it available in  another JSP page. It keeps track of the client requests. |
| **application** | An object created once the JSP is initialized and removed if JSP is removed via the method jspDestroy() |
| **config** | Handles error messages and is only available if set to true. |
| **page** | An object to represent the JSP page. |
| **pageContext** | An object used to access a JSP page, client request and others. |
| **exception** | Contains the exception thrown from a previous page. |

**JSP Elements**

**Scriptlets**

**-** These are tags used to contain java language inside a JSP page and are placed inside the service method. When a client requests, the service method is invoked and the code it contains is executed for the request. Scriplet has the following syntax:

<% codes %>

<% ……………….. %>

**Declaration Tags in JSP**

- are tags used to declare java classes, methods, and wide variables. They are kept outside the service() method.

**Syntax** : <%! Jsp declaration %>

**Example** :

<html>

<head>

<title> JSP – Declaration Tags</title>

</head>

<body>

<%! String greeting = “Hello World”; %>

</body>

</html>

**Expression**

**-**  JSP expression tags are used in evaluating expression placed inside it and then is converted into string before sending it to the client via the response object. An expression tag is placed inside:

<%= codes here %>

Example:

<html>

<head>

<title>On Expression Tags</title></head>

<body>

<p>Today's date: <%= (new java.util.Date()).toLocaleString()%></p>

</body>

</html>

**Directives**

- These are elements used to control the JSP page processing. It provides the server necessary steps or instructions regarding how the page processing should be. A directive has the syntax:

<%directive name [attribute name=“value” …… ........]%>

3 Types:

* Page directives- These are directives for the entire JSP page containing attributes with a corresponding value and has the following syntax:

<%@page \_\_\_\_\_ = “”%>

* Include directives – This directive allows the contents of another JSP page to be used or included inside another JSP file.

Syntax : <%@include \_\_\_\_=””%>

* Taglib directives - Defines or provides tag libraries.

Sytax : <%@ taglib uri=”” %>

**JSP Expression Language**

* Expressions with the syntax ${expression}  and whatever value is placed inside the braces are evaluated. These expressions may be an arithmetic or logical. A function may also be placed inside a JSP Expression.

<html>   
 <head>  
 <title>Expression Language Example</title>  
 </head>  
 <body>

${255+262}  
 </body>   
</html>

**JSP Custom Tags**

* In JSP, you can create or provide a custom tag which you can use in the JSP page.
* Custom tags can be created by extending the **SimpleTagSupport** class and override the method  **doTag()** where you can place your code which generates the content for the tag.

import javax.servlet.jsp.tagext.\*;  
import javax.servlet.jsp.\*;  
import java.io.\*;  
public class Demo extends SimpleTagSupport {  
   public void doTag() throws JspException, IOException {  
      JspWriter out = getJspContext().getOut();  
      out.println("This is a custom tag example");  
   }  
}

**JSP Standard Elements**

* Are special XML tags inside a JSP page which controls the servlet engine behaviour.

Some of the elements are:

* jsp:include - an action tag which lets you insert a JSP file into another

Syntax :  <jsp:include page="URL"/>

* jsp:forward - an action allowing a request to be forwarded to another page. In the syntax below, there are two page values: the page to be inserted and the second to where it will be inserted.

Syntax : <jsp:forward page="error.jsp" />

<jsp:forward page="<%= java-expression %>" />

References:

<https://beginnersbook.com/2013/11/jsp-implicit-objects/>

<http://www.ntu.edu.sg/home/ehchua/programming/java/JavaServerPages.html>

<https://www.tutorialspoint.com/jsp/jsp_custom_tags.htm>

<https://docs.oracle.com/cd/E19316-01/819-3669/bnanr/index.html>

<https://www.guru99.com/jsp-elements.html>

<https://beginnersbook.com/2013/05/jsp-tutorial-life-cycle/>