**JAVA**

**JAVA SERVLETS**

JSP is for Java Server Page. This is, you have a HTML page, but with .jsp extension. You have too **directives**e.g. include some code from another jsp:

1. <%@include file="thisisotherjsppage.jsp" %>

And **scriptlets,** this is code fragments like:

1. <% String incrusted = "HEY!"%>
2. <!-- meanwhile in other fragment of HTML-->
3. **<span>** this could be a normal span but <%=incrusted%> it does not! **</span>**

So, you can embed Java code into HTML code (nasty by the way).

In the other hand, you have Servlets, that is, Java classes that extends from one Class **HttpServlet,**or **GenericServlet**or implement the **Servlet**inteface, AND declare your servlet in your **web.xml** configuration file, OR annotate your servlet with **WebServlet**in order to be recognized by your web container.

In your servlet, you can write HTML as well, BUT is a very big PITA because you have to do somnething like:

1. **out**.println("<html><head><title>"+thisIsMyMessage+"</title></head>");

The object **out**is an object obtained from your **response** object…

In resume, the difference is, you can HTML with embedded Java code, or you can generate your HTML code from a servlet class.

## SERVLETS

A servlet is a Java class that runs in a Java-enabled server. An HTTP servlet is a special type of servlet that handles an HTTP request and provides an HTTP response, usually in the form of an HTML page. The most common use of WebLogic HTTP servlets is to create interactive applications using standard Web browsers for the client-side presentation while WebLogic Server handles the business logic as a server-side process. WebLogic HTTP servlets can access databases, Enterprise JavaBeans, messaging APIs, HTTP sessions, and other facilities of WebLogic Server.

### SERVLETS AND JAVA EE

### What You Can Do with Servlets

* Create dynamic Web pages that use HTML forms to get end-user input and provide HTML pages that respond to that input. Examples of this utilization include online shopping carts, financial services, and personalized content.
* Create collaborative systems such as online conferencing.
* Have access to a variety of APIs and features by using servlets running in WebLogic Server. For example:
  + Session tracking—Allows a Web site to track a user's progress across multiple Web pages. This functionality supports Web sites such as e-commerce sites that use shopping carts. WebLogic Server supports session persistence to a database, providing failover between server down time and session sharing between clustered servers. For more information see [Session Tracking from a Servlet](https://docs.oracle.com/cd/E17904_01/web.1111/e13712/sessions.htm#i150257).
  + JDBC drivers—JDBC drivers provide basic database access. With WebLogic Server's multi-tier JDBC implementations, you can take advantage of connection pools, server-side data caching, and transactions. For more information see [Accessing Databases](https://docs.oracle.com/cd/E17904_01/web.1111/e13712/progservlet.htm#i159912).
  + Enterprise JavaBeans—Servlets can use Enterprise JavaBeans (EJB) to encapsulate sessions, data from databases, and other functionality. See [Referencing External EJBs](https://docs.oracle.com/cd/E17904_01/web.1111/e13712/configureresources.htm#i158935), [More about the ejb-ref\* Elements](https://docs.oracle.com/cd/E17904_01/web.1111/e13712/configureresources.htm#i158940), and [Referencing Application-Scoped EJBs](https://docs.oracle.com/cd/E17904_01/web.1111/e13712/configureresources.htm#i158945).
  + Java Messaging Service (JMS)—JMS allows your servlets to exchange messages with other servlets and Java programs. See [*Programming JMS for Oracle WebLogic Server*](https://docs.oracle.com/cd/E17904_01/web.1111/e13727/toc.htm).
  + Java JDK APIs—Servlets can use the standard Java JDK APIs.
  + Forwarding requests—Servlets can forward a request to another servlet or other resource. [Forwarding a Request](https://docs.oracle.com/cd/E17904_01/web.1111/e13712/progservlet.htm#i177843).
* Easily deploy servlets written for any Java EE-compliant servlet engine to WebLogic Server.

### Servlet Development Key Points

The following are a few key points relating to servlet development:

* Programmers of HTTP servlets utilize a standard Java API, **javax.servlet.http**, to create interactive applications.
* HTTP servlets can read HTTP headers and write HTML coding to deliver a response to a browser client.
* Servlets are deployed to WebLogic Server as part of a Web application. A Web application is a grouping of application components such as servlet classes, JavaServer Pages (JSPs), static HTML pages, images, and security.