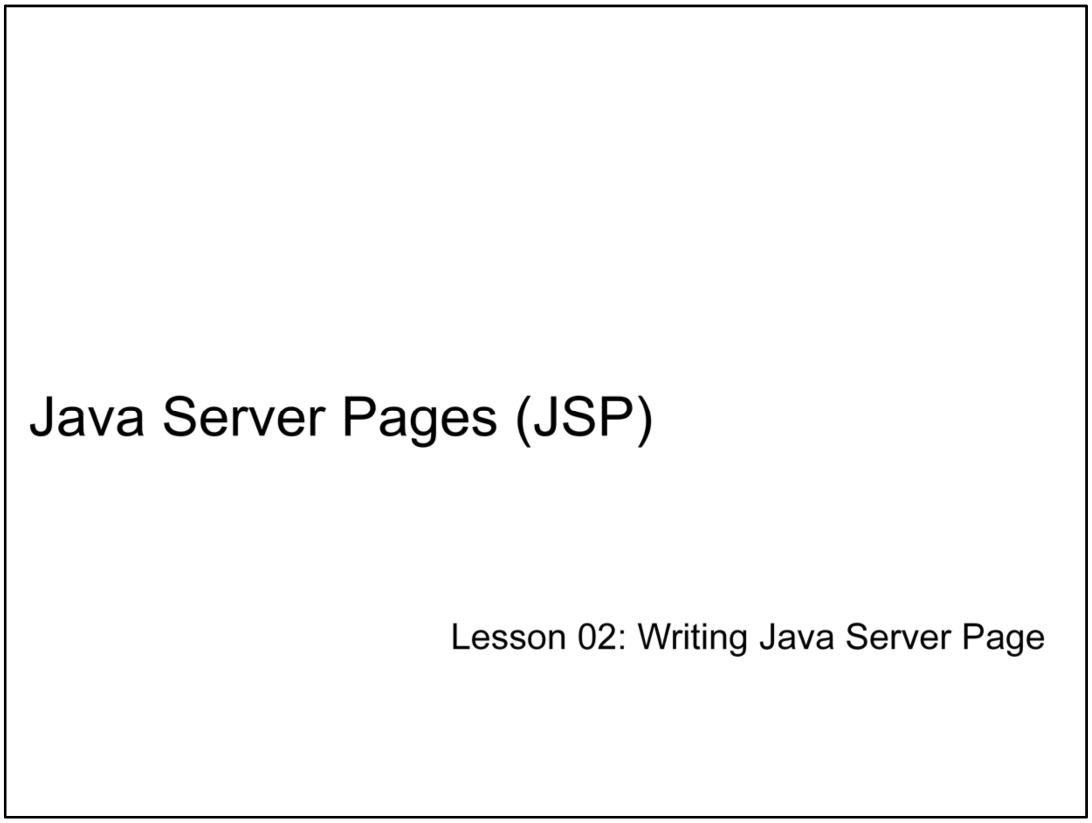
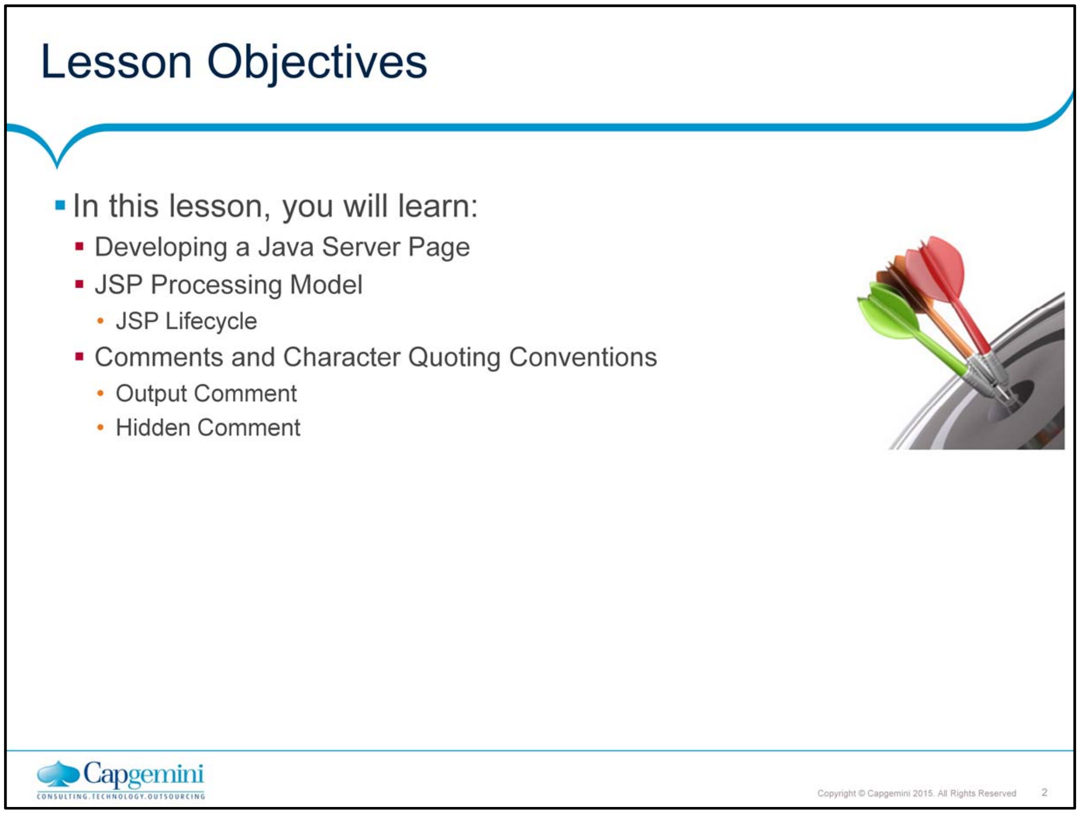


Java Server Pages (JSP)

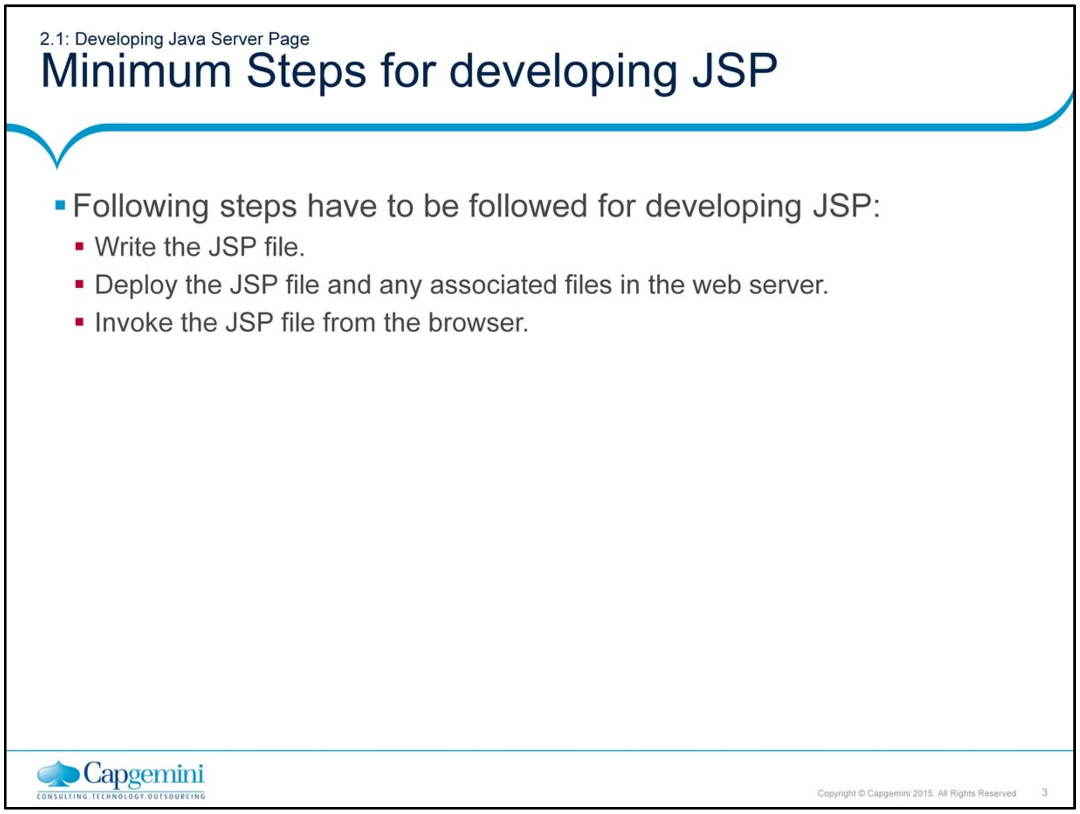
Lesson 02: Writing Java Server Page



Lesson Objectives

- In this lesson, you will learn:
  - Developing a Java Server Page
  - JSP Processing Model
    - JSP Lifecycle
  - Comments and Character Quoting Conventions
    - Output Comment
    - Hidden Comment

Capgemini CONSULTING.TECHNOLOGY.OUTSOURCING



2.1: Developing Java Server Page

Minimum Steps for developing JSP

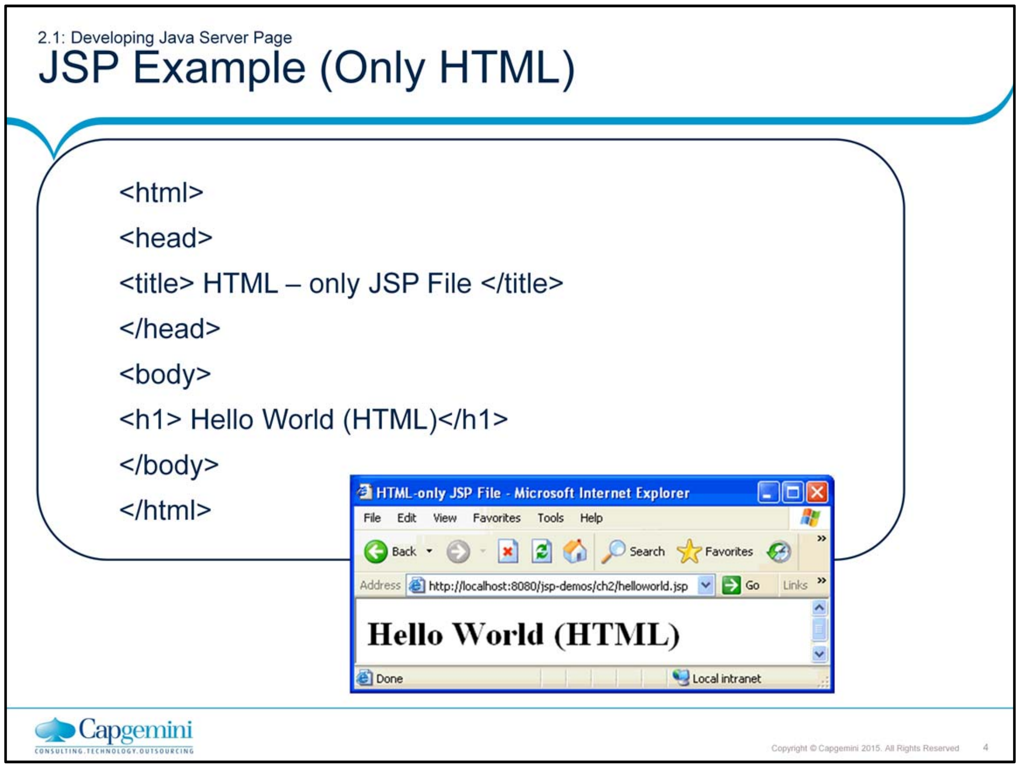
- Following steps have to be followed for developing JSP:
  - Write the JSP file.
  - Deploy the JSP file and any associated files in the web server.
  - Invoke the JSP file from the browser.

__Developing Java Server Page__**:**

Minimum steps needed to develop a Java Server Page:

Following steps have to be followed for developing JSP:

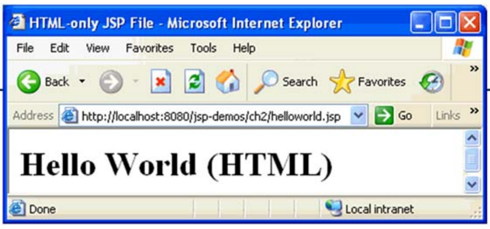
1. Write the JSP File
   a) Declare any JavaBeans components.
   b) Use **tag-centric syntax** to access Bean properties or **scripting-centric syntax** to provide desired functionality.
   c) Save the file with a **.jsp** filename extension.
2. Deploy the JSP file and any associated files in the web server (Please refer to the JSP lab book for more details).
   a) Place the **.jsp** files under the context root directory.
   b) Place associated **.class** files in **WEB-INF\classes** folder and **.jar** files in **web-inf\ lib** folder.
3. Invoke the JSP file from a web browser.
   a) Type the URL in the browser as follows:
   http://<host-name>/<context-root>/<path>/<jsp file name>



2.1: Developing Java Server Page

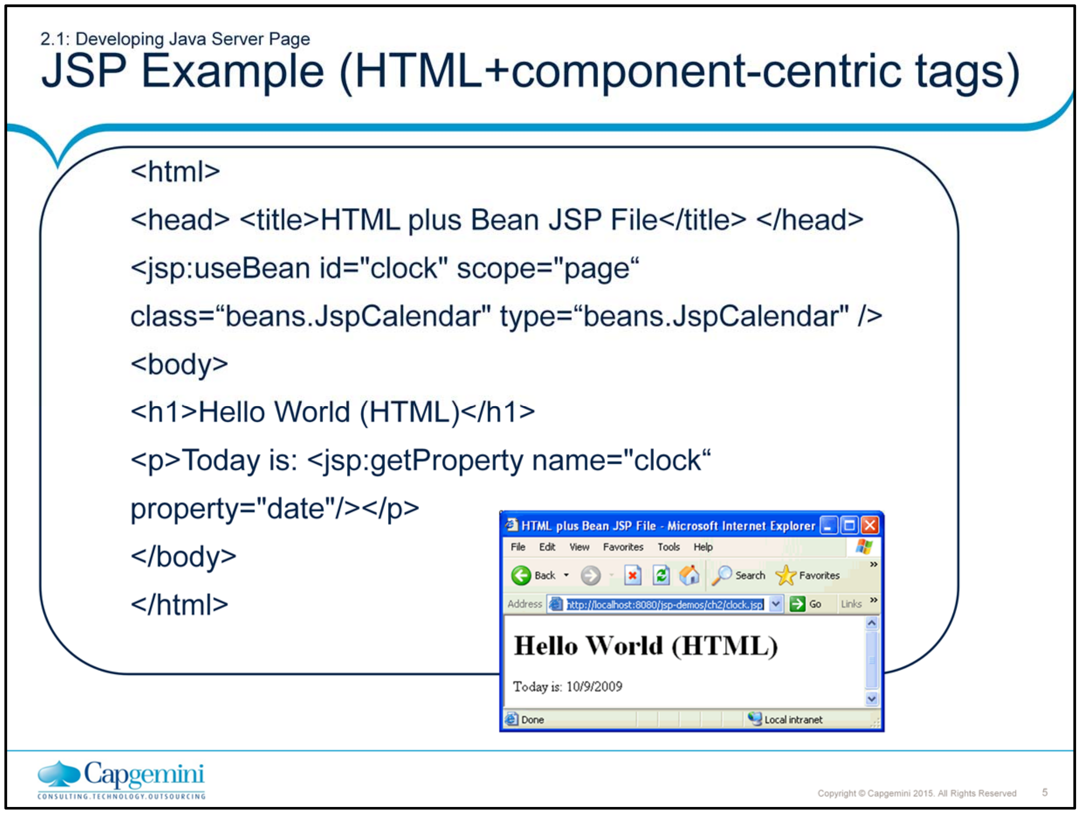
JSP Example (Only HTML)

```
<html>
<head>
<title> HTML – only JSP File </title>
</head>
<body>
<h1> Hello World (HTML)</h1>
</body>
</html>
```


HTML-only JSP File - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8080/jsp-demos/ch2/helloworld.jsp Go Links
Done Local intranet

Hello World (HTML)

Developing Java Server Page:
The above slide shows an example of a JSP file that contains only HTML.
A file containing nothing but standard HTML code can be renamed with a filename extension of .jsp and therefore meet the minimum requirements to be invoked as a Java Server Page. Such a file would still be parsed and compiled to a servlet. The servlet would still return a response when the page was invoked - its response to the client would simply contain the original HTML.



2.1: Developing Java Server Page

JSP Example (HTML+component-centric tags)

```
<html>
<head> <title>HTML plus Bean JSP File</title> </head>
<jsp:useBean id="clock" scope="page“
class=“beans.JspCalendar" type=“beans.JspCalendar" />
<body>
<h1>Hello World (HTML)</h1>
<p>Today is: <jsp:getProperty name="clock“
property="date"/></p>
</body>
</html>
```

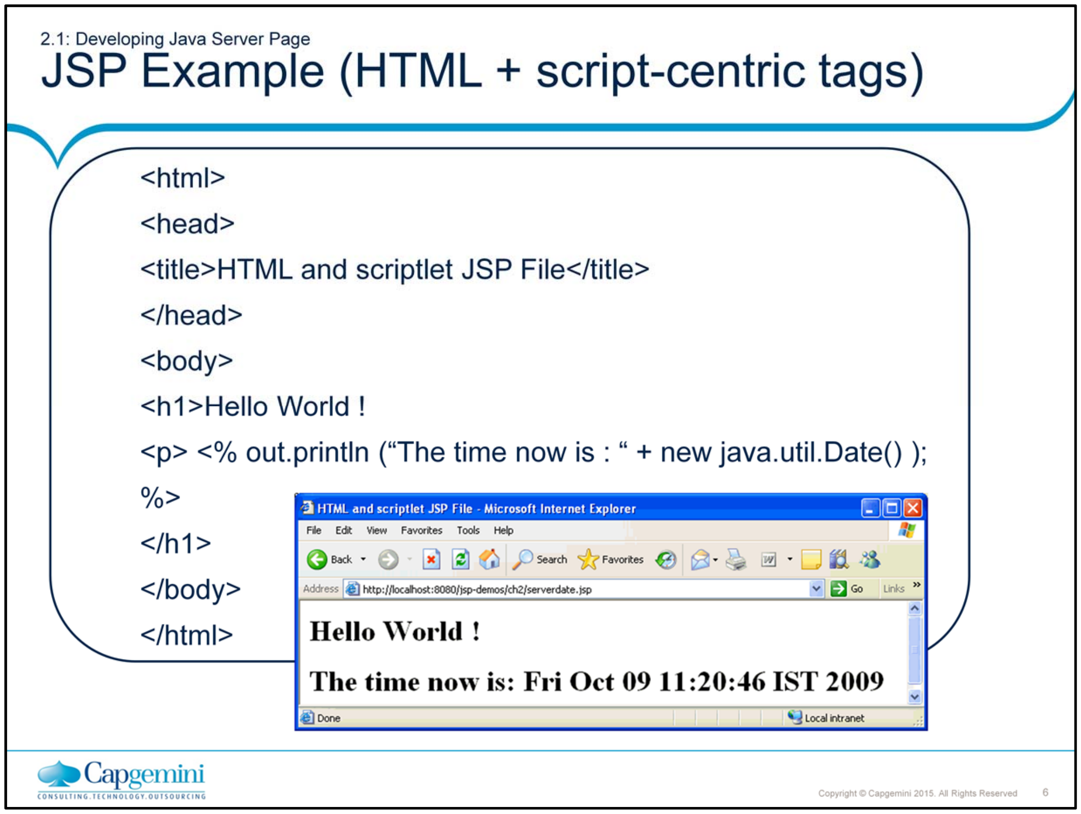
HTML plus Bean JSP File - Microsoft Internet Explorer

Address http://localhost:8080/jsp-demos/ch2/clock.jsp

```
Hello World (HTML)
Today is: 10/9/2009
```

<u>Developing Java Server Page</u>:
JSP Example (HTML + component-centric tags):

- The above slide shows an example of a JSP file that contains **HTML** and **component-centric tags**.
- Of course, the whole point of Java Server Pages technology is to simply and easily manage dynamic content on the server side. So, the plain HTML file from above, we can add some Java Server Pages tags to interact with Java components. Component can be as simple as a Bean, which returns the current time.



2.1: Developing Java Server Page

JSP Example (HTML + script-centric tags)

```
<html>
<head>
<title>HTML and scriptlet JSP File</title>
</head>
<body>
<h1>Hello World !
<p> <% out.println (“The time now is : “ + new java.util.Date() );
%>
</h1>
</body>
</html>
```

HTML and scriptlet JSP File - Microsoft Internet Explorer

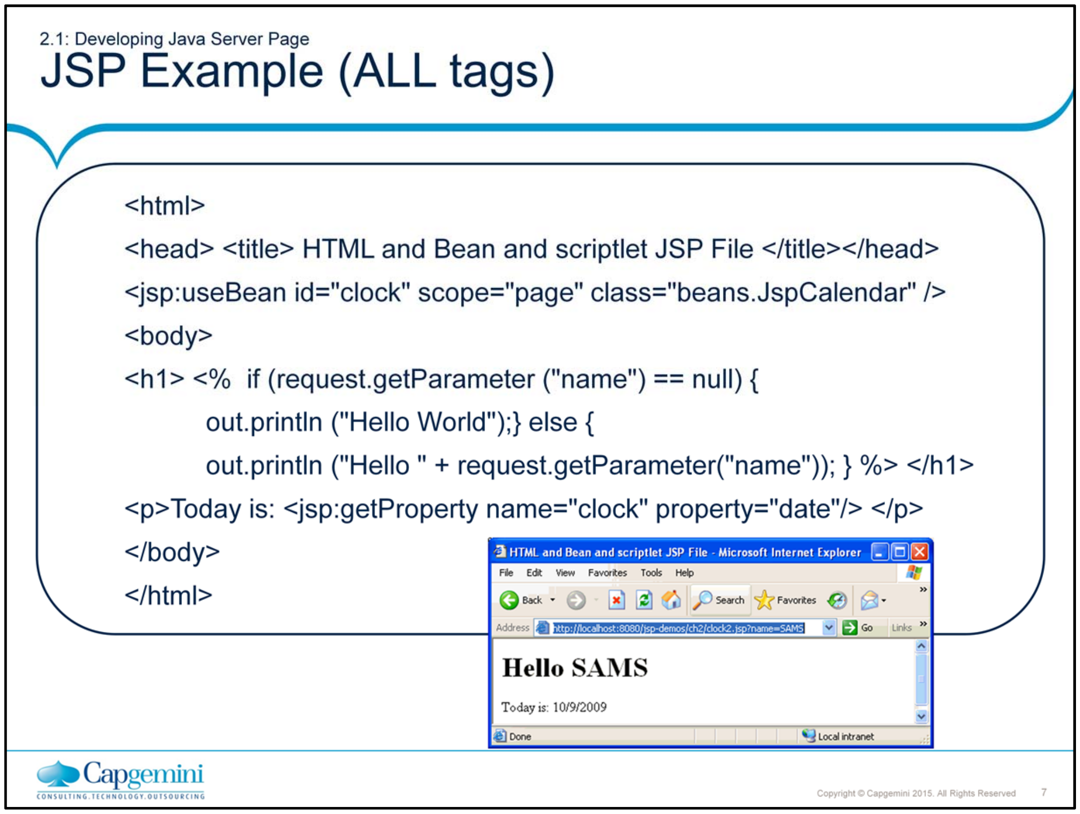
Address http://localhost:8080/jsp-demos/ch2/serverdate.jsp

```
Hello World !

The time now is: Fri Oct 09 11:20:46 IST 2009
```

File Edit View Favorites Tools Help
Back
Search
Favorites
Go
Links
Done
Local intranet
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

<u>Developing Java Server Page</u>:
JSP Example (HTML + script-centric tags)

- The above slide shows an example of a JSP file that contains **HTML** and **script-centric tags**.
- Although Java Server Pages architecture encourages the use of **componentization** for ease of maintenance and reusability, It does not required to use components. This example uses a simple **scriptlet** containing raw Java code.



2.1: Developing Java Server Page

JSP Example (ALL tags)

```
<html>
<head> <title> HTML and Bean and scriptlet JSP File </title></head>
<jsp:useBean id="clock" scope="page" class="beans.JspCalendar" />
<body>
<h1> <%  if (request.getParameter ("name") == null) {
        out.println ("Hello World");} else {
        out.println ("Hello " + request.getParameter("name")); } %> </h1>
<p>Today is: <jsp:getProperty name="clock" property="date"/> </p>
</body>
</html>
```

HTML and Bean and scriptlet JSP File - Microsoft Internet Explorer

Address http://localhost:8080/jsp-demos/ch2/clock2.jsp?name=SAMS

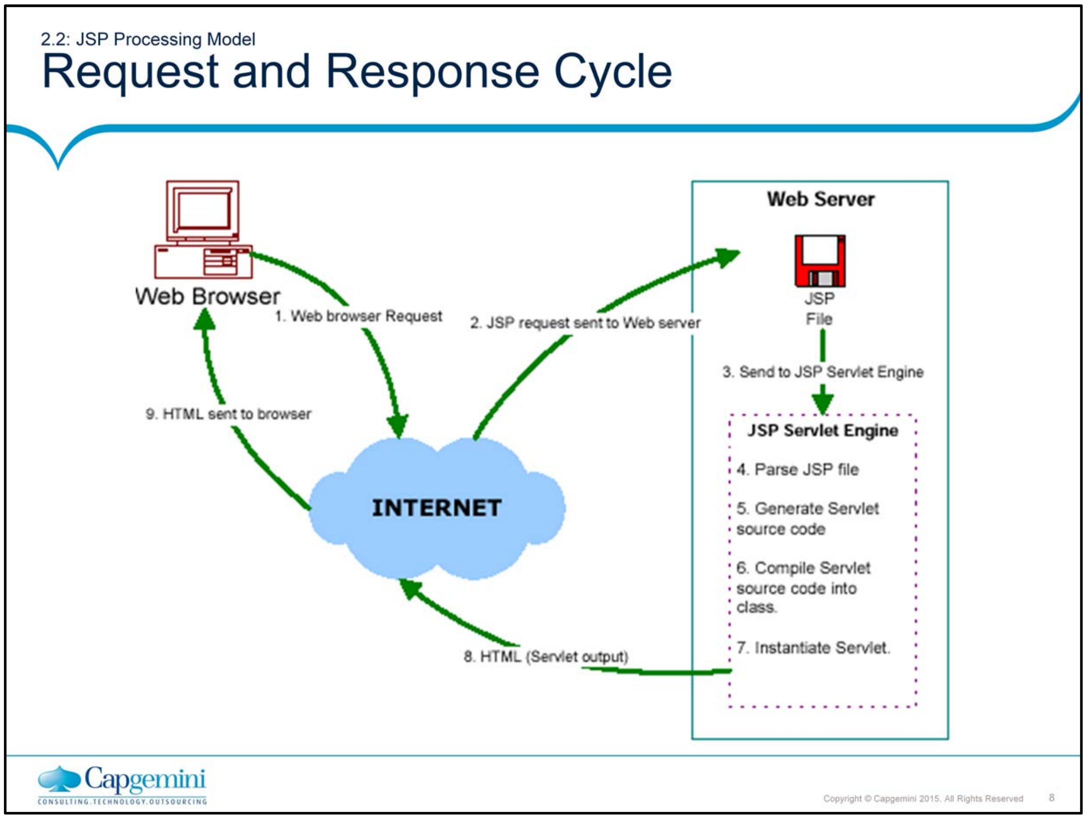
```
Hello SAMS

Today is: 10/9/2009
```

Copyright © Capgemini 2015. All Rights Reserved 7

<u>**Developing Java Server Page**</u>**:**

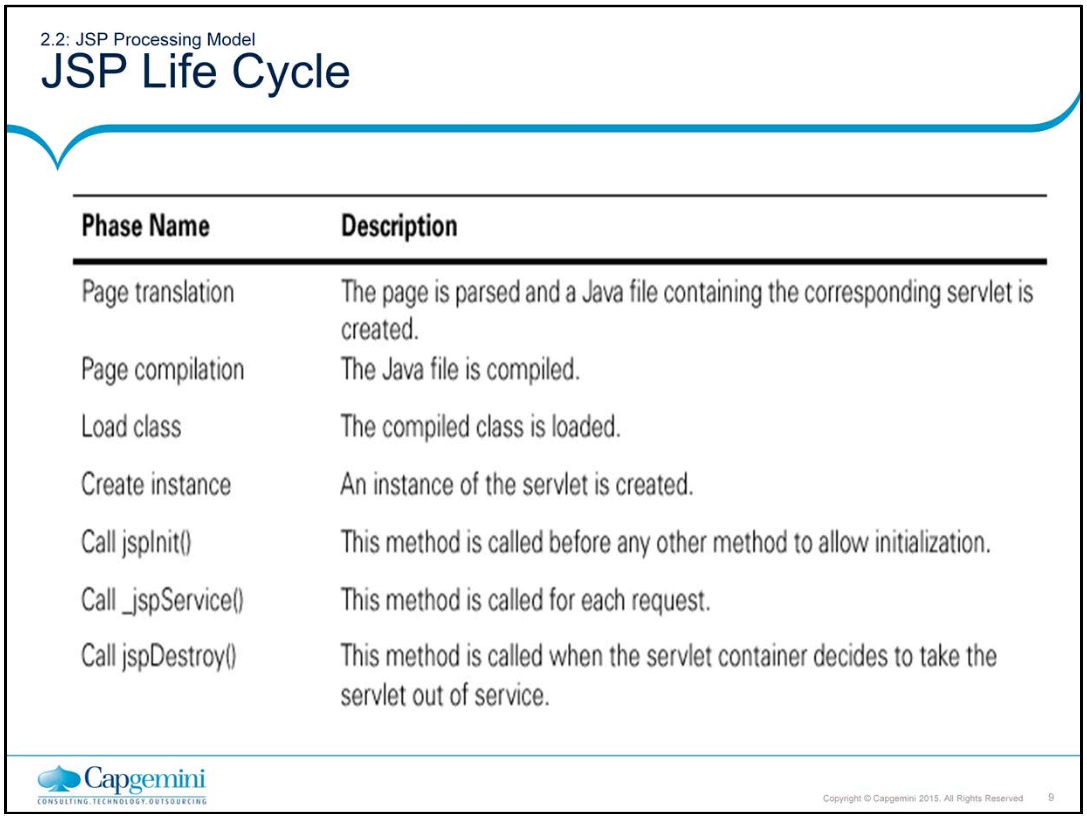
JSP Example (All tags):

- The above slide shows an example of a JSP file that contains **HTML, component centric** and **script-centric tags**.
  This is the same example where we had only HTML tags with the substitution of two lines of code for the original Hello line. By checking for a name parameter in the incoming request (available in the automatic request object), the page can return either a generic or a personalized greeting.
- Remember to save the file with a **.jsp** filename extension. This tells the web server that the file is a Java Server Pages file and to process it accordingly.



2.2: JSP Processing Model

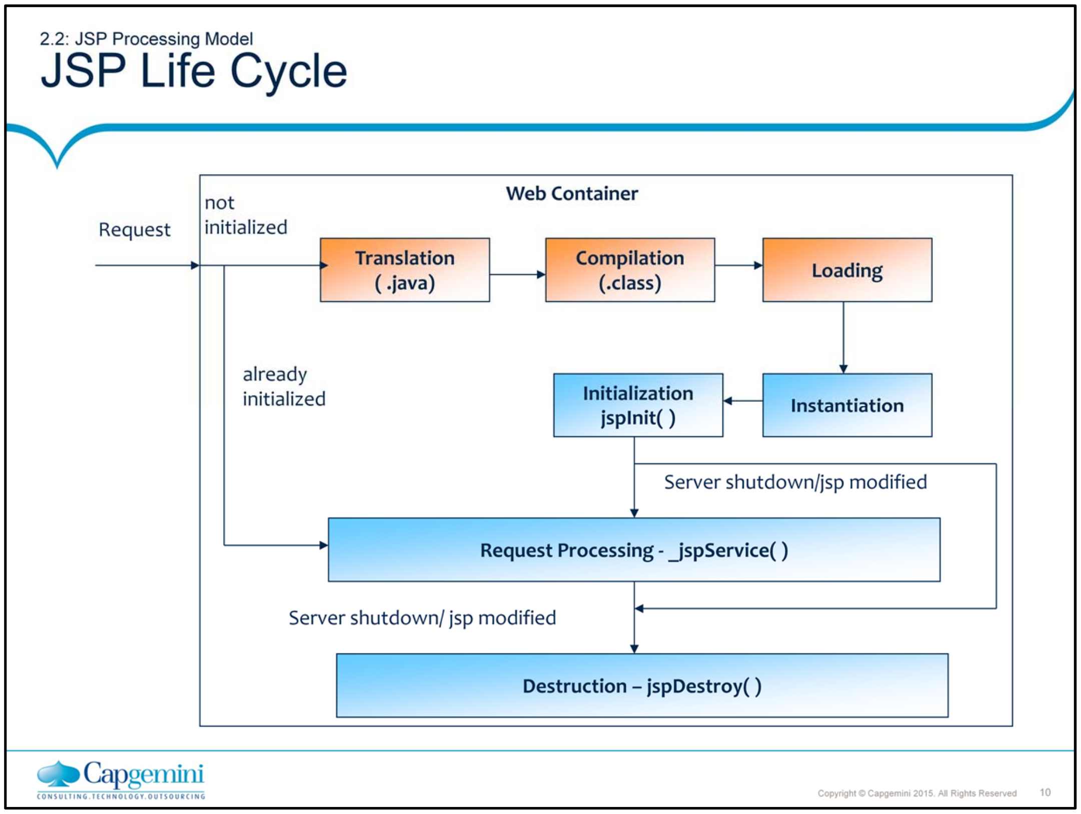
Request and Response Cycle



2.2: JSP Processing Model

JSP Life Cycle

Phase Name	Description
Page translation	The page is parsed and a Java file containing the corresponding servlet is created.
Page compilation	The Java file is compiled.
Load class	The compiled class is loaded.
Create instance	An instance of the servlet is created.
Call jspInit()	This method is called before any other method to allow initialization.
Call _jspService()	This method is called for each request.
Call jspDestroy()	This method is called when the servlet container decides to take the servlet out of service.

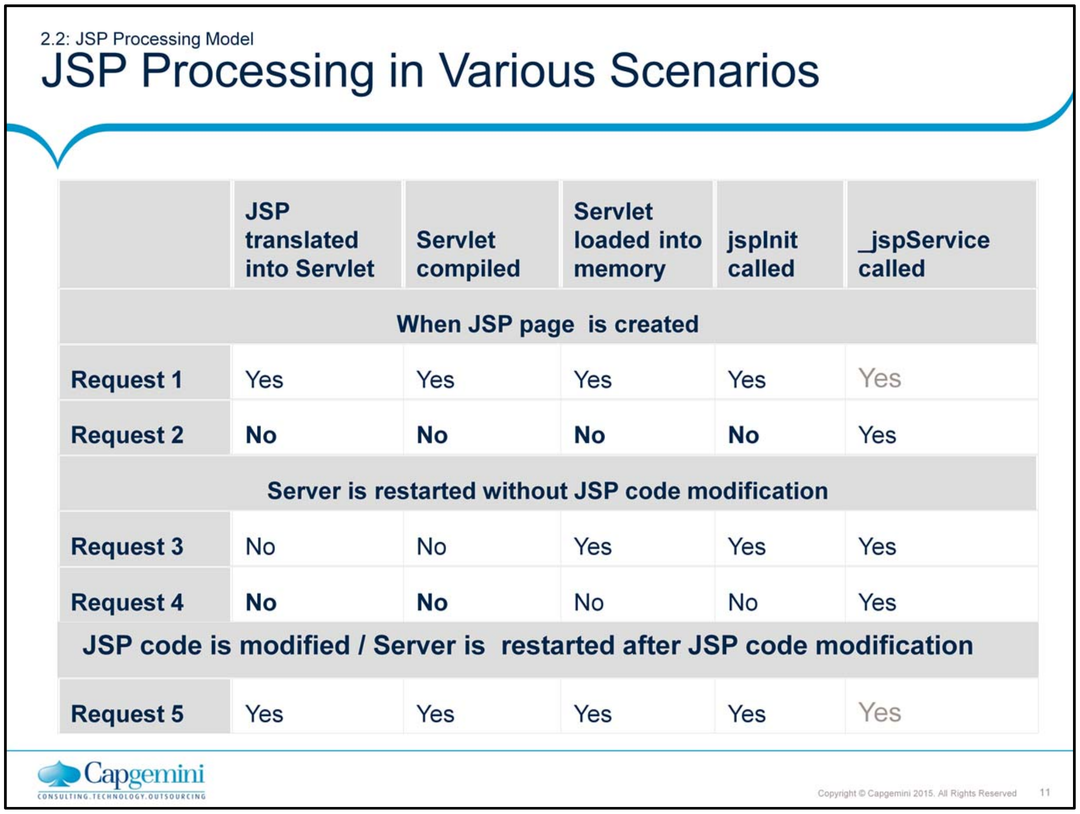

2.2: JSP Processing Model

JSP Life Cycle

Web Container
Request
not initialized
Translation (.java)
Compilation (.class)
Loading
already initialized
Initialization jspInit( )
Instantiation
Server shutdown/jsp modified
Request Processing - _jspService( )
Server shutdown/ jsp modified
Destruction – jspDestroy( )

JSP Processing Model:
JSP Life Cycle:

- As mentioned earlier, JSP is stored as text file (.jsp) in the web application. When the client requests for the page for the first time, the server (web container) translates the JSP into **Java source code** (.java) and if there are no translation errors then it is compiled into a **servlet class** file. The servlet class is then loaded by the class loader, and then instantiated. For the first time request, the **jspInit( )** method (if present) is called to initialize resources.
- If the request for the JSP page is not the first time request, then the translation to Initialization steps are skipped and the request is directly processed by the **_jspService( )** method.
- The web container invokes the **jspDestroy** method (if present), to cleanup resources, when the server shuts down or if the JSP page is replaced with a modified one.
- The lifecycle methods of the JSP page translated into the servlet have the following form:
  - ➢ public void jspInit ( )
  - ➢ public void jspDestroy( )
  - ➢ public void _jspService (HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException
- The .java and .class file generated from the JSP page have the same name as the jsp file with a postfix "**_jsp**".
  For example: If the jsp file name is **helloworld.jsp**, then the generated files have the names as **helloworld_jsp.java** and **helloworld_jsp.class**.



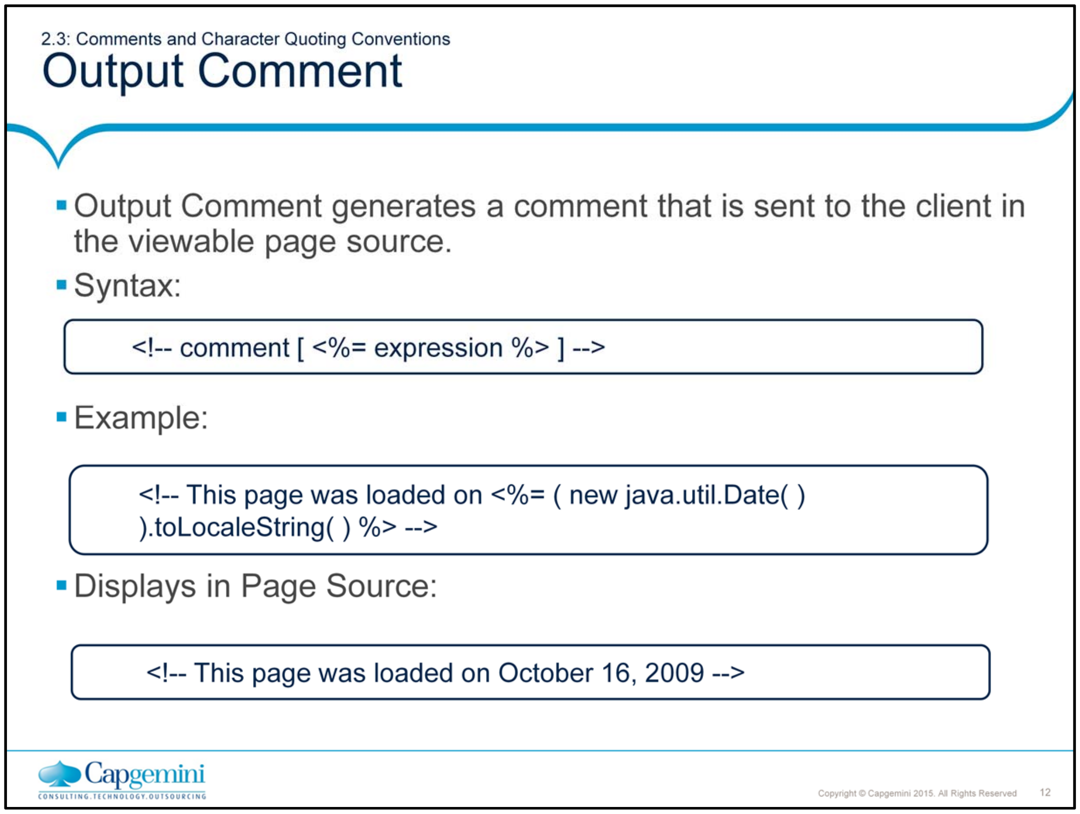
2.2: JSP Processing Model

JSP Processing in Various Scenarios

	JSP translated into Servlet	Servlet compiled	Servlet loaded into memory	jspInit called	_jspService called
When JSP page is created					
Request 1	Yes	Yes	Yes	Yes	Yes
Request 2	**No**	**No**	**No**	**No**	Yes
Server is restarted without JSP code modification					
Request 3	No	No	Yes	Yes	Yes
Request 4	**No**	**No**	No	No	Yes
JSP code is modified / Server is restarted after JSP code modification					
Request 5	Yes	Yes	Yes	Yes	Yes

JSP Processing in Various Scenarios:

- The table shown on the above slide gives some common scenarios and tells us how a JSP is processed in each of these scenarios.
- Note that the servlets resulting from JSP pages use the **_jspService** method and not **doGet()** or **doPost()** methods.



2.3: Comments and Character Quoting Conventions

Output Comment

- Output Comment generates a comment that is sent to the client in the viewable page source.
- Syntax:

```
<!-- comment [ <%= expression %> ] -->
```

- Example:

```
<!-- This page was loaded on <%= ( new java.util.Date( )
).toLocaleString( ) %> -->
```

- Displays in Page Source:

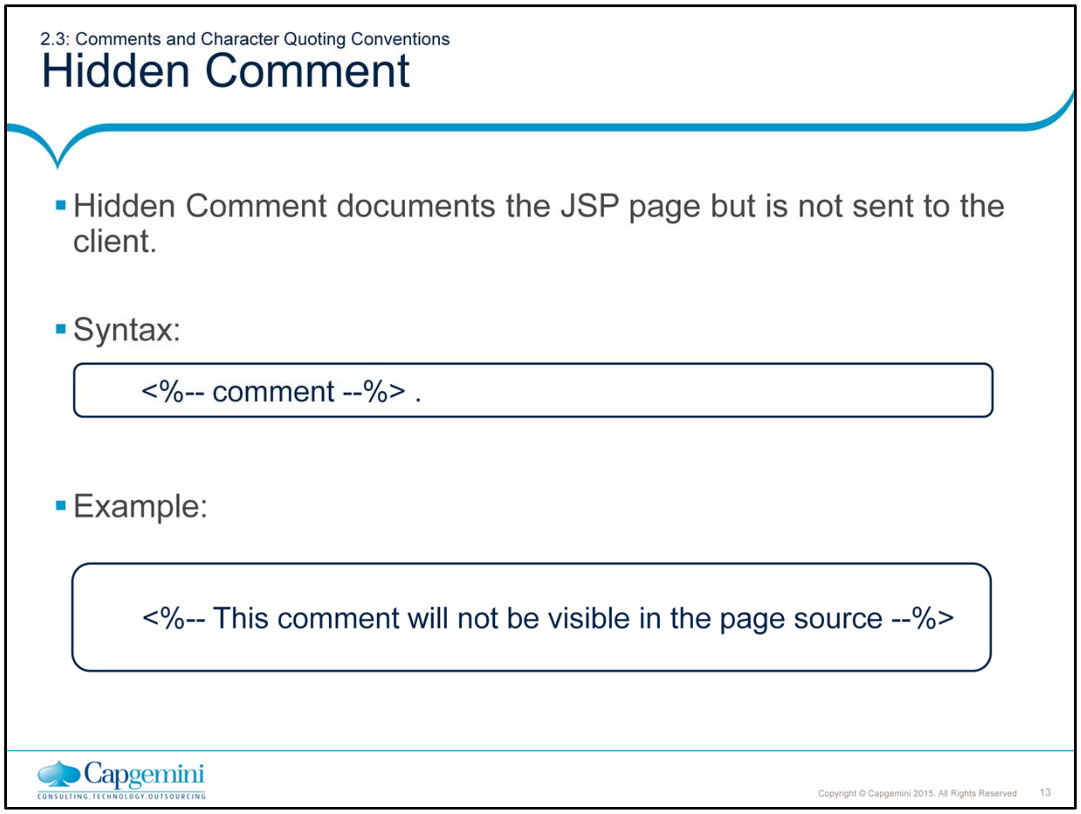
```
<!-- This page was loaded on October 16, 2009 -->
```

Capgemini CONSULTING.TECHNOLOGY.OUTSOURCING

12

<u>Comments and Character Quoting Conventions</u>:
Output Comment:

- The **JSP engine** handles an **output comment** as **non-interpreted HTML text**, returning the comment in the HTML output sent to the client. User can see the comment by viewing the page source from Web browser.
- An **Expression** included in a comment is dynamic. It is evaluated when the Web browser loads the page (that is, when the user first loads the page or reloads it later). So, any valid JSP expression can be used here.



2.3: Comments and Character Quoting Conventions

Hidden Comment

- Hidden Comment documents the JSP page but is not sent to the client.
- Syntax:

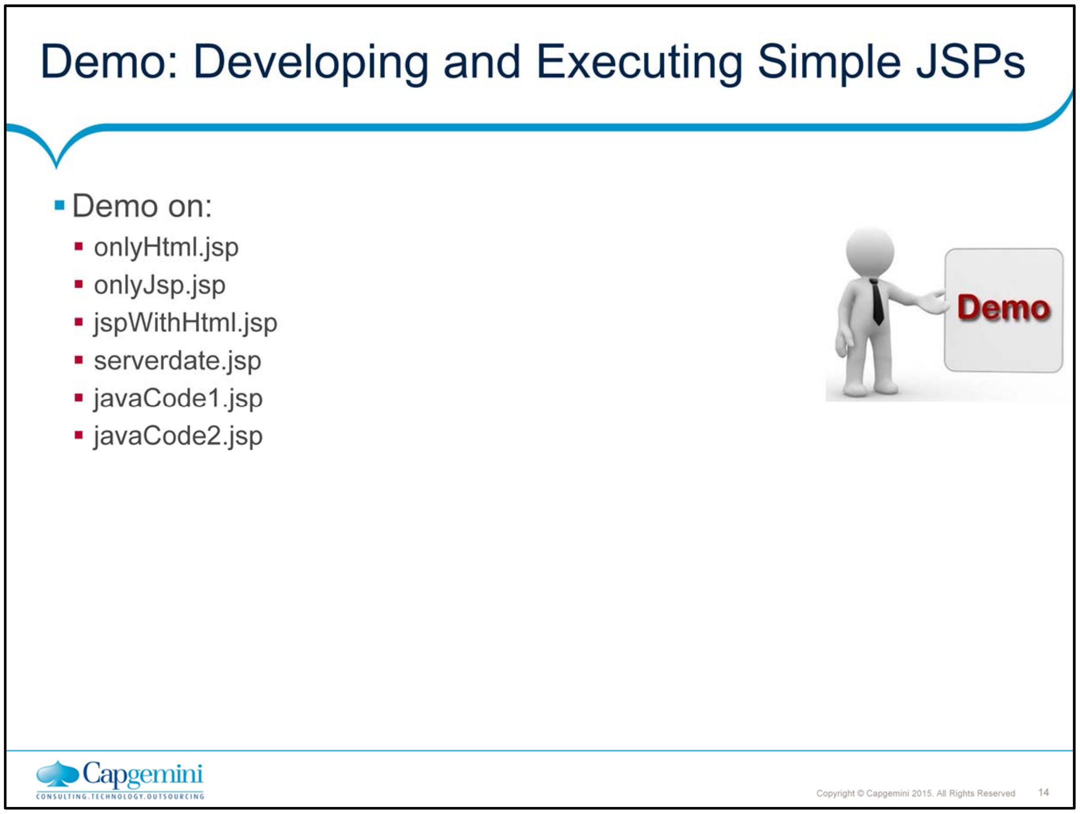
```
<%-- comment --%> .
```

- Example:

```
<%-- This comment will not be visible in the page source --%>
```

<u>Comments and Character Quoting Conventions</u>:
Hidden Comment:

- The JSP engine ignores a hidden comment, and does not process any code within hidden comment tags. A hidden comment is not sent to the client, either in the displayed JSP page or the HTML page source. The hidden comment is useful when we want to hide or "comment out" part of JSP page.
- Any characters in the body of the comment except the closing --%> combination can be used. If --%> is used in comment, it can be escaped by typing --%\>.

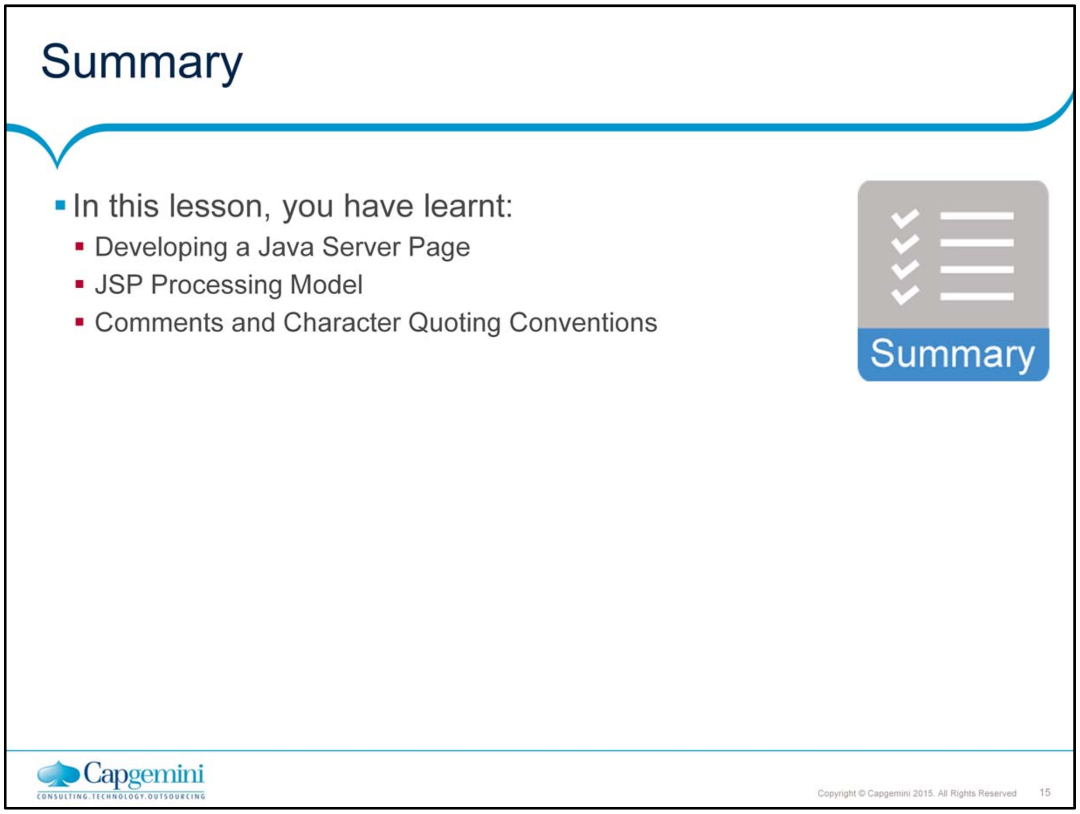


Demo: Developing and Executing Simple JSPs

- Demo on:
  - onlyHtml.jsp
  - onlyJsp.jsp
  - jspWithHtml.jsp
  - serverdate.jsp
  - javaCode1.jsp
  - javaCode2.jsp

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

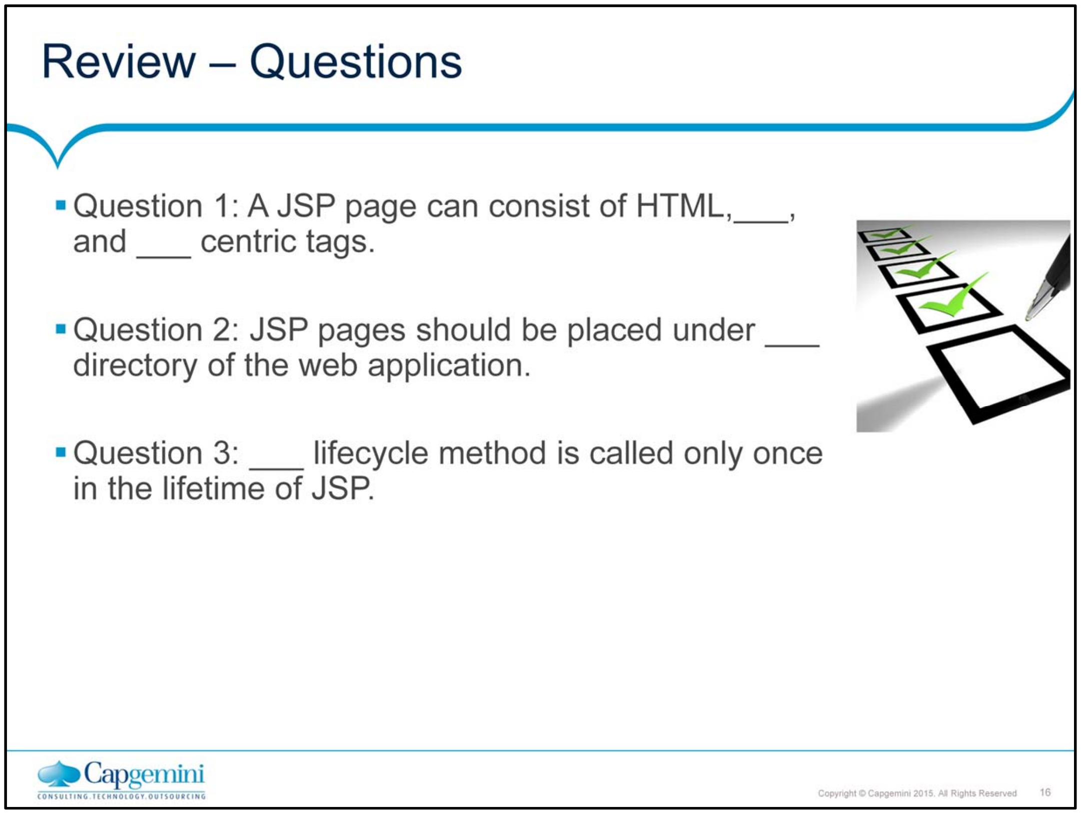
Deploy web application **Lesson2-JSPIntroduction** and show demo by executing each of the above JSP pages.



Summary

- In this lesson, you have learnt:
  - Developing a Java Server Page
  - JSP Processing Model
  - Comments and Character Quoting Conventions

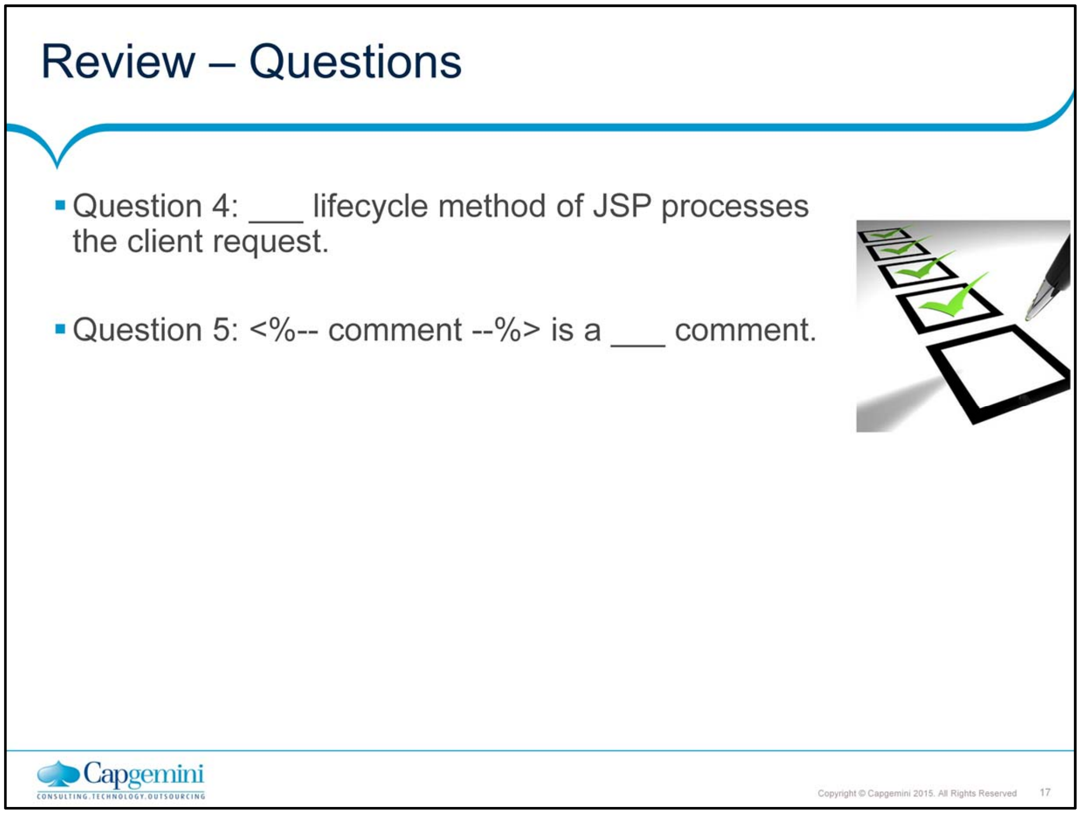
15



Review – Questions

- Question 1: A JSP page can consist of HTML,___, and ___ centric tags.
- Question 2: JSP pages should be placed under ___ directory of the web application.
- Question 3: ___ lifecycle method is called only once in the lifetime of JSP.

Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING



Review – Questions

- Question 4: ___ lifecycle method of JSP processes the client request.
- Question 5: <%-- comment --%> is a ___ comment.