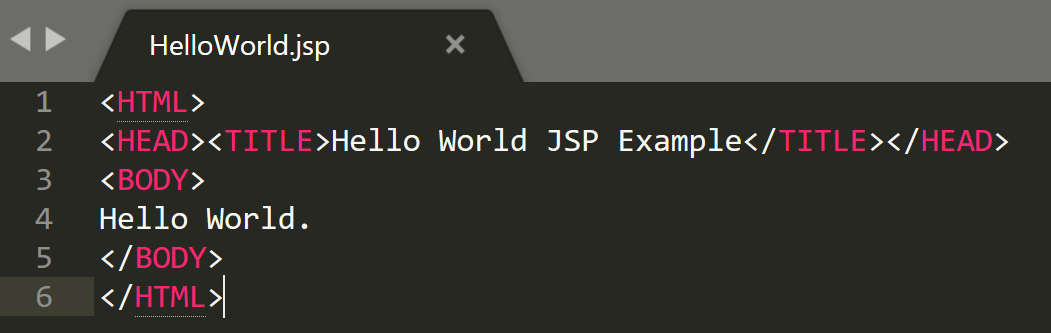
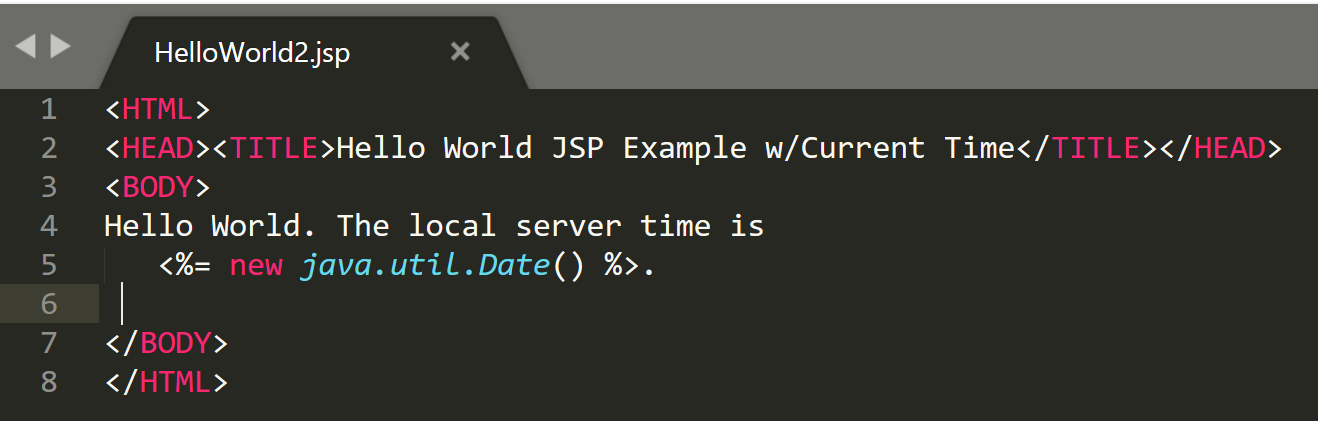
**JavaServer Pages (JSP)**

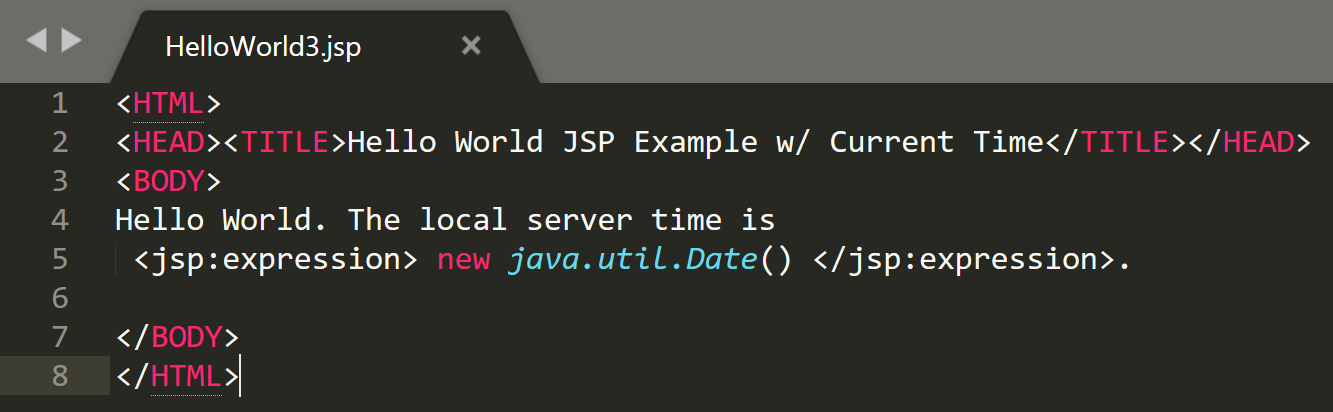
* One of the usable and dynamic tool which a developer can effectively use in developing, building, and creating content of a website is the JSP.
* It is a combination of HTML and XML, and integrates different applications of Java. Java code is attached into an HTML pages which is also an essential way of writing servlets.





**Note:** In the JSP code shown above, the server’s time is displayed which indicates that JSP code runs and executes on the server-side.

An alternative syntax for the JSP elements is by using XML tags. The same example, written using XML tags, looks like this:



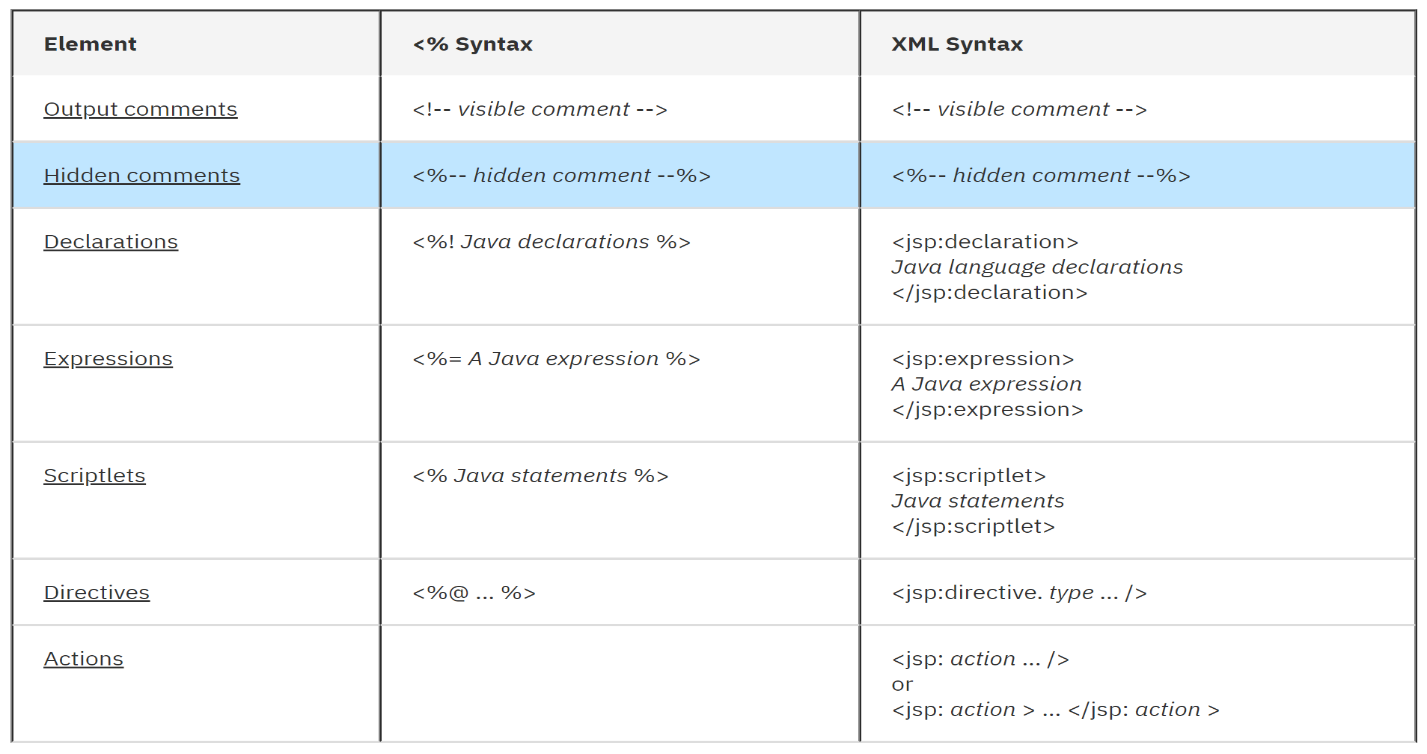
**JSP Page Interface**

* Servlet class that a page generates is a pillar of the JSP interface.
* In connection with the JSP Pages and container, JSP Page should extend Servlet which is very important.
* jspService() – primary method that handles requests generated during page translation.

**JSP Syntax**

* **JSP Syntactic elements**

The table shown below are the JSP element syntax.



* **Comments**

JSP comments can be implemented in three ways:

1. Using the HTML comment format

2. Using JSP comment format

3. scripting element comment attach to it

* **Output comments**

Standard format in writing an HTML comment:

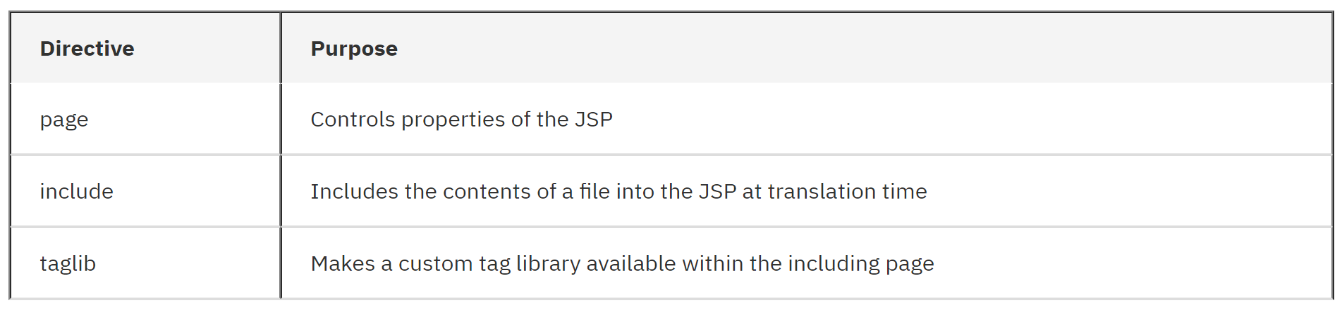
**<!-How to write an HTML comment-->**

### Hidden comments

### Standard format in writing a comment using the JSP notation:

**<%--This is a JSP notation comment--%>**

**JSP Directives**

By using the Directives, which describes the attributes within the page, it also gives a supplement information in JSP container which is only available in Tag Files.

**JSP Declarations**

* **Declarations** - Used in JSP as a new declare member of data and function which is the result of the class generated in servlet throughout the page translation. JSP can be written in two approaches:

1. <%!  Here is java declarations %>
2. <jsp:declaration>  example of java declaration </jsp:declaration>

**JSP Expressions**

1. The local server time is **<%= new java.util.Date() %>** by using this format.
2. The text **<%= new java.util.Date() %>** is an example of JSP expression.

**JSP Scriplets**

Throughout the process of time request, **scriplets** are being executed during that process. The output relies on the scriptlet code within the output stream. A sample scriplet format is shown below:

**<% java-statements %>**

**or**

|  |  |
| --- | --- |
|  | **<jsp:scriptlet>**  **java-statements**  **</jsp:scriptlet>** |
|  |  |

Again, when writing a scriplets, it is preferable to use an XML syntax/format.

**Error Handling**

By the use of JSP container as way of the JSP source to be translated into a servlet class, errors may probably happen or occur during this phase such as the following:

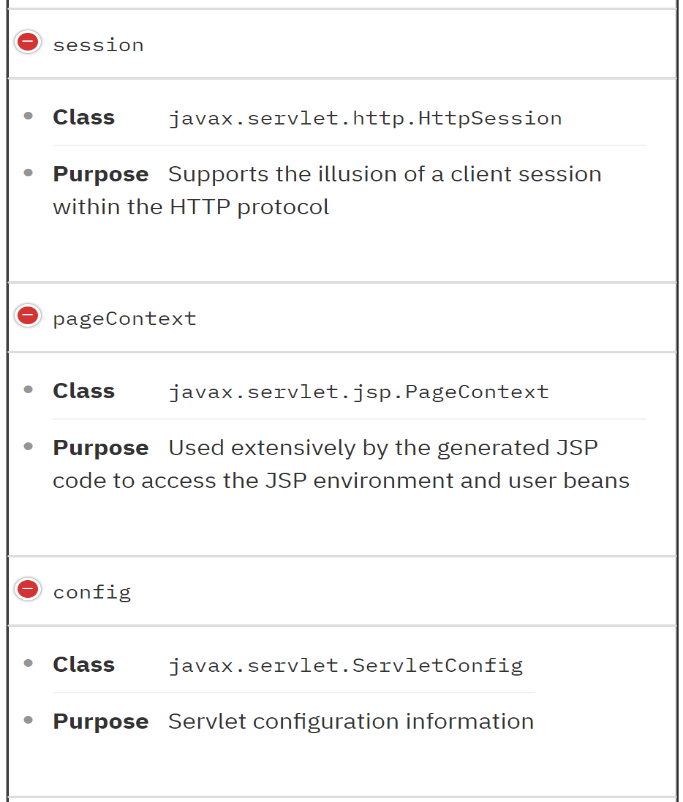
* Before the client request in the JSP, the JSP was already utilized within the JSP container
* If the JSP is requested by the client.

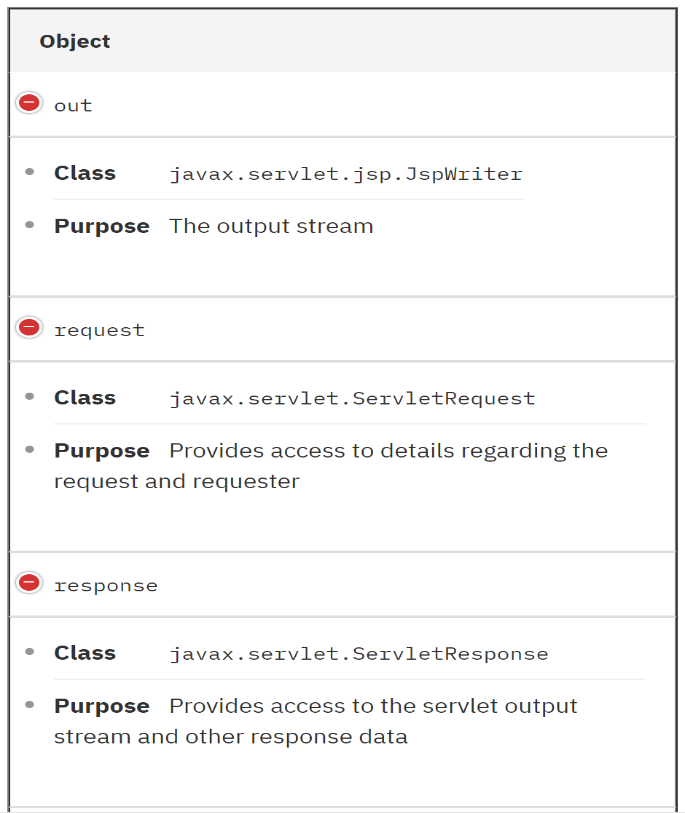
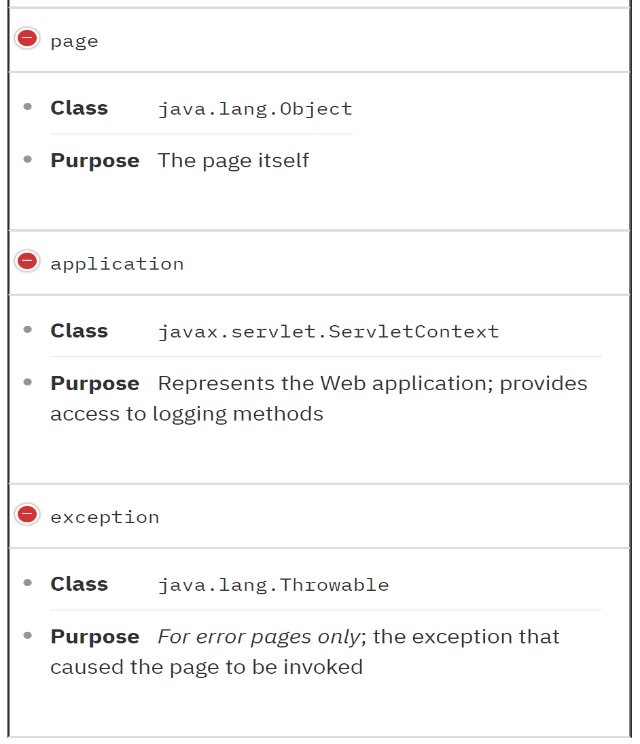
**JSP Configuration**

* **Changes in web.xml**
* **Jsp-config element**

- gives the information on global configuration among the JSP files that are within a Web application. There composed a two sub-elements of the jsp-config element:

1. **taglib** – library which implements the use of JSP such as tag mapping.
2. **jsp-property-group** - These are the groups within the JSP files.

**Implicit Objects**



**Object Scopes**

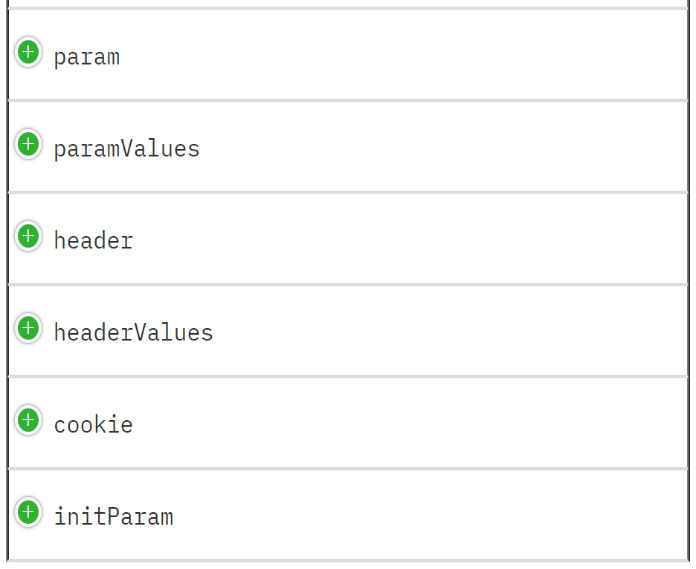
Scope refers on different entities that can be obtain by different objects respectively. Any of the following scopes can identify an object such as the following:

* **page -** Objects that are made or created within the page scope is where it is only available.
* **Request -** Objects are only available within the pages with a request scope where it is instantiated.
* **Session -** Pages where the processing of requests occurs within a similar session where it is instantiation can be accessible with objects.
* **Application -** Similar requests within the similar application where it is instantiated are accessible within the pages with application scope objects.

**Expression Language**

An expression language is usually used when assigning values to element attributes. It belongs to the specification of the first version of Java Standard Tag Library (JSTL), yet by now it belongs to the second version of JSP specification.

* **Implicit objects, EL basics**

EL expressions support several implicit objects. The table below shows and describes different Implicit Objects.

**Syntax of Expression Language**

* **[] operator**, you are able to use different properties of the JavaBeans such as its arrays of objects or lists.
* **. operator**, is the same as the [] operator for accessing different properties of the the JavaBean object.
* **Arithmetic operators** such as +, -, \*, and / are used for computations.
* **Java relational operators** such as < , > , <= , >= , == , and != are used for relational comparisons.
* As well as **Logical operators** and **Conditional operators** can be used.

**Actions**

Compared to declarations, scriplets, and expressions, **Actions** provides more functionality and also independent when it comes to scripting language.

Actions can be categorized into three standards such as the following:

1. By using the JavaBeans components
2. By controlling the run-time forwarding/including
3. By preparing HTML for the Java plug-in

**Tag Files**

* **Basic Tags**

There are two possible locations where you can place basic tags for them to be recognize by the JSP container.

1. Place the tag files within the **/META-INF/tags/** directory if the Tag Files and JSP Files is included in a JAR File.
2. If not part of the JAR files, place the tag files within the **/WEB-INF/tags/** directory.