**Review about JAVA**

*This Java Programming Cheatsheet review from the beginning to basics and popular functions that all programmers need to know :   
This included syntax and functions*

Common in use libraries

<http://introcs.cs.princeton.edu/java/11cheatsheet/>

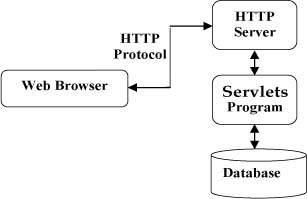
**Servlet**

Java Servlets are programs that run on a Web or Application server and act as a middle layer between a request coming from a Web browser or other HTTP client and databases or applications on the HTTP server.

Servlets perform the following major tasks:

* Read the explicit data sent by the clients (browsers). This includes an HTML form on a Web page or it could also come from an applet or a custom HTTP client program.
* Read the implicit HTTP request data sent by the clients (browsers). This includes cookies, media types and compression schemes the browser understands, and so forth.
* Process the data and generate the results. This process may require talking to a database, executing an RMI or CORBA call, invoking a Web service, or computing the response directly.
* Send the explicit data (i.e., the document) to the clients (browsers). This document can be sent in a variety of formats, including text (HTML or XML), binary (GIF images), Excel, etc.
* Send the implicit HTTP response to the clients (browsers). This includes telling the browsers or other clients what type of document is being returned (e.g., HTML), setting cookies and caching parameters, and other such tasks.

**Architecture:**



Downloading an implementation of the Java Software Development Kit (SDK) and setting up PATH environment variable.

Download SDK from: <http://java.sun.com/products/servlet/>

set PATH=C:\jdk1.5.0\_20\bin;%PATH%

set JAVA\_HOME=C:\jdk1.5.0\_20

Go to My Computer, select Properties, then Advanced, then Environment Variables. Then, you would update the PATH value and press the OK button.

Please test if your IDE know where your Java is located.

**Setting up Web Server: Tomcat**

Download: <http://tomcat.apache.org/>

Create CATALINA\_HOME environment variable pointing to setup locations.

Execute startup.bat to run Tomcat.

C:\apache-tomcat-5.5.29\bin\startup.bat

Shutdown Tomcat

C:\apache-tomcat-5.5.29\bin\shutdown

Setting up servlet CLASSPATH

Servlets are not part of the Java Platform, Standard Edition, you must identify the servlet classes to the compiler.

set CATALINA=C:\apache-tomcat-5.5.29

set CLASSPATH=%CATALINA%\common\lib\servlet-api.jar;%CLASSPATH%

Right-click on My Computer, select Properties, then Advanced, then Environment Variables. Then, you would update the CLASSPATH value and press the OK button.

**SAMPLE WORK**

HelloWorld.java

// Import required java libraries

import java.io.\*;

import javax.servlet.\*;

import javax.servlet.http.\*;

// Extend HttpServlet class

public class HelloWorld extends HttpServlet {

private String message;

public void init() throws ServletException

{

// Do required initialization

message = "Hello World";

}

public void doGet(HttpServletRequest request,

HttpServletResponse response)

throws ServletException, IOException

{

// Set response content type

response.setContentType("text/html");

// Actual logic goes here.

PrintWriter out = response.getWriter();

out.println("<h1>" + message + "</h1>");

}

public void destroy()

{

// do nothing.

}

Servlet Deployment

If you have a fully qualified class name of com.myorg.MyServlet, then this servlet class must be located in WEB-INF/classes/com/myorg/MyServlet.class.

 Create following entries in web.xml <directory>/webapps/ROOT/WEB-INF/

<servlet>

<servlet-name>HelloWorld</servlet-name>

<servlet-class>HelloWorld</servlet-class>

</servlet>

<servlet-mapping>

<servlet-name>HelloWorld</servlet-name>

<url-pattern>/HelloWorld</url-pattern>

</servlet-mapping>

Advanced Servlet:

File Upload <https://www.tutorialspoint.com/servlets/servlets-file-uploading.htm>

Handling Date <https://www.tutorialspoint.com/servlets/servlets-handling-date.htm>

Page Redirect <https://www.tutorialspoint.com/servlets/servlets-page-redirect.htm>

Hits Counter <https://www.tutorialspoint.com/servlets/servlets-hits-counter.htm>

Auto Refresh <https://www.tutorialspoint.com/servlets/servlets-auto-refresh.htm>

Send Email <https://www.tutorialspoint.com/servlets/servlets-sending-email.htm>

**Maven**

<https://www.tutorialspoint.com/maven/maven_quick_guide.htm>

Maven is a project management and comprehension tool. Maven provides developers a complete build lifecycle framework. Development team can automate the project's build infrastructure in almost no time as Maven uses a standard directory layout and a default build lifecycle.

Maven provides developers ways to manage following:

* Builds
* Documentation
* Reporting
* Dependencies
* SCMs
* Releases
* Distribution
* Mailing list

To summarize, Maven simplifies and standardizes the project build process. It handles compilation, distribution, documentation, team collaboration and other tasks seamlessly. Maven increases reusability and takes care of most of build related tasks.

Maven primary goal is to provide developer

* A comprehensive model for projects which is reusable, maintainable, and easier to comprehend.
* Plugins or tools that interact with this declarative model.

Maven project structure and contents are declared in an xml file, pom.xml referred as Project Object Model (POM), which is the fundamental unit of the entire Maven system.

Installing MAVEN

Please update to Java latest version.

Setup JAVA environment.

Download Maven <http://maven.apache.org/download.cgi>

Extract the archive, to the directory you wish to install Maven 3.3.3. The subdirectory apache-maven-3.3.3 will be created from the archive.

Add M2\_HOME, M2, MAVEN\_OPTS to environment variables

|  |  |
| --- | --- |
| Windows | Set the environment variables using system properties.   *M2\_HOME=C:\Program Files\Apache Software Foundation\apache-maven-3.3.3*  *M2=%M2\_HOME%\bin*  *MAVEN\_OPTS=-Xms256m -Xmx512m* |

Add Maven bin directory location to system path

|  |  |
| --- | --- |
| Windows | Append the string ;%M2% to the end of the system variable, Path. |

execute the following mvn command. To verify maven installation.

**Maven POM**

POM stands for Project Object Model. It is fundamental Unit of Work in Maven. It is an XML file. It always resides in the base directory of the project as pom.xml.

The POM contains information about the project and various configuration detail used by Maven to build the project(s).

Some of the configuration that can be specified in the POM are following:

* project dependencies
* plugins
* goals
* build profiles
* project version
* developers
* mailing list

# **Maven - Eclipse IDE**

[**https://www.tutorialspoint.com/maven/maven\_eclispe\_ide.htm**](https://www.tutorialspoint.com/maven/maven_eclispe_ide.htm)

**OOP concepts in java**

Inheritance

<https://www.tutorialspoint.com/java/java_inheritance.htm>

Overriding

<https://www.tutorialspoint.com/java/java_overriding.htm>

Polymorphism

<https://www.tutorialspoint.com/java/java_polymorphism.htm>

Abstraction (It is impossible to instantiate an abstract class, but, it is possible if a class inherit an abstract class)

<https://www.tutorialspoint.com/java/java_abstraction.htm>

Encapsulation

<https://www.tutorialspoint.com/java/java_encapsulation.htm>

Interface (Method inheritance)

<https://www.tutorialspoint.com/java/java_interfaces.htm>

MVC

<https://www.tutorialspoint.com/design_pattern/mvc_pattern.htm>

JDBC

<https://www.tutorialspoint.com/jdbc/jdbc-quick-guide.htm>