



Making
IT
happen **Computing Science**

CMPUT 391

Database Management Systems

JavaServer Pages (JSP)



CMPUT 391 – Database Management Systems
Department of Computing Science
University of Alberta



What Is JSP?

- JSP – Java Server Page
- SUN's solution to both CGI and ASP
- Java version of ASP – a Web page with Java code embedded inside it that runs on the Web server
- JSP source code runs on the web server in the JSP engine. The JSP engine dynamically generates the HTML and sends the HTML output to the web browser

JSP vs. ASP

- **Fairly similar in the functionality**
- **A web page with code embedded inside it that runs on the Web server**
- **ASP-- only on Microsoft platforms,
JSP-- on **any platform** that conforms to the J2EE specification**

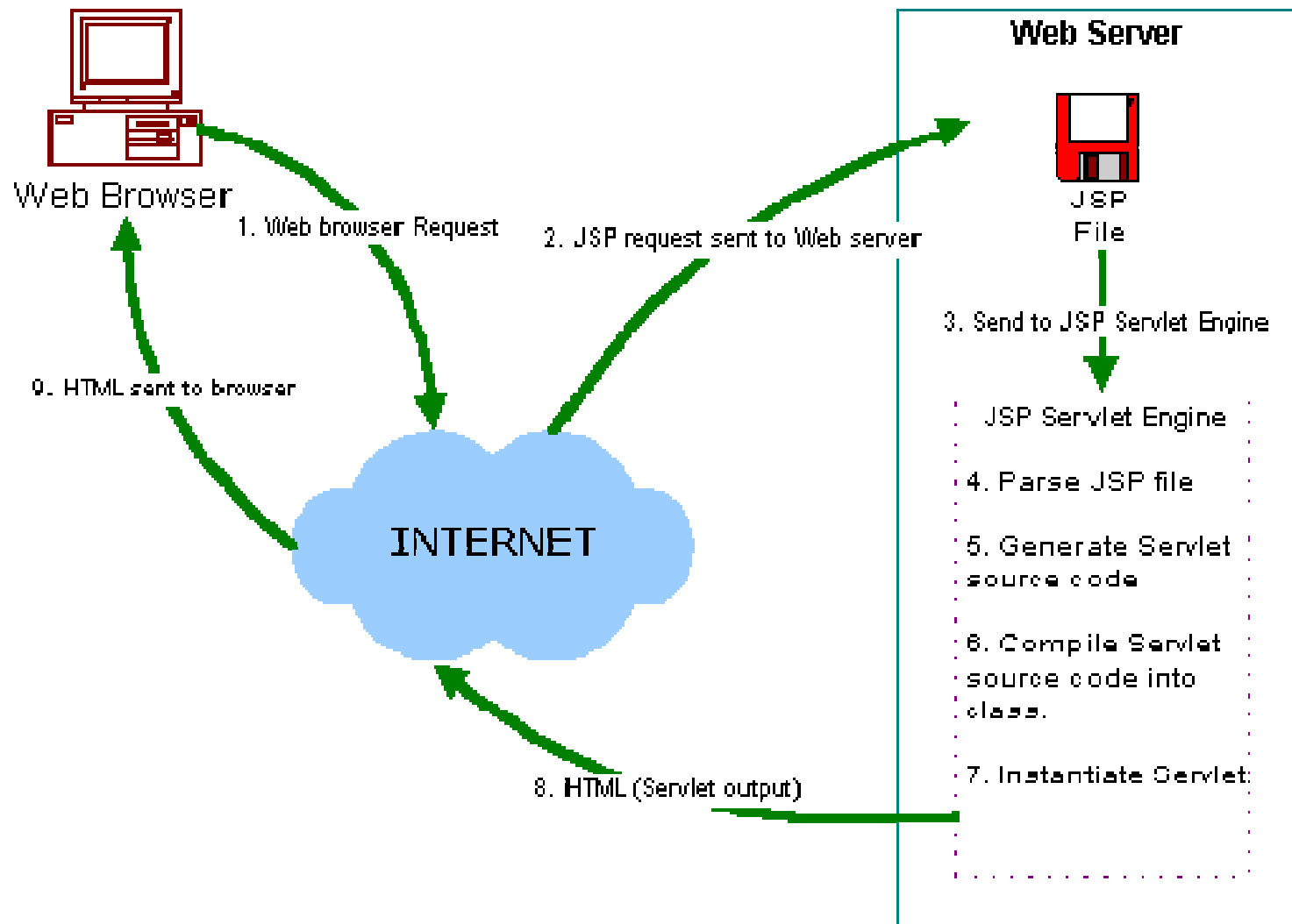
JSP vs. Servlet

- Both are Sun's technology and have many features in common
- Both can be used for generating dynamic web content
- Servlet -- Java class that provides special server side service,
JSP -- **HTML page with embedded code**
- All JSPs eventually **become servlets** before executing

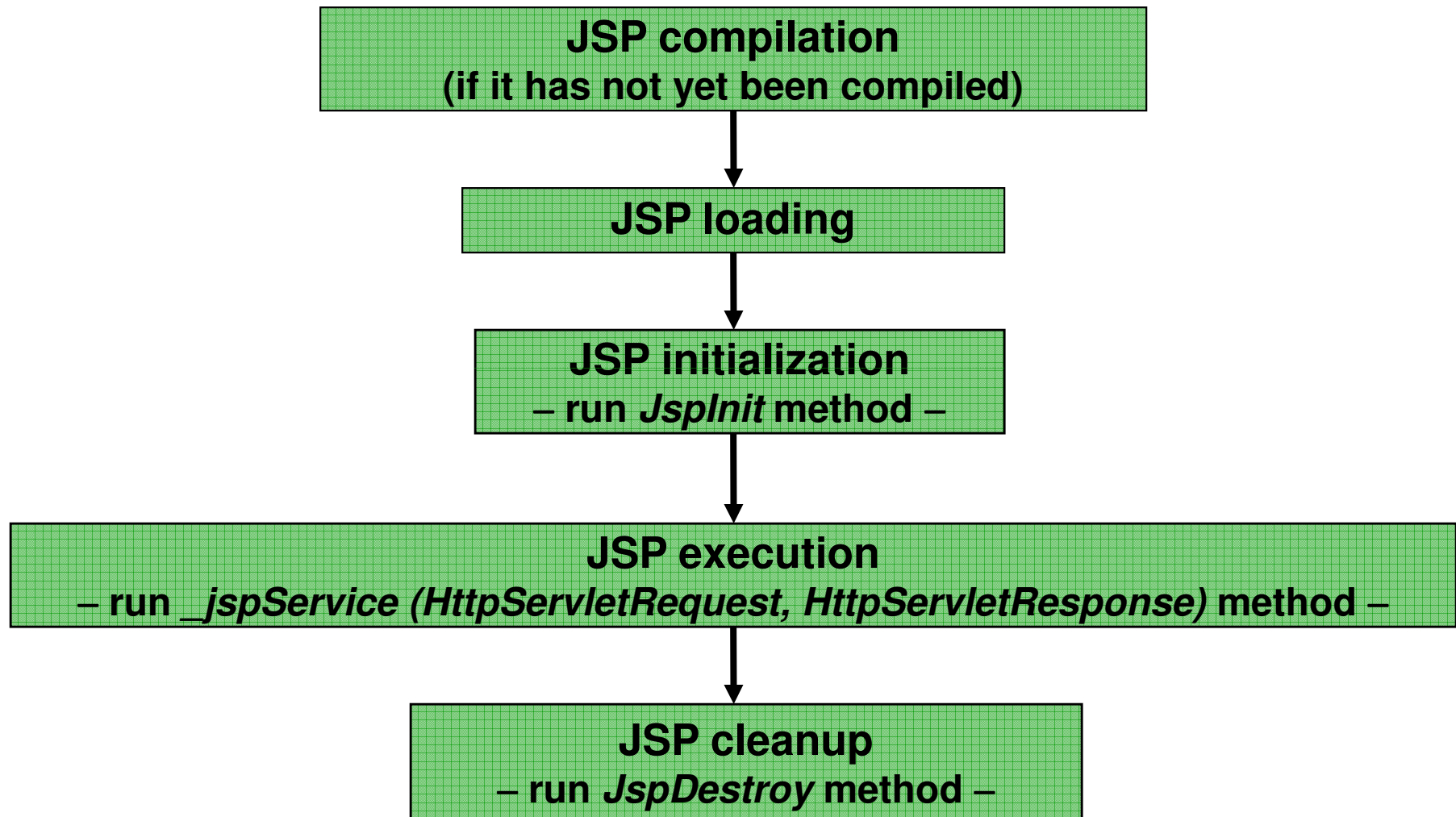
Why Use JSP?

- **Separation of static from dynamic content (different from Servlet)**
- **Write Once Run Anywhere**
- **Completely leverages the Servlet API**

JSP Architecture



JSP Lifecycle



Your First JSP

```
<HTML>
```

```
<BODY>
```

```
    Welcome to CMPUT391 Lab!
```

```
</BODY>
```

```
</HTML>
```

- Store the file in the your JSP directory:
YourHomeDirectory/catalina/webapps/proj1
- Load the new file, with the ".jsp" extension, in your browser

Find JSPExample1.jsp in “JSP-Examples.zip” posted on the course page



Adding Dynamic Content via Expressions

- The ability to embed Java
- To embed Java expressions in JSP pages put them within the Expression tag: **<%= ... %>**

```
<HTML>
  <BODY>
    Hello! The time is now
    <%= new java.util.Date() %>
  </BODY>
</HTML>
```

Find JSPExample2.jsp in “JSP-Examples.zip” posted on the course page



Scriptlets

- Scriptlet: a block of Java code inside JSP
- Executed every time JSP is invoked
- To create a scriplet, place your Java code within the tag `<% ... %>` (just like Expression tag, but without the = sign at the start of the sequence)

Scriptlets: Example

```
<HTML>
  <BODY>
    <%
      // This is a scriptlet. Notice that the "date"
      // variable we declare here is available in the
      // embedded expression later on.
      java.util.Date date = new java.util.Date();
      %>
      Hello! The time is now <%= date %>
    </BODY>
  </HTML>
```

Find JSPExample3.jsp in “JSP-Examples.zip” posted on the course page



Generating HTML in Scriptlets

- A **scriptlet** does not generate HTML by itself (unlike an Expression tag)
- Use a **variable called "out"**.
- NO need to declare
(**already predefined** for scriptlets)

Generating HTML in Scriptlets: Example

```
<HTML>
<BODY>
  <%
    // This scriptlet declares and initializes "date"
    java.util.Date date = new java.util.Date();
  %>
  Hello! The time is now
  <%
    // This scriptlet generates HTML output
    out.println(String.valueOf(date));
  %>
</BODY>
</HTML>
```

Find JSPExample4.jsp in “JSP-Examples.zip” posted on the course page



Pre-defined Objects for JSPs

- There are several objects that are automatically available in JSP:

Variable	Type
request	Javax.servlet.http.HttpServletRequest
response	Javax.servlet.http.HttpServletResponse
out	Javax.servlet.jsp.JspWriter
session	Javax.servlet.http.HttpSession
pageContext	Javax.servlet.jsp.PageContext
application	Javax.servlet.http.ServletContext
config	Javax.servlet.http.ServletConfig
page	Java.lang.Object

Mixing Scriptlets and HTML

```
<TABLE BORDER=2>
  <%   for ( int i = 0; i < 10; i++ ) {           %>
    <TR>
      <TD>Number</TD>
      <TD> <%= i+1 %> </TD>
    </TR>
  <%   }   %>
</TABLE>
```

Find JSPExample5.jsp in “JSP-Examples.zip” posted on the course page



JSP Directives

```
<%@ page import="java.util.*" %>
```

```
<HTML>
```

```
<BODY>
```

```
<% Date date = new Date(); %>
```

```
Hello! The time is now <%= date %>
```

```
</BODY>
```

```
</HTML>
```

Find JSPExample6.jsp in “JSP-Examples.zip” posted on the course page

JSP Directives (cont.)

- A JSP directive gives special information about the page to the JSP Engine.
- Directive tag: **<%@ directive ... %>**
- There are **three main types** of directives:
 - **page** - processing information for this page
 - List of imported packages:
<%@ page import="java.util.*, java.io.*" %>
 - **include** - files to be included
 - **tag library** - tag library to be used in this page.

JSP Directives: Include (Example)

```
<HTML>
```

```
<BODY>
```

Going to include another.jsp ...

```
<BR>
```

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

```
</BODY>
```

```
</HTML>
```

JSP Declarations

- For a JSP, **all the code** within the `<% %>` tags and all the expressions within the `<%= %>` tags belong to **one big Java method** in the generated servlet
- A **JSP declaration** is used to enclosed any declarations that belong outside the big method that generates the page
- Declaration tag: `<%! ... %>`

JSP Declarations: Example

```
<%@ page import="java.util.*" %>
<HTML>
  <BODY>
    <%! Date theDate = new Date();
      Date getDate()  {
        System.out.println( "In getDate() method" );
        return theDate;
      } %>
    Hello! The time is now <%= getDate() %>
  </BODY>
</HTML>
```

Find JSPExample7.jsp in “JSP-Examples.zip” posted on the course page



Inserting Comments into JSPs

- If your JSP is generating HTML, you can use **HTML** comments: `<!-- ... -->`
- **Java one-line** comment: `<%// ... %-->`
- **Java multi-liners** comments:

`<% /* %>`

`...`

`...`

`<% */ %>`

Processing a Form with JSP

```
<html>
<head></head>
<body>
  <form action="JSPExample8.jsp"
    method="post">
    Enter a test string:<br>
    <input type="text" name="test"><br>
    <input type="submit" name="submit">
  </form>
</body>
</html>
```

JSPExample8.html

```
<html>
<head></head>
<body>
  Your info has been received:
  <br><br>
  <%
    String sName =
      request.getParameter("test");
    out.print(sName);
  %>
</body>
</html>
```

JSPExample8.jsp

Find the above examples in “JSP-Examples.zip” posted on the course page

Summary

- JSPs are eventually turned into Servlets
- Though JSPs are much simpler than servlets, for some cases Servlets are still useful
- One could combine JSP and Servlet in one project

Further Readings

- Other JSP tutorials:
 - See “Reference Materials” section
- JSP specifications:
 - See “Reference Materials” section

Exercise

- Create and populate a table in your Oracle account (At least 2 columns & 4 records)
- Only use JSP, implement:
 - connect to Oracle database
 - execute the SQLStatement received from the browser (e.g. select * from “table name”;
 - display ResultSet (including column labels) & format the output (i.e. align the columns. Tip: use ResultSetMetaData & “Table” tag in HTML)
 - display an error if the SQL statement was not executed correctly in Oracle (e.g. select * from “wrong table”)