

Step I. 下載 java jdk


▲ <http://www.oracle.com/technetwork/java/javase/downloads/index.html>


Oracle Technology Network > Java > Java SE > Downloads

Overview Downloads Documentation Community Technologies Training

Java SE Downloads

Next Releases (Early Access) Embedded Use Previous Releases

Java SE 7u51

[DOWNLOAD](#)

NetBeans

[DOWNLOAD](#)

JDK 7u51 & NetBeans 7.4

Java Platform, Standard Edition

Java SE 7u51
 This release includes important security fixes. Oracle strongly recommends that all Java SE 7 users upgrade to this release.

Java SDKs and Tools

- Java SE
- Java EE and Glassfish
- Java ME
- JavaFX
- Java Card
- NetBeans IDE
- Java Mission Control

Java Resources

- Java APIs
- Technical Articles
- Demos and Videos
- Forums
- Java Magazine
- Java.net
- Developer Training
- Tutorials

▲ Click “Accept License Agreement”

Java SE Development Kit 7u51

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

☒ Accept License Agreement ☐ Decline License Agreement

Product / File Description	File Size	Download
Linux ARM v6/v7 Hard Float ABI	67.7 MB	jdk-7u51-linux-arm-vfp-hflt.tar.gz
Linux ARM v6/v7 Soft Float ABI	67.68 MB	jdk-7u51-linux-arm-vfp-sflt.tar.gz
Linux x86	115.65 MB	jdk-7u51-linux-i586.rpm
Linux x86	132.98 MB	jdk-7u51-linux-i586.tar.gz
Linux x64	116.96 MB	jdk-7u51-linux-x64.rpm
Linux x64	131.8 MB	jdk-7u51-linux-x64.tar.gz
Mac OS X x64	179.49 MB	jdk-7u51-macosx-x64.dmg
Solaris x86 (SVR4 package)	140.02 MB	jdk-7u51-solaris-i586.tar.Z
Solaris x86	95.13 MB	jdk-7u51-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	24.53 MB	jdk-7u51-solaris-x64.tar.Z
Solaris x64	16.28 MB	jdk-7u51-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	139.39 MB	jdk-7u51-solaris-sparc.tar.Z
Solaris SPARC	98.19 MB	jdk-7u51-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	23.94 MB	jdk-7u51-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	18.33 MB	jdk-7u51-solaris-sparcv9.tar.gz
Windows x86	123.64 MB	jdk-7u51-windows-i586.exe
Windows x64	125.46 MB	jdk-7u51-windows-x64.exe


Java SE Development Kit 7u51 Demos and Samples Downloads

▲ choice your OS for example window x64

Solaris SPARC 64-bit (SVR4 package)	23.94 MB	jdk-7u51-solaris-sparcv9.tar.gz
Solaris SPARC 64-bit	18.33 MB	jdk-7u51-solaris-sparcv9.tar.gz
Windows x86	123.64 MB	jdk-7u51-windows-i586.exe
Windows x64	125.46 MB	jdk-7u51-windows-x64.exe

If you don't know your Windows' version, check the link below.

<http://support.microsoft.com/kb/827218/zh-tw>

 自動版本偵測結果

您目前使用的是 64 位元的作業系統。

[↑ 回此頁最上方](#) | [提供意見](#)

Step II. install java jdk

就一直按

但請記得安裝的位置，然後找到 bin file

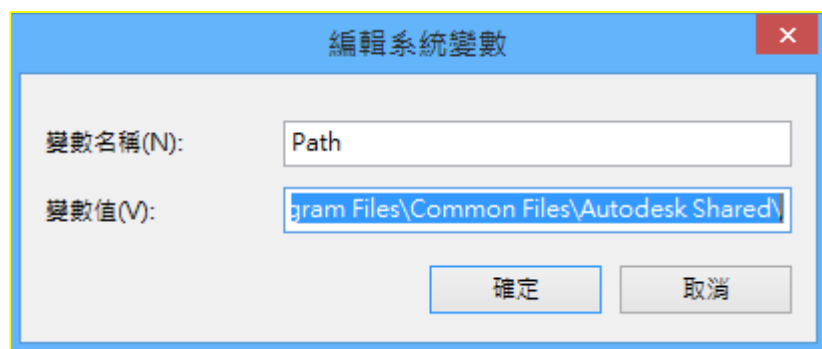
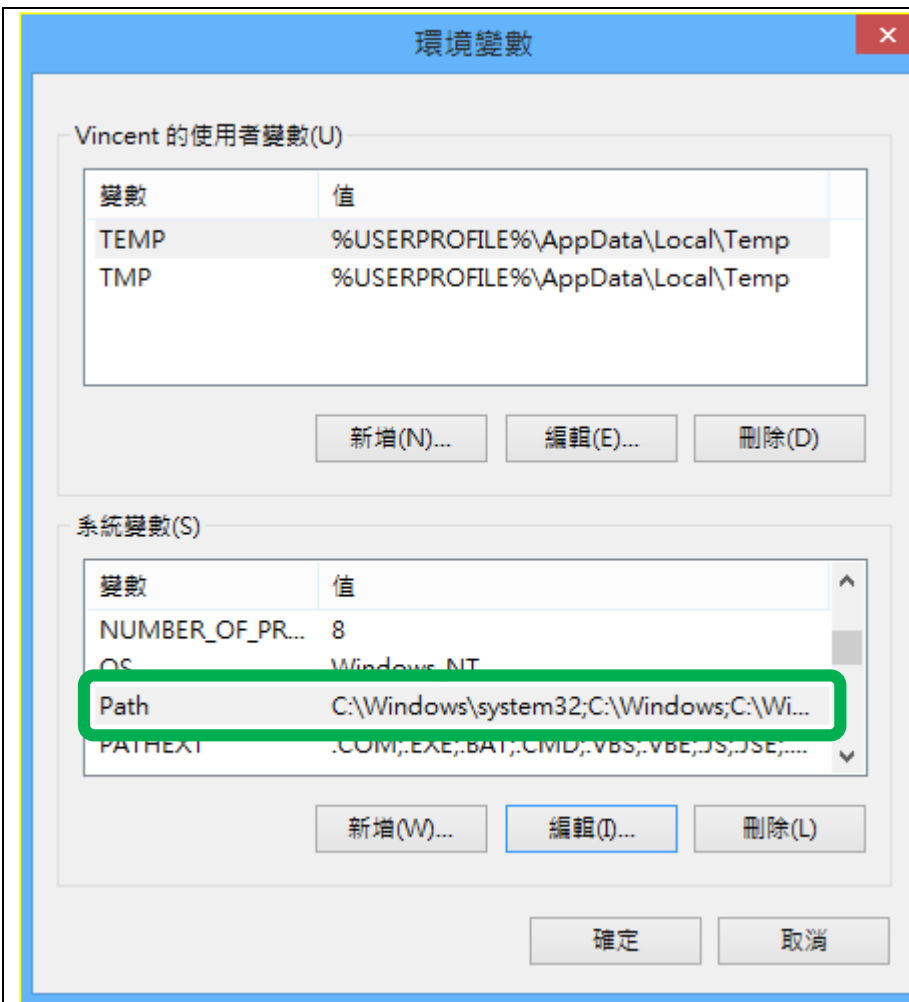
如 D:\Program Files\Java\jdk1.7.0_51\bin

Step III. Setting path

從控制台找到系統

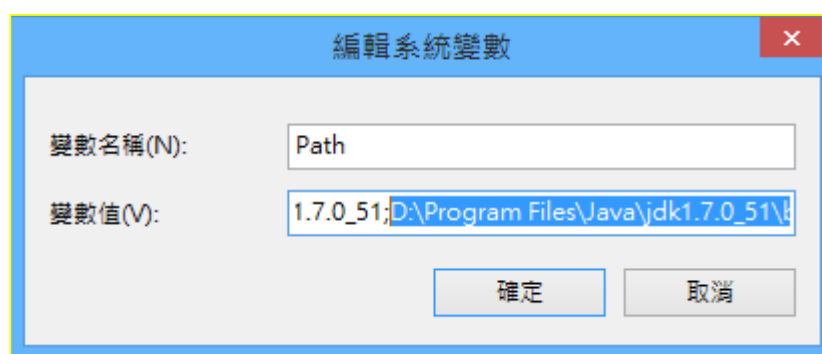
控制台\系統及安全性\系統





在變數值的最末端打上 ;D:\Program Files\Java\jdk1.7.0_51\bin;

注意：不要刪到其它的文字，然後如果原先最末端沒有”;"記得打上”;"(為了分隔其它程式的 path)



Step IV. Testing

Win7:在「搜尋程式及檔案」的地方輸入「cmd」按 enter

Win8:在開始畫面直接打 cmd 按 enter

在命令提示字元上打 javac 和 java

有下面的訊息則表示成功

```
C:\Users>java
用法: java [-options] class [args...]
        <執行類別>
    或 java [-options] -jar jarfile [args...]
        <執行 jar 檔案>

選項包括:
    -d32          使用 32 位元資料模型 <如果有的話>
    -d64          使用 64 位元資料模型 <如果有的話>
    -server       選取 "server" VM
    -hotspot      是 "server" VM 的同義字 [已不再使用]
                  預設的 VM 為 server.

    -cp <class search path of directories and zip/jar files>
    -classpath <class search path of directories and zip/jar files>
                  使用 ; 區隔的目錄、JAR 存檔以及
                  ZIP 存檔清單來搜尋類別檔案。
    -D<name>=<value>
                  設定系統屬性
    -verbose:[class|gc|jni]
                  啟用詳細資訊輸出
    -version       列印產品版本並結束
    -version:<value>
                  需要指定的版本才能執行
    -showversion   列印產品版本並繼續
    -jre-restrict-search | -no-jre-restrict-search
                  在版本搜尋中包括/排除使用者專用 JRE
    -? -help       列印此說明訊息
    -X             列印非標準選項的說明
    -ea[:<packagename>...[:<classname>]]
    -enableassertions[:<packagename>...[:<classname>]]
                  啟用含指定詳細程度的宣告
    -da[:<packagename>...[:<classname>]]
    -disableassertions[:<packagename>...[:<classname>]]
                  停用含指定詳細程度的宣告
    -esa | -enablesystemassertions
                  啟用系統宣告
    -dsa | -disablesystemassertions
                  停用系統宣告
    -agentlib:<libname>[=<options>]
                  載入原生代理程式程式庫 <libname>, 例如 -agentlib:hprof
                  另請參閱 -agentlib:jdwp=help 與 -agentlib:hprof=help
    -agentpath:<pathname>[=<options>]
                  使用完整路徑名稱載入原生代理程式程式庫
    -javaagent:<jarpath>[=<options>]
                  載入 Java 程式語言代理程式, 請參閱 java.lang.instrument
    -splash:<imagepath>
                  顯示指定影像的軟體資訊畫面
```

請參閱 <http://www.oracle.com/technetwork/java/javase/documentation/index.html> 瞭解詳細資訊。

```
C:\Users>
```

C:\Users>javac

Usage: javac <options> <source files>

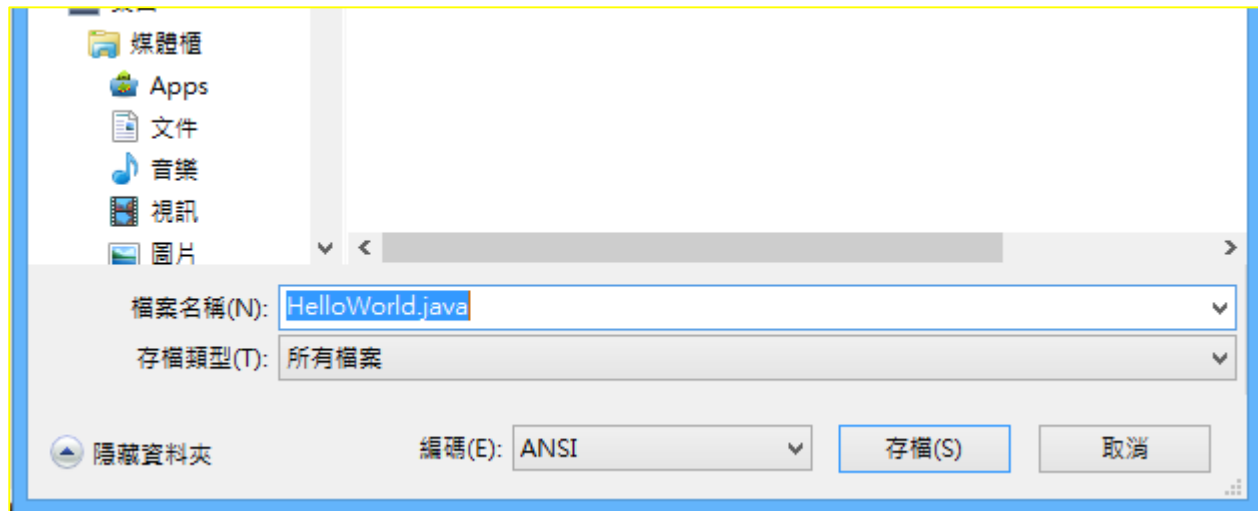
where possible options include:

-g	Generate all debugging info
-g:none	Generate no debugging info
-g:{lines,vars,source}	Generate only some debugging info
-nowarn	Generate no warnings
-verbose	Output messages about what the compiler is doing
-deprecation	Output source locations where deprecated APIs are used
-classpath <path>	Specify where to find user class files and annotation processors
-cp <path>	Specify where to find user class files and annotation processors
-sourcepath <path>	Specify where to find input source files
-bootclasspath <path>	Override location of bootstrap class files
-extdirs <dirs>	Override location of installed extensions
-endorseddirs <dirs>	Override location of endorsed standards path
-proc:{none,only}	Control whether annotation processing and/or compilation is done.
-processor <class1>[,<class2>,<class3>...]	Names of the annotation processors to run; bypasses default discovery process
-processorpath <path>	Specify where to find annotation processors
-d <directory>	Specify where to place generated class files
-s <directory>	Specify where to place generated source files
-implicit:{none,class}	Specify whether or not to generate class files for implicitly referenced files
-encoding <encoding>	Specify character encoding used by source files
-source <release>	Provide source compatibility with specified release
-target <release>	Generate class files for specific VM version
-version	Version information
-help	Print a synopsis of standard options
-Akey[=value]	Options to pass to annotation processors
-X	Print a synopsis of nonstandard options
-J<flag>	Pass <flag> directly to the runtime system
-Werror	Terminate compilation if warnings occur
@<filename>	Read options and filenames from file

Step V. First JAVA code

在一個資料匣底下建立一個檔案 HelloWorld.java

注意：如果是用新增文字的方式建立，請將存檔類型改成所有檔案



開啟它並打上：

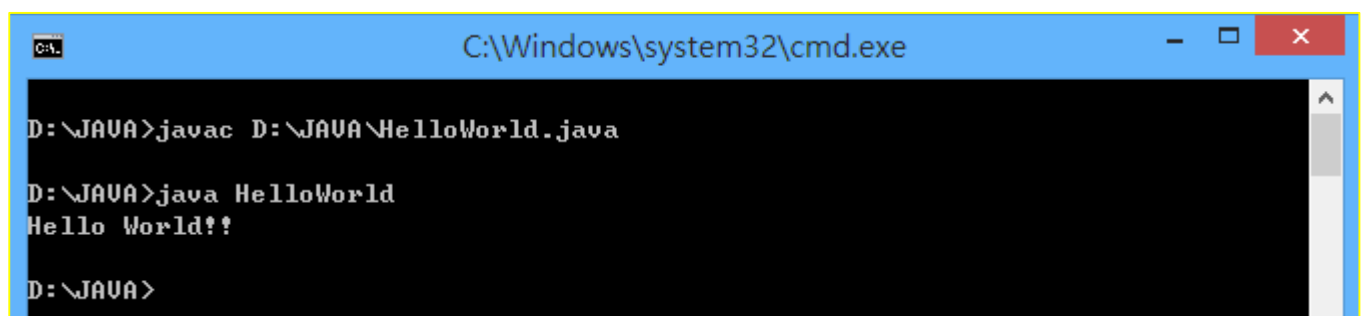
```
public class HelloWorld{  
    public static void main(String args[]){  
        System.out.println("Hello World!!");  
    }  
}
```

接著在當下的資料匣內的空白處按住 shift 加上滑鼠右鍵，點選在此處開啟命令提示視窗。
最後照著下方打，同樣者代表成功了

●javac “你當下資料匣路徑”\HelloWorld.java

此時你會發現當下的資料匣出現 HelloWorld.class 檔

●java HelloWorld



Step VI. First JAVA applet code

▲和上一步一樣，在同一個資料匣底下建立一個叫 HelloApplet.java 的檔案並打上：

```
import java.applet.Applet;
import java.awt.*;
public class HelloApplet extends Applet {
    public void paint(Graphics g) {
        g.drawString("Hello Applet", 40, 10);
    }
}
```

▲接著試著自己編譯成 class 檔

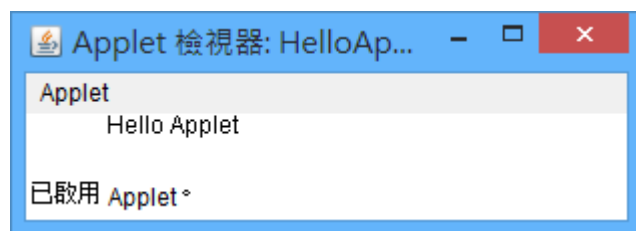
▲在同一個資料匣底下建立一個叫 appletExample.html 的檔案並打上：

```
<html>
<body>
<applet code="HelloApplet.class" height="30" width="300">
</applet>
</body>
</html>
```

▲在命令提示視窗下執行

```
D:\JAVA>appletviewer appletExample.html
```

產生下面結果則成功



最後，這是 java 所有官方的 class，以後有想用的 class 不知道怎麼用，請上這個 Java document:

<http://docs.oracle.com/javase/7/docs/api/>