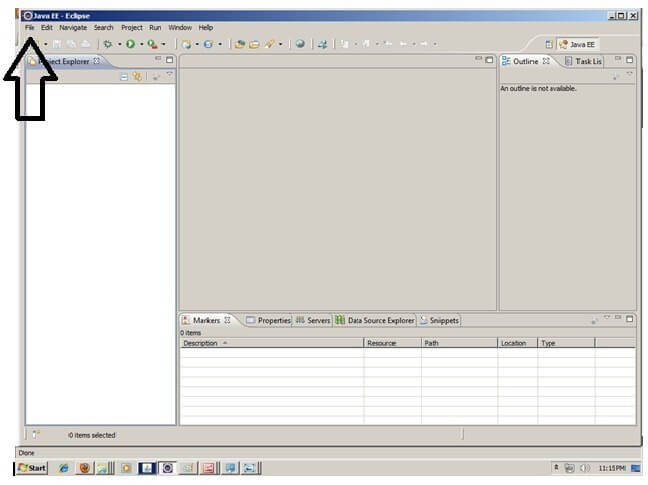
Creating JSP in Eclipse IDE with Tomcat server

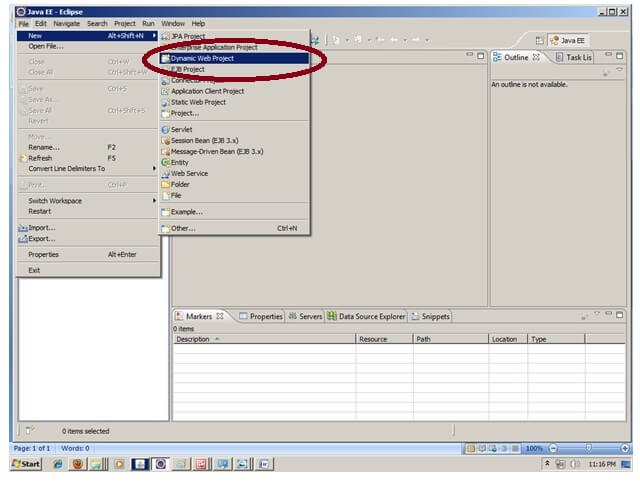
1. [Creating JSP in Eclipse IDE with Tomcat](https://www.javatpoint.com/creating-jsp-in-eclipse-ide)
   1. [Create a Dynamic web project](https://www.javatpoint.com/creating-jsp-in-eclipse-ide#step1)
   2. [create a jsp](https://www.javatpoint.com/creating-jsp-in-eclipse-ide#step2)
   3. [start tomcat server and deploy the project](https://www.javatpoint.com/creating-jsp-in-eclipse-ide#step3)

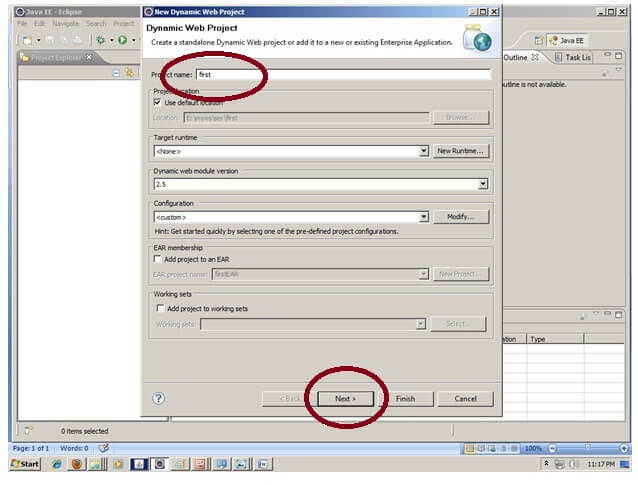
* Create a Dynamic web project
* create a jsp
* start tomcat server and deploy the project

1) Create the dynamic web project

For creating a dynamic web project click on File Menu -> New -> dynamic web project -> write your project name e.g. first -> Finish.



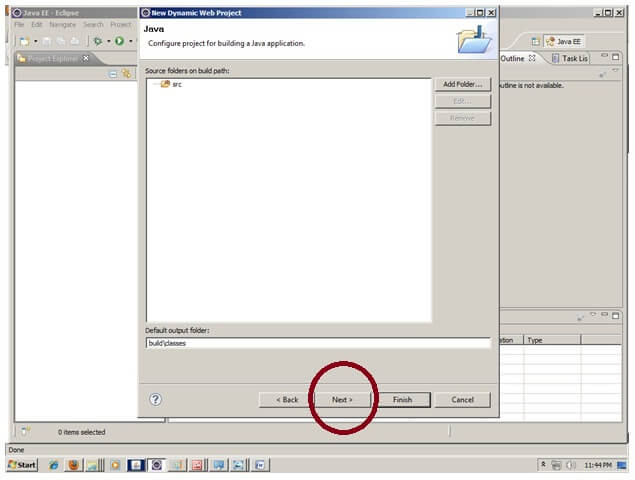


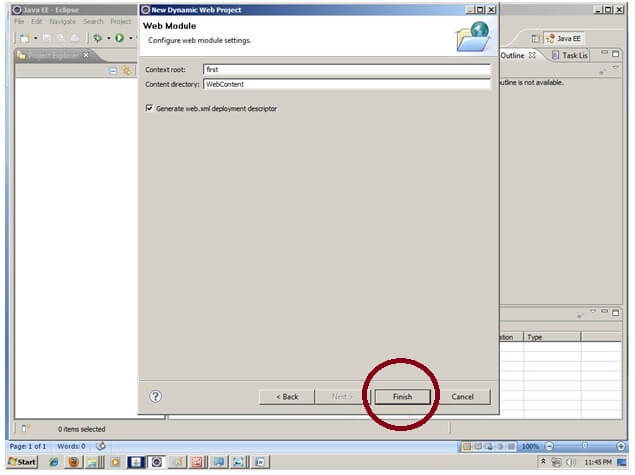


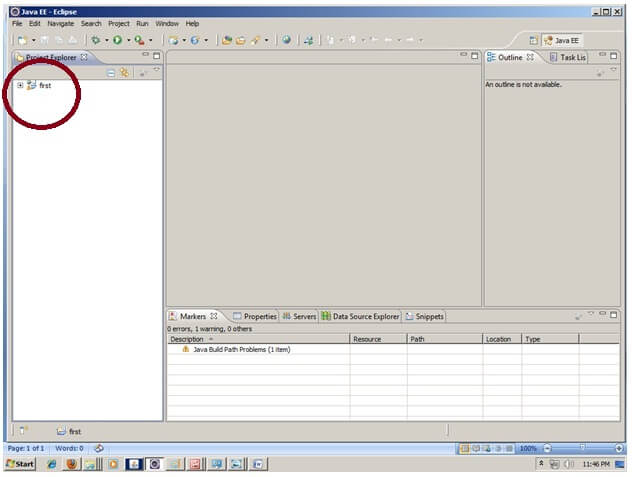
24.3M

334

Triggers in SQL (Hindi)

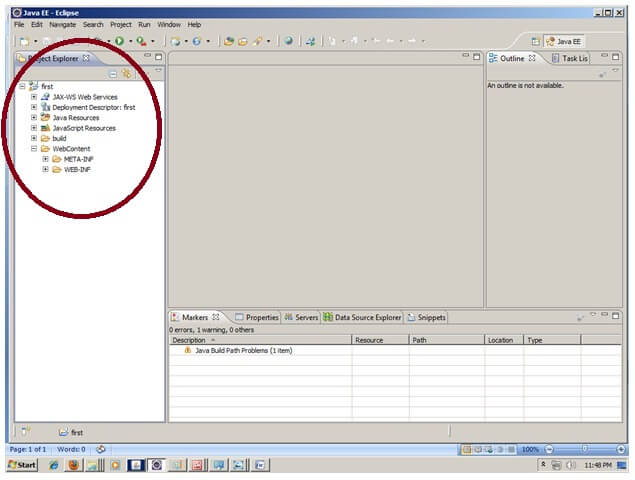


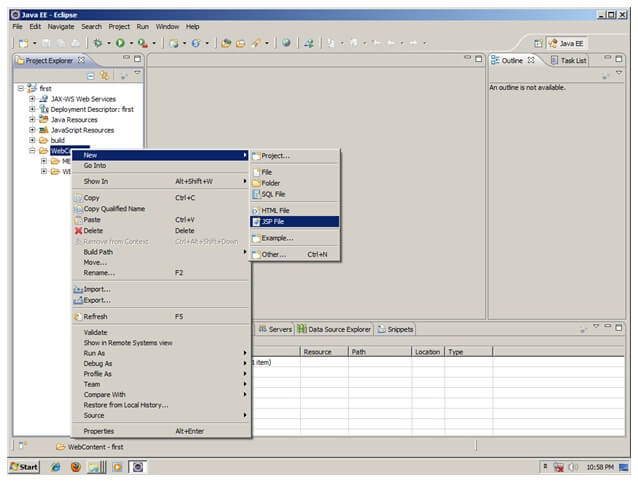


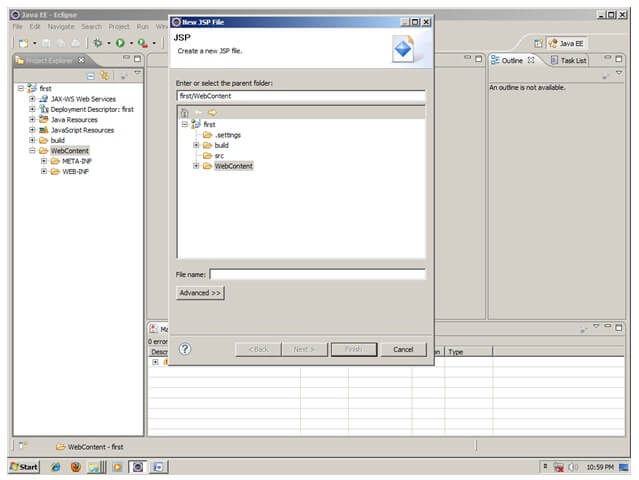


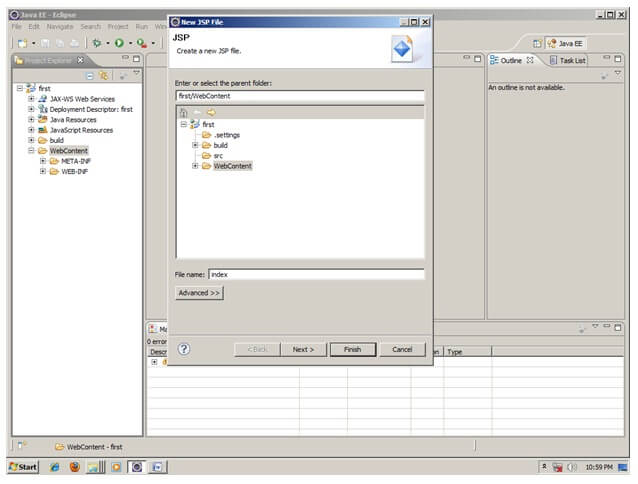
2) Create the JSP file in eclipse IDE

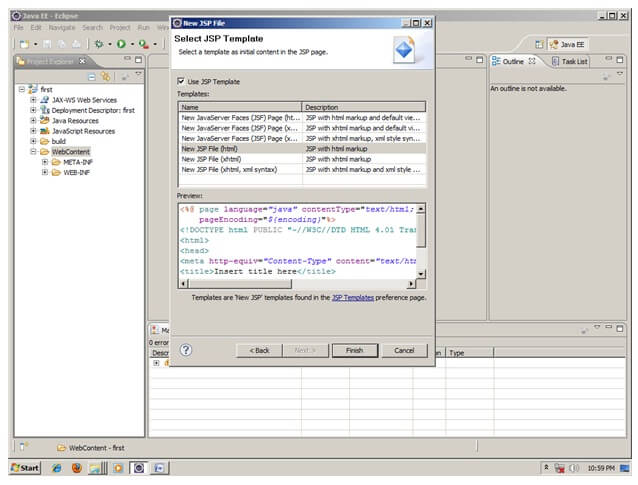
For creating a jsp file explore the project by clicking the + icon -> right click on WebContent -> New -> jsp -> write your jsp file name e.g. index -> next -> Finish.

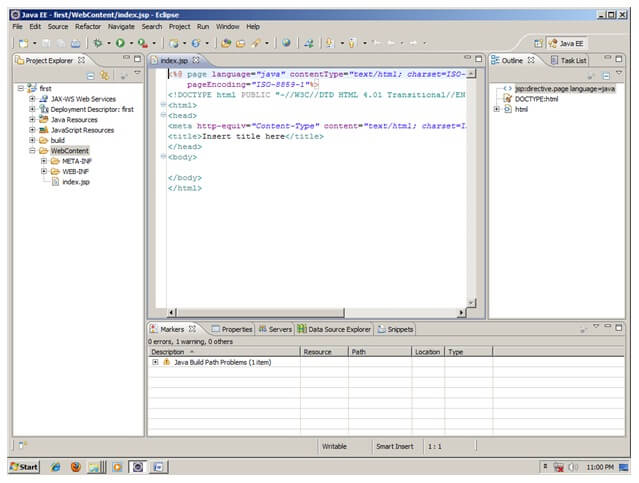




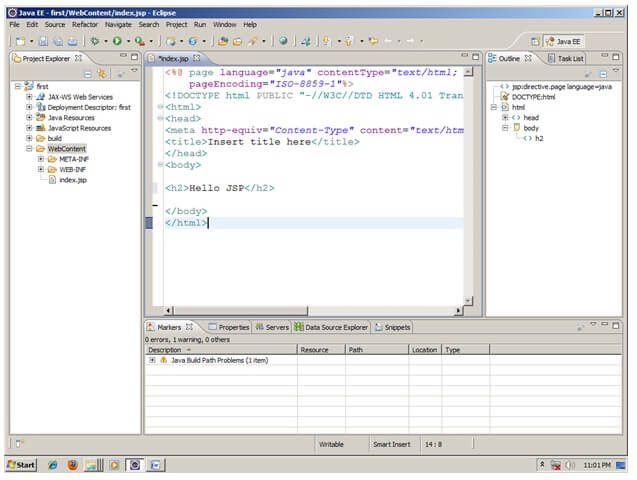








Now JSP file is created, let's write some code.



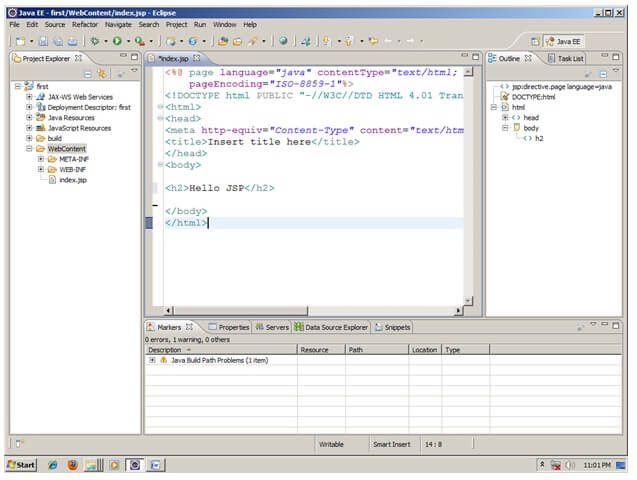
3) Start the server and deploy the project:

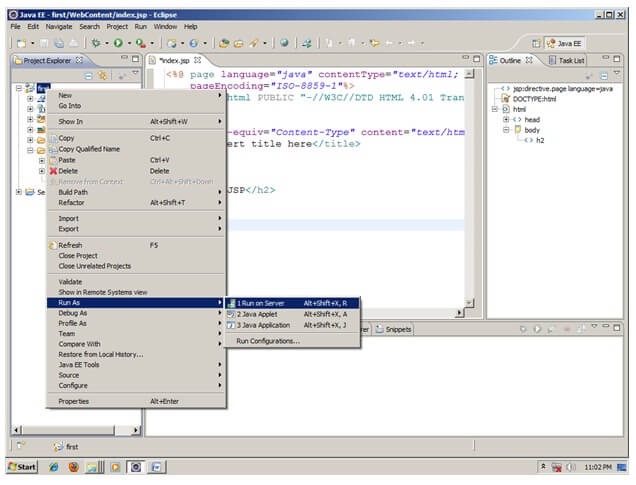
For starting the server and deploying the project in one step Right click on your project -> Run As -> Run on Server -> choose tomcat server -> next -> addAll -> finish.

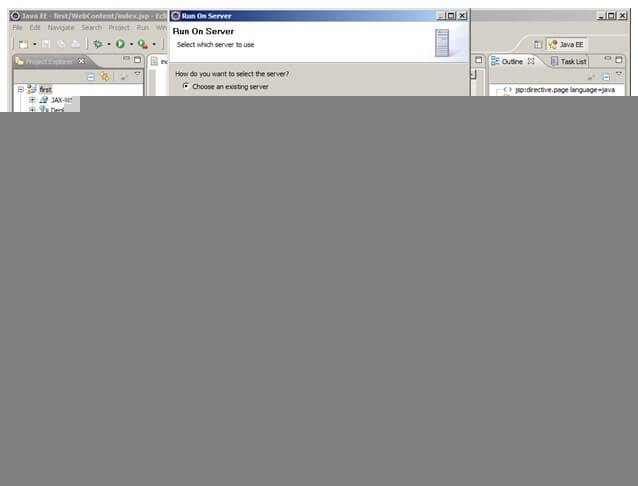
If you are using Eclipse IDE first time, you need to configure the tomcat server First. Click for [How to configure tomcat server in eclipse IDE](https://www.javatpoint.com/how-to-configure-tomcat-server-in-eclipse-ide)

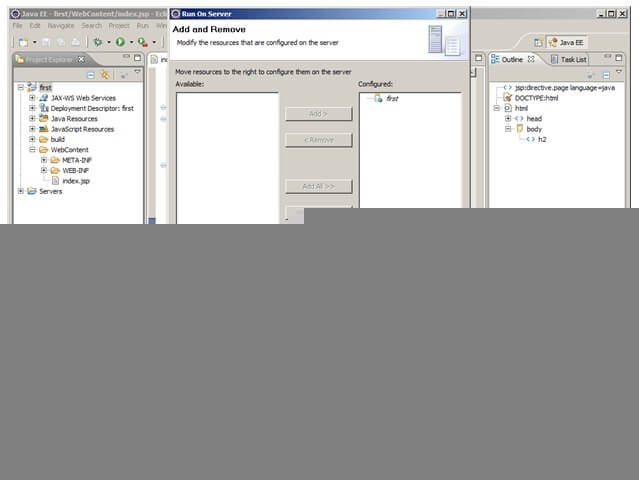
Now start the tomcat server and deploy project

For starting the server and deploying the project in one step Right click on your project -> Run As -> Run on Server -> choose tomcat server -> next -> addAll -> finish.

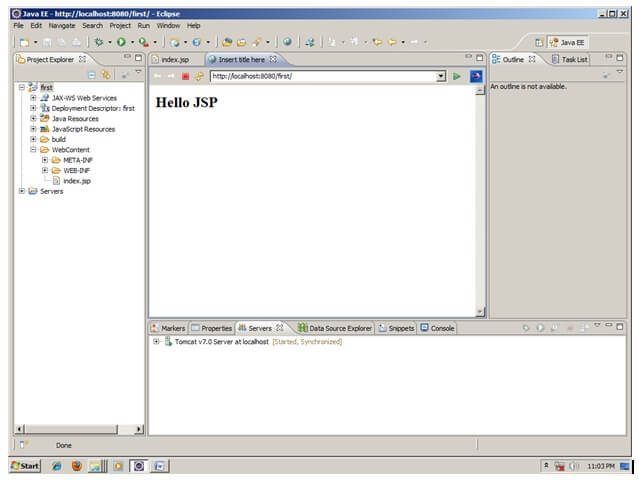








Yes, Let's see JSP is successfully running now.



# JSP Scriptlet tag (Scripting elements)

1. [Scripting elements](https://www.javatpoint.com/jsp-scriptlet-tag)
2. [JSP scriptlet tag](https://www.javatpoint.com/jsp-scriptlet-tag#scriptlet)
3. [Simple Example of JSP scriptlet tag](https://www.javatpoint.com/jsp-scriptlet-tag#scriptletex1)
4. [Example of JSP scriptlet tag that prints the user name](https://www.javatpoint.com/jsp-scriptlet-tag#scriptletex2)

In JSP, java code can be written inside the jsp page using the scriptlet tag. Let's see what are the scripting elements first.

## JSP Scripting elements

The scripting elements provides the ability to insert java code inside the jsp. There are three types of scripting elements:

* scriptlet tag
* expression tag
* declaration tag

### JSP scriptlet tag

A scriptlet tag is used to execute java source code in JSP. Syntax is as follows:

1. <%  java source code %>

### Example of JSP scriptlet tag

In this example, we are displaying a welcome message.

1. **<html>**
2. **<body>**
3. **<**% out.print("welcome to jsp"); %**>**
4. **</body>**
5. **</html>**

### Example of JSP scriptlet tag that prints the user name

In this example, we have created two files index.html and welcome.jsp. The index.html file gets the username from the user and the welcome.jsp file prints the username with the welcome message.

*File: index.html*

1. **<html>**
2. **<body>**
3. **<form** action="welcome.jsp"**>**
4. **<input** type="text" name="uname"**>**
5. **<input** type="submit" value="go"**><br/>**
6. **</form>**
7. **</body>**
8. **</html>**

*File: welcome.jsp*

1. <html>
2. <body>
3. <%
4. String name=request.getParameter("uname");
5. out.print("welcome "+name);
6. %>
7. </form>
8. </body>
9. </html>

# JSP expression tag

The code placed within **JSP expression tag** is written to the output stream of the response. So you need not write out.print() to write data. It is mainly used to print the values of variable or method.

### Syntax of JSP expression tag

1. **<**%=  statement %**>**

### Example of JSP expression tag

In this example of jsp expression tag, we are simply displaying a welcome message.

1. **<html>**
2. **<body>**
3. **<**%= "welcome to jsp" %**>**
4. **</body>**
5. **</html>**

#### Note: Do not end your statement with semicolon in case of expression tag.

### Example of JSP expression tag that prints current time

To display the current time, we have used the getTime() method of Calendar class. The getTime() is an instance method of Calendar class, so we have called it after getting the instance of Calendar class by the getInstance() method.

*index.jsp*

1. **<html>**
2. **<body>**
3. Current Time: **<**%= java.util.Calendar.getInstance().getTime() %**>**
4. **</body>**
5. **</html>**

### Example of JSP expression tag that prints the user name

In this example, we are printing the username using the expression tag. The index.html file gets the username and sends the request to the welcome.jsp file, which displays the username.

*File: index.jsp*

1. **<html>**
2. **<body>**
3. **<form** action="welcome.jsp"**>**
4. **<input** type="text" name="uname"**><br/>**
5. **<input** type="submit" value="go"**>**
6. **</form>**
7. **</body>**
8. **</html>**

*File: welcome.jsp*

1. **<html>**
2. **<body>**
3. **<**%= "Welcome "+request.getParameter("uname") %**>**
4. **</body>**
5. **</html>**

# JSP Declaration Tag

1. [JSP declaration tag](https://www.javatpoint.com/jsp-declaration-tag)
2. [Difference between JSP scriptlet tag and JSP declaration tag](https://www.javatpoint.com/jsp-declaration-tag#diff)
3. [Example of JSP declaration tag that declares field](https://www.javatpoint.com/jsp-declaration-tag#declarationex1)
4. [Example of JSP declaration tag that declares method](https://www.javatpoint.com/jsp-declaration-tag#declarationex2)

The **JSP declaration tag** is used to declare fields and methods.

The code written inside the jsp declaration tag is placed outside the service() method of auto generated servlet.

So it doesn't get memory at each request.

#### Syntax of JSP declaration tag

The syntax of the declaration tag is as follows:

Play Videox[](https://campaign.adpushup.com/get-started/?utm_source=banner&utm_campaign=growth_hack)

1. **<**%!  field or method declaration %**>**

### Difference between JSP Scriptlet tag and Declaration tag

|  |  |
| --- | --- |
| **Jsp Scriptlet Tag** | **Jsp Declaration Tag** |
| The jsp scriptlet tag can only declare variables not methods. | The jsp declaration tag can declare variables as well as methods. |
| The declaration of scriptlet tag is placed inside the \_jspService() method. | The declaration of jsp declaration tag is placed outside the \_jspService() method. |

### Example of JSP declaration tag that declares field

In this example of JSP declaration tag, we are declaring the field and printing the value of the declared field using the jsp expression tag.

### index.jsp

1. **<html>**
2. **<body>**
3. **<**%! int data=50; %**>**
4. **<**%= "Value of the variable is:"+data %**>**
5. **</body>**
6. **</html>**

### Example of JSP declaration tag that declares method

In this example of JSP declaration tag, we are defining the method which returns the cube of given number and calling this method from the jsp expression tag. But we can also use jsp scriptlet tag to call the declared method.

### index.jsp

1. **<html>**
2. **<body>**
3. **<**%!
4. int cube(int n){
5. return n\*n\*n\*;
6. }
7. %**>**
8. **<**%= "Cube of 3 is:"+cube(3) %**>**
9. **</body>**
10. **</html>**

JSP Implicit Objects

1. [JSP Implicit Objects](https://www.javatpoint.com/jsp-implicit-objects)
2. [out implicit object](https://www.javatpoint.com/jsp-implicit-objects#out)
3. [Example of out implicit object](https://www.javatpoint.com/jsp-implicit-objects#outex)

There are **9 jsp implicit objects**. These objects are *created by the web container* that are available to all the jsp pages.

The available implicit objects are out, request, config, session, application etc.

A list of the 9 implicit objects is given below:

|  |  |
| --- | --- |
| **Object** | **Type** |
| out | JspWriter |
| request | HttpServletRequest |
| response | HttpServletResponse |
| config | ServletConfig |
| application | ServletContext |
| session | HttpSession |
| pageContext | PageContext |
| page | Object |
| exception | Throwable |

# JSP directives

1. [JSP directives](https://www.javatpoint.com/jsp-page-directive)
   1. [page directive](https://www.javatpoint.com/jsp-page-directive#page)
   2. [Attributes of page directive](https://www.javatpoint.com/jsp-page-directive#pageattr)

The **jsp directives** are messages that tells the web container how to translate a JSP page into the corresponding servlet.

There are three types of directives:

* page directive
* include directive
* taglib directive

### Syntax of JSP Directive

1. <%@ directive attribute="value" %>

### JSP page directive

The page directive defines attributes that apply to an entire JSP page.

### Syntax of JSP page directive

1. <%@ page attribute="value" %>

### Attributes of JSP page directive

* import
* contentType
* extends
* info
* buffer
* language
* isELIgnored
* isThreadSafe
* autoFlush
* session
* pageEncoding
* errorPage
* isErrorPage

### 1)import

|  |
| --- |
| The import attribute is used to import class,interface or all the members of a package.It is similar to import keyword in java class or interface. |

### Example of import attribute

1. <html>
2. <body>
4. <%@ page **import**="java.util.Date" %>
5. Today is: <%= **new** Date() %>
7. </body>
8. </html>

### 2)contentType

The contentType attribute defines the MIME(Multipurpose Internet Mail Extension) type of the HTTP response.The default value is "text/html;charset=ISO-8859-1".

Play Videox[](https://campaign.adpushup.com/get-started/?utm_source=banner&utm_campaign=growth_hack)

### Example of contentType attribute

1. <html>
2. <body>
4. <%@ page contentType=application/msword %>
5. Today is: <%= **new** java.util.Date() %>
7. </body>
8. </html>

### 3)extends

The extends attribute defines the parent class that will be inherited by the generated servlet.It is rarely used.

### 4)info

This attribute simply sets the information of the JSP page which is retrieved later by using getServletInfo() method of Servlet interface.

### Example of info attribute

1. <html>
2. <body>
4. <%@ page info="composed by Sonoo Jaiswal" %>
5. Today is: <%= **new** java.util.Date() %>
7. </body>
8. </html>

The web container will create a method getServletInfo() in the resulting servlet.For example:

1. **public** String getServletInfo() {
2. **return** "composed by Sonoo Jaiswal";
3. }

### 5)buffer

The buffer attribute sets the buffer size in kilobytes to handle output generated by the JSP page.The default size of the buffer is 8Kb.

### Example of buffer attribute

1. <html>
2. <body>
4. <%@ page buffer="16kb" %>
5. Today is: <%= **new** java.util.Date() %>
7. </body>
8. </html>

### 6)language

The language attribute specifies the scripting language used in the JSP page. The default value is "java".

### 7)isELIgnored

|  |
| --- |
| We can ignore the Expression Language (EL) in jsp by the isELIgnored attribute. By default its value is false i.e. Expression Language is enabled by default. We see Expression Language later. |

1. <%@ page isELIgnored="true" %>//Now EL will be ignored

### 8)isThreadSafe

|  |
| --- |
| Servlet and JSP both are multithreaded.If you want to control this behaviour of JSP page, you can use isThreadSafe attribute of page directive.The value of isThreadSafe value is true.If you make it false, the web container will serialize the multiple requests, i.e. it will wait until the JSP finishes responding to a request before passing another request to it.If you make the value of isThreadSafe attribute like: |

<%@ page isThreadSafe="false" %>

The web container in such a case, will generate the servlet as:

1. **public** **class** SimplePage\_jsp **extends** HttpJspBase
2. **implements** SingleThreadModel{
3. .......
4. }

### 9)errorPage

The errorPage attribute is used to define the error page, if exception occurs in the current page, it will be redirected to the error page.

### Example of errorPage attribute

1. //index.jsp
2. <html>
3. <body>
5. <%@ page errorPage="myerrorpage.jsp" %>
7. <%= 100/0 %>
9. </body>
10. </html>

### 10)isErrorPage

The isErrorPage attribute is used to declare that the current page is the error page.

#### Note: The exception object can only be used in the error page.

### Example of isErrorPage attribute

1. //myerrorpage.jsp
2. <html>
3. <body>
5. <%@ page isErrorPage="true" %>
7. Sorry an exception occured!<br/>
8. The exception is: <%= exception %>
10. </body>
11. </html>

Jsp Include Directive

1. [Include directive](https://www.javatpoint.com/jsp-include-directive)
2. [Advantage of Include directive](https://www.javatpoint.com/jsp-include-directive#includeadv)
3. [Example of include directive](https://www.javatpoint.com/jsp-include-directive#includeex)

The include directive is used to include the contents of any resource it may be jsp file, html file or text file. The include directive includes the original content of the included resource at page translation time (the jsp page is translated only once so it will be better to include static resource).

Advantage of Include directive

Code Reusability

Syntax of include directive

1. <%@ include file="resourceName" %>

Example of include directive

In this example, we are including the content of the header.html file. To run this example you must create an header.html file.

1. <html>
2. <body>
4. <%@ include file="header.html" %>
6. Today is: <%= java.util.Calendar.getInstance().getTime() %>
8. </body>
9. </html>

# JSP Taglib directive

1. [JSP Taglib directive](https://www.javatpoint.com/jsp-taglib-directive)
2. [Example of JSP Taglib directive](https://www.javatpoint.com/jsp-taglib-directive#taglibex)

The JSP taglib directive is used to define a tag library that defines many tags. We use the TLD (Tag Library Descriptor) file to define the tags. In the custom tag section we will use this tag so it will be better to learn it in custom tag.

#### Syntax JSP Taglib directive

1. <%@ taglib uri="uriofthetaglibrary" prefix="prefixoftaglibrary" %>

### Example of JSP Taglib directive

In this example, we are using our tag named currentDate. To use this tag we must specify the taglib directive so the container may get information about the tag.

1. <html>
2. <body>
4. <%@ taglib uri="http://www.javatpoint.com/tags" prefix="mytag" %>
6. <mytag:currentDate/>
8. </body>
9. </html>