JUNIT - ENVIRONMENT SETUP

> ### *Try it Option Online*
>
> *You really do not need to set up your own environment to start learning Java & JUnit programming language. Reason is very simple, we already have setup Java Programming environment online, so that you can compile and execute all the available examples online at the same time when you are doing your theory work. This gives you confidence in what you are reading and to check the result with different options. Feel free to modify any example and execute it online.*
>
> *Try following example using **Try it** option available at the top right corner of the below sample code box:*

```
public class MyFirstJavaProgram {

    public static void main(String []args) {
       System.out.println("Hello World");
    }
}
```

> *For most of the examples given in this tutorial, you will find **Try it** option, so just make use of it and enjoy your learning.*

Local Environment Setup

JUnit is a framework for Java, so the very first requirement is to have JDK installed in your machine.

System Requirement

JDK	1.5 or above.
Memory	no minimum requirement.
Disk Space	no minimum requirement.
Operating System	no minimum requirement.

Step 1 - verify Java installation in your machine

Now open console and execute the following **java** command.

OS	Task	Command
Windows	Open Command Console	c:\> java -version
Linux	Open Command Terminal	$ java -version
Mac	Open Terminal	machine:~ joseph$ java -version

Let's verify the output for all the operating systems:

OS	Output

Windows	java version "1.6.0_21" JavaTM SE Runtime Environment $build1.6.0_21-b07$ Java HotSpotTM Client VM $build17.0-b17, mixedmode, sharing$
Linux	java version "1.6.0_21" JavaTM SE Runtime Environment $build1.6.0_21-b07$ Java HotSpotTM Client VM $build17.0-b17, mixedmode, sharing$
Mac	java version "1.6.0_21" JavaTM SE Runtime Environment $build1.6.0_21-b07$ Java HotSpotTM64-Bit Server VM $build17.0-b17, mixedmode, sharing$

If you do not have Java installed, install the Java Software Development Kit SDK from http://www.oracle.com/technetwork/java/javase/downloads/index.html. We are assuming Java 1.6.0_21 as installed version for this tutorial.

Step 2: Set JAVA environment

Set the **JAVA_HOME** environment variable to point to the base directory location where Java is installed on your machine. For example

OS	Output
Windows	Set the environment variable JAVA_HOME to C:\Program Files\Java\jdk1.6.0_21
Linux	export JAVA_HOME=/usr/local/java-current
Mac	export JAVA_HOME=/Library/Java/Home

Append Java compiler location to System Path.

OS	Output
Windows	Append the string ;C:\Program Files\Java\jdk1.6.0_21\bin to the end of the system variable, Path.
Linux	export PATH=$PATH$:JAVA_HOME/bin/
Mac	not required

Verify Java Installation using **java -version** command explained above.

Step 3: Download Junit archive

Download latest version of JUnit jar file from http://www.junit.org. At the time of writing this tutorial, I downloaded *Junit-4.10.jar* and copied it into C:\>JUnit folder.

OS	Archive name

Windows	junit4.10.jar
Linux	junit4.10.jar
Mac	junit4.10.jar

Step 4: Set JUnit environment

Set the **JUNIT_HOME** environment variable to point to the base directory location where JUNIT jar is stored on your machine. Assuming, we've stored junit4.10.jar in JUNIT folder on various Operating Systems as follows.

OS	Output
Windows	Set the environment variable JUNIT_HOME to C:\JUNIT
Linux	export JUNIT_HOME=/usr/local/JUNIT
Mac	export JUNIT_HOME=/Library/JUNIT

Step 5: Set CLASSPATH variable

Set the **CLASSPATH** environment variable to point to the JUNIT jar location. Assuming, we've stored junit4.10.jar in JUNIT folder on various Operating Systems as follows.

OS	Output
Windows	Set the environment variable CLASSPATH to %CLASSPATH%;%JUNIT_HOME%\junit4.10.jar;.;
Linux	export CLASSPATH=*CLASSPATH*:JUNIT_HOME/junit4.10.jar:.
Mac	export CLASSPATH=*CLASSPATH*:JUNIT_HOME/junit4.10.jar:.

Step 6: Test JUnit Setup

Create a java class file name TestJunit in **C:\ > JUNIT_WORKSPACE**

```
import org.junit.Test;
import static org.junit.Assert.assertEquals;
public class TestJunit {
   @Test
   public void testAdd() {
      String str= "Junit is working fine";
      assertEquals("Junit is working fine",str);
   }
}
```

Create a java class file name TestRunner in **C:\ > JUNIT_WORKSPACE** to execute Test case*s*

```
import org.junit.runner.JUnitCore;
import org.junit.runner.Result;
import org.junit.runner.notification.Failure;

public class TestRunner {
   public static void main(String[] args) {
      Result result = JUnitCore.runClasses(TestJunit.class);
      for (Failure failure : result.getFailures()) {
         System.out.println(failure.toString());
```

```
        }
        System.out.println(result.wasSuccessful());
    }
}
```

Step 7: Verify the Result

Compile the classes using **javac** compiler as follows

```
C:\JUNIT_WORKSPACE>javac TestJunit.java TestRunner.java
```

Now run the Test Runner to see the result

```
C:\JUNIT_WORKSPACE>java TestRunner
```

Verify the output.

```
true
```