

Java Programming

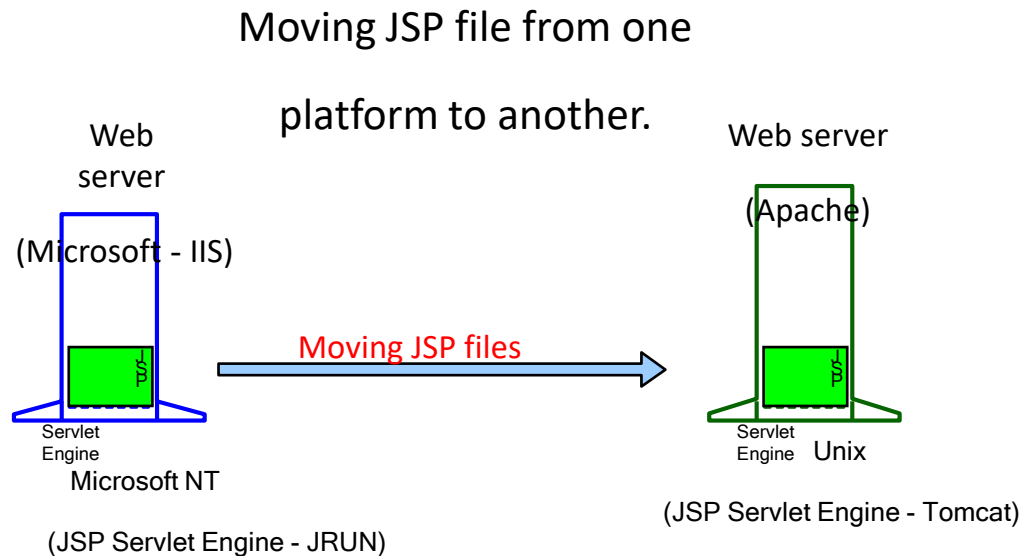
V P Jayachitra

JavaServer Pages (JSP)

- JavaServer Pages (JSP) is a server-side programming technology based on the Java language and enables the development of dynamic web sites.
- JSP was developed by Sun Microsystems to allow server side development.
- JSP files are HTML files with special Tags containing Java source code that provide the dynamic content.

Why JSP?

- JSP is based on Java, an object- oriented language.
- JSP offers a robust platform for web development.
- Multi platform
- Component reuse by using Javabeans and EJB.



JSP compared to ASP

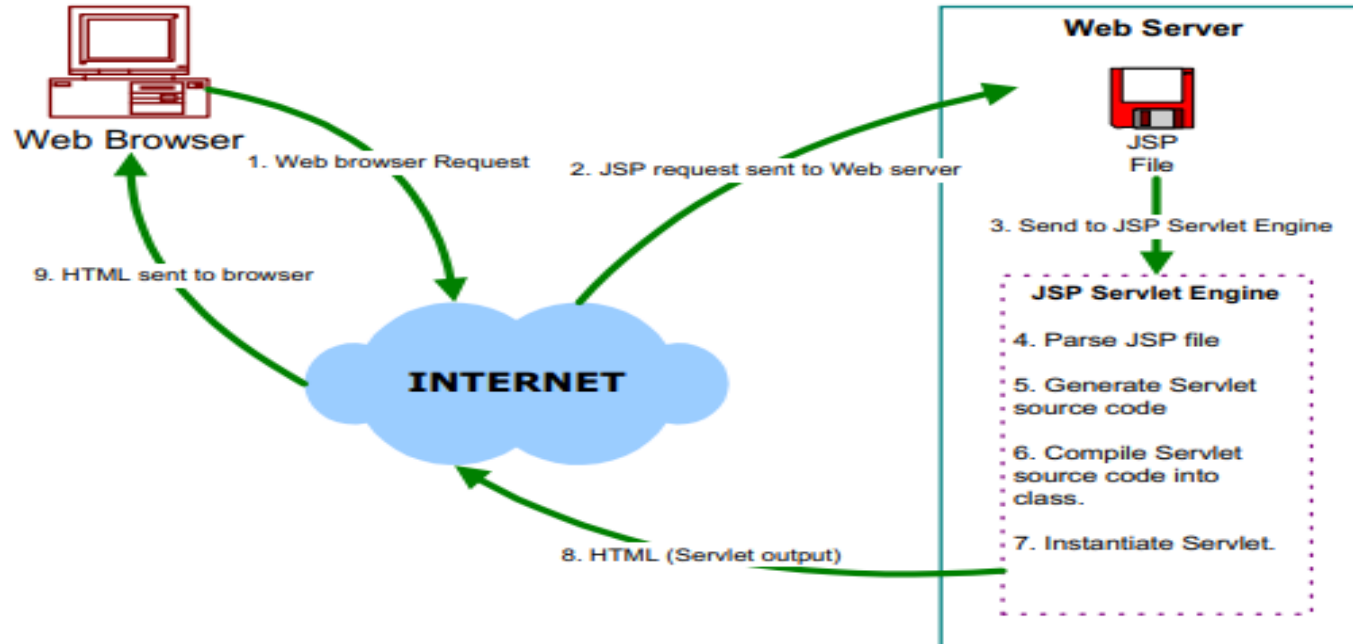
- JSP and ASP are fairly similar in the functionality that they provide.
- JSP may have slightly higher learning curve.
- Both allow embedded code in an HTML page, session variables and database access and manipulation.
- ASP is mostly found on Microsoft platforms i.e. NT
- JSP can operate on any platform that conforms to the J2EE specification.

JSP compared to Servlets

- A Servlet is a Java class that provides special server side service.
- It is hard work to write HTML code in Servlets.
- In Servlets lots of `println` statements are required to generate HTML.

JSP architecture

- JSPs are essentially an HTML page with special JSP tags embedded.
 - These JSP tags can contain Java code.
- The JSP file extension is .jsp rather than .htm or .html.
- The JSP engine parses the .jsp and creates a Java servlet source file.
- It then compiles the source file into a class file, this is done the first time and this is why the JSP is probably slower the first time it is accessed.
- Any time after this the special compiled servlet is executed and is therefore returns faster.



Creating first JSP page

- **JSP Comments** `<%-- JSP comment --%>`

```
<html>
<head>
<title>My first JSP page
</title>
</head>
<body>
<%-- This JSP comment - not visible in the page source --%>
<% out.println("Hello World"); %>
</body>
</html>
```

JSP tags

- There are five main tags:
 - Declaration tag
 - Expression tag
 - Directive Tag
 - Scriptlet tag
 - Action tag

Declaration tag (<%! %>)

- This tag allows the developer to declare variables or methods.
- Before the declaration specify <%!
- At the end of the declaration, specify %>
- Code placed in this tag must end in a semicolon ;

For Example,

```
<%!  
    private int counter = 0 ;  
%>
```

Expression tag (<%= %>)

- This tag allows the developer to embed any Java expression and is short for out.println().
- A semicolon (;) does not appear at the end of the code inside the tag.
- For example, to show the current date and time.

```
<%-- JSP comment --%>
<HTML>
<HEAD>
<TITLE>MESSAGE</TITLE>
</HEAD>
<BODY>
<% out.print("Hello, Sample JSP code"); %>
Date : <%= 5+6%>
</BODY>
</HTML>
```

Directive tag (<%@ directive ... %>)

- A JSP directive gives special information about the page to the JSP Engine.
- There are three main types of directives:
 1. **page** – processing information for this page.
 2. **Include** – files to be included.
 3. **Tag library** – tag library to be used in this page.
- Directives do not produce any visible output when the page is requested but change the way the JSP Engine processes the page.

Page directive

Language	Which language the file uses.	<code><%@ page language = "java" %></code>
extends	Superclass used by the JSP engine for the translated Servlet. extends specifies the super class of the corresponding servlet code.	<code><%@ page extends ="com.taglib..." %></code>
import	<p>Import all the classes in a java package into the current JSP page. This allows the JSP page touse other java classes.</p> <p>The following packages areimplicitly imported.</p> <p>java.lang.* javax.servlet.* javax.servlet.jsp.* javax.servlet.http.*</p>	<code><%@ page import = "java.util.*" %></code> <code><%@ page import = " " %></code>
session	Does the page make use of sessions. By default all JSP pageshave session data available. There are performance benefitsto switching session to false.	Default is set to true.
buffer	<ol style="list-style-type: none"> Controls the use of buffered output for a JSP page. buffer sets the size of the buffer used by our JSP page. The default value is 8kb. 	<code><%@ page buffer = "none" %></code> <code><%@ page buffer="19kb" %></code>
autoFlush	Flush output buffer when full. autoFlush controls the buffer output, clearing it out when the buffer size is reached. The default value is true.	<code><%@ page autoFlush = "true" %></code>

Page directive

isThreadSafe	<i>isThreadSafe</i> has a default value of <i>true</i> . <i>isThreadSafe</i> determines whether or not the JSP can use Servlet multi-threading.	<%@ page isThreadSafe="false" %>
info	info is used to set a text-based description for the JSP.	<%@ page info="This is my JSP!" %>
errorPage	<ul style="list-style-type: none">•Different page to deal with•errorPage specifies a JSP page as an error page.	<%@ page errorPage ="/error/error.jsp" %>

Include directive

- Allows a JSP developer to include contents of a file inside another.
- Typically include files are used for navigation, tables, headers and footers that are common to multiple pages.

This includes the html from privacy.html found in the include directory into the current jsp page.

```
<%@ include file = "include/privacy.html" %>
```

or to include a navigation menu (jsp file) found in the current directory.

```
<%@ include file = "navigation.jsp" %>
```

Scriptlet tag (<% ... %>)

- Between <% and %> tags, any valid Java code is called a Scriptlet. This code can access any variable declared.
- For example, to print a variable.

```
<%  
    String username = "JSP";  
    System.out.println (username);  
%>
```

Tag Lib directive

- A tag lib is a collection of custom tags that can be used by the page.
- ```
<%@ taglib uri = "tag library URI" prefix = "tag Prefix" %>
```



# Action tag

- There are three main roles of action tags :
  - enable the use of server side Javabeans
  - transfer control between pages
  - browser independent support for applets.
- **JSP Custom Tags**
  - Custom tag in JSP is know as user defined tag. In JSP a user can create its own tag for performing the specific task.

# Implicit Objects

| Variable    | Of type                                |
|-------------|----------------------------------------|
| request     | javax.servlet.http.HttpServletRequest  |
| response    | javax.servlet.http.HttpServletResponse |
| out         | javax.servlet.jsp.JspWriter            |
| session     | javax.servlet.http.HttpSession         |
| pagecontent | javax.servlet.jsp.PageContext          |
| application | javax.servlet.http.ServletContext      |
| config      | javax.servlet.http.ServletConfig       |
| page        | java.lang.Object                       |
| exception   | java.lang.Throwable                    |

# Implicit Objects

## request object

Access to information associated with a request. This object is normally used in looking up parameter values and cookies.

```
<% String NAME = request.getParameter("UNAME"); %>
Welcome : <%out.println(NAME) %>
```

# Implicit Objects

session object

A session object uses a key/value combination to store information.  
To retrieve information from a session,

```
session.putValue("visitcounter", totalvisits)
```

```
session.getValue("visitcounter")
```