

Java 程式設計入門

Java技術簡介

鄭安翔

ansel_cheng@hotmail.com

課程大綱

1) Java 簡介

- Java 發展簡史
- Java 語言特性
- Java 三大平台及架構
- Java 組織及社群

2) 建置 Java 開發環境

3) Java 程式初體驗

Java 誕生

- Sun Microsystems 公司「Green Project」
 - 開發手持式裝置 Star7 中應用程式的程式語言
 - 有限記憶體與運算資源
 - Java 中有許多為了節省資源而作的設計，像是動態載入類別檔案、字串池（String pool）等特性
 - 一開始名稱為 Oak
 - 後來發現 Oak 名稱已經被註冊了，工程師們邊喝咖啡邊討論著新名稱，最後靈機一動而改名為 Java



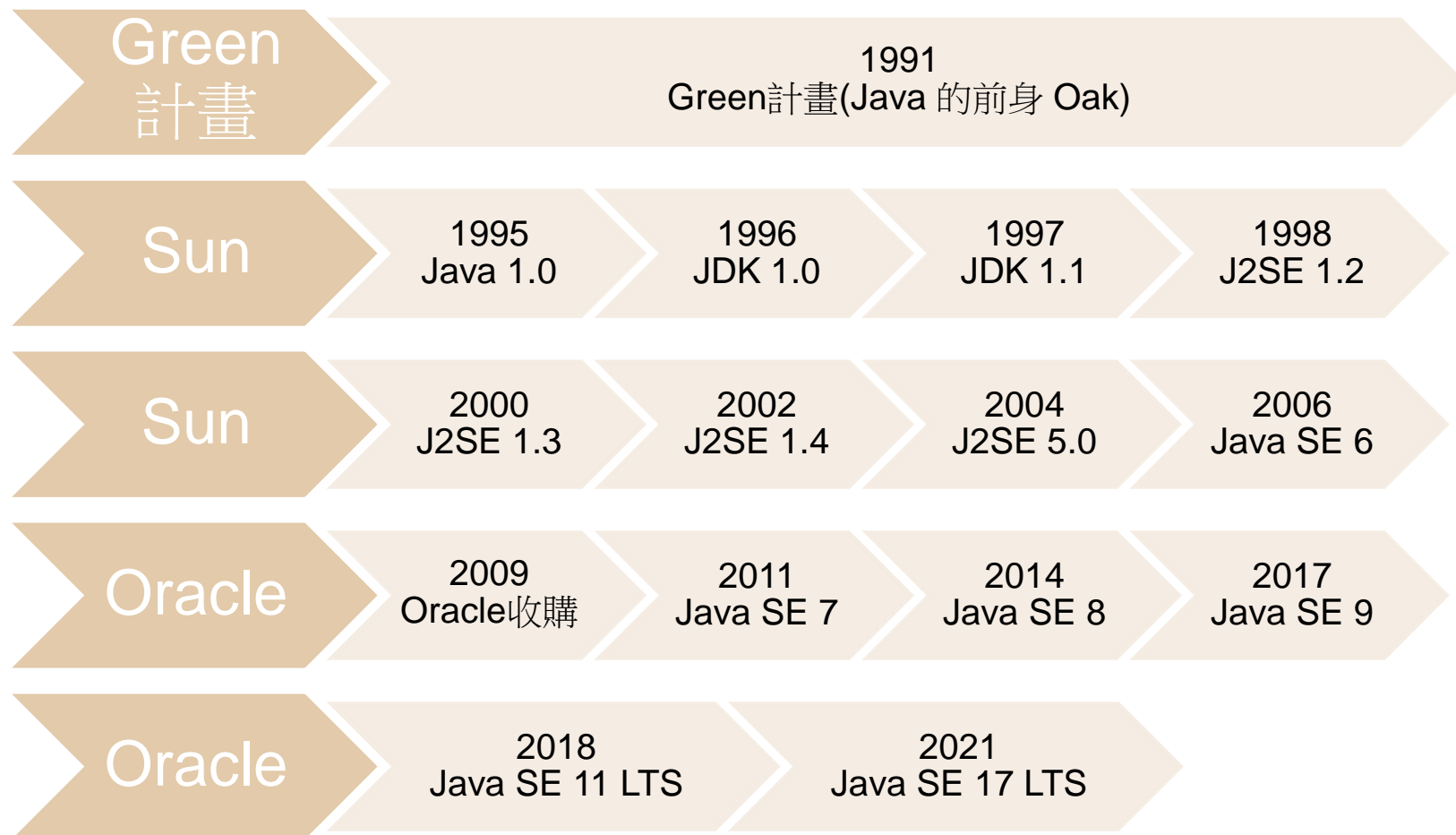
Java 誕生

- Star 7產品並未成功，Green Team面臨裁撤命運
 - 全球資訊網（World Wide Web）興起，Green Team仿照Mosaic瀏覽器，開發網路瀏覽器WebRunner(後改名HotJava)，支援Java Applet程式來製作互動式網頁元件



- 1995年5月23 日，Java Development Kits（當時JDK全名）1.0 alpha版本正式對外發表
- 1996年Netscape Navigator 2.0也正式支援Java，Microsoft Explorer亦開始支援Java

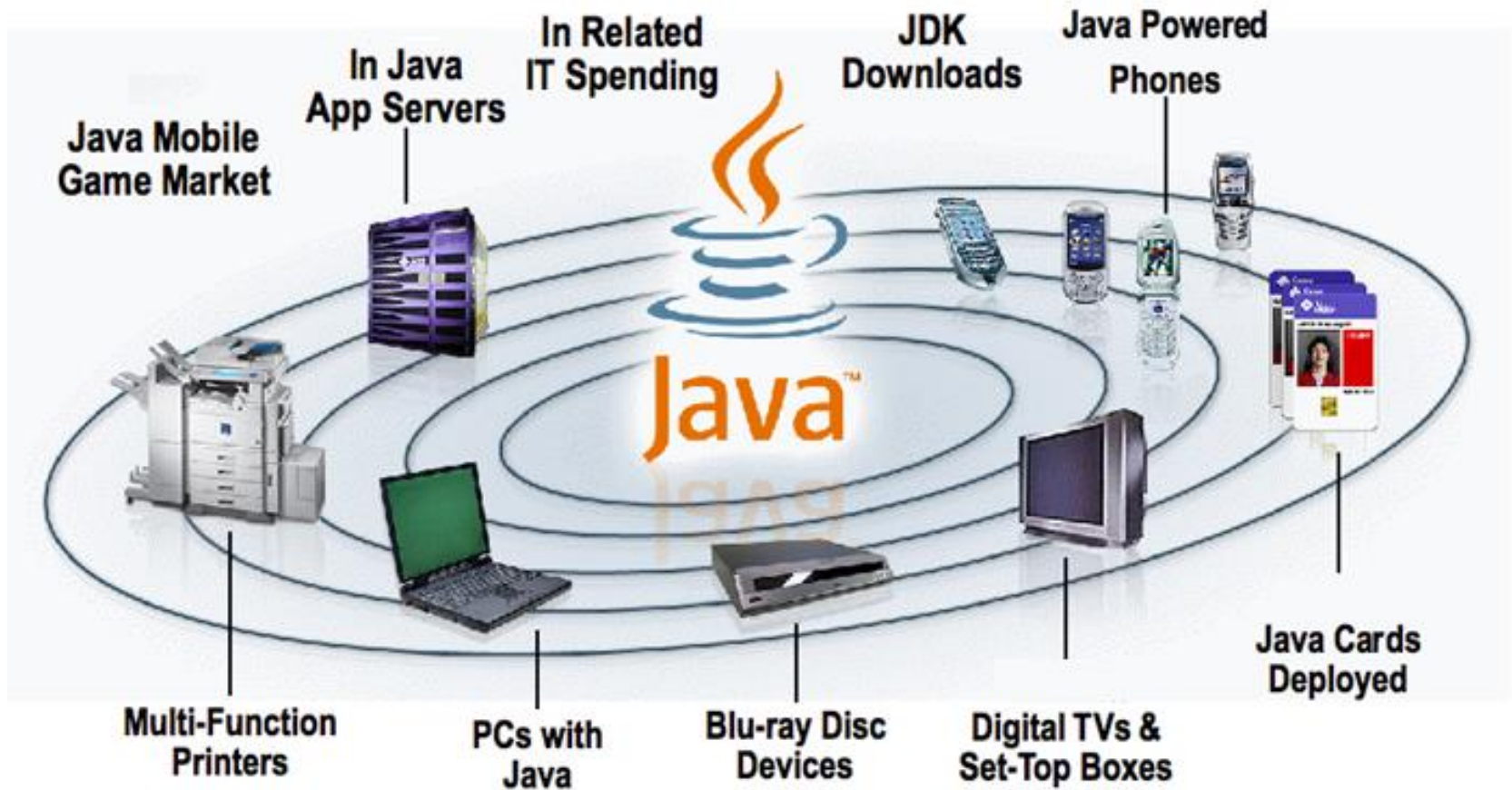
Java 版本演進



版本	發布日期	最終免費公開更新時間	最後延伸支援日期
JDK Beta	1995 年 5 月		
JDK 1.0	1996 年 1 月		
JDK 1.1	1997 年 2 月		
J2SE 1.2	1998 年 12 月		
J2SE 1.3	2000 年 5 月		
J2SE 1.4	2002 年 2 月	2008 年 10 月（由 Oracle 提供更新）	
J2SE 5.0	2004 年 9 月	2009 年 11 月（由 Oracle 提供更新）	
Java SE 6	2006 年 12 月	2013 年 4 月（由 Oracle 提供更新）	
Java SE 7	2011 年 7 月	2019 年 7 月（由 Oracle 提供更新）	2022 年 7 月
Java SE 8 (LTS)	2014 年 3 月	AdoptOpenJDK 在 2022 年 3 月或之前停止免費更新	2030 年 12 月
Java SE 9	2017 年 9 月	2018 年 3 月（由 Oracle 提供更新）	不適用
Java SE 10 (18.3)	2018 年 3 月	2018 年 9 月（由 Oracle 提供更新）	不適用
Java SE 11 (18.9 LTS)	2018 年 9 月	AdoptOpenJDK 在 2023 年 9 月或之前停止免費更新	2026 年 9 月
Java SE 12 (19.3)	2019 年 3 月	OpenJDK 在 2019 年 9 月停止免費更新	不適用
Java SE 13	2019 年 9 月	OpenJDK 在 2020 年 3 月停止免費更新	不適用
Java SE 14	2020 年 3 月	OpenJDK 在 2020 年 9 月停止免費更新	不適用
Java SE 15	2020 年 9 月	OpenJDK 在 2021 年 3 月停止免費更新	不適用
Java SE 16	2021 年 3 月	OpenJDK 在 2021 年 9 月停止免費更新	不適用
Java SE 17 (LTS)	2021 年 9 月	AdoptOpenJDK 在 2026 年 9 月或之前停止免費更新	2029 年 9 月
Java SE 18	2022 年 3 月	OpenJDK 在 2022 年 9 月停止免費更新	不適用
Java SE 19	2022 年 9 月	OpenJDK 在 2023 年 3 月停止免費更新	不適用
Java SE 20	2023 年 3 月	OpenJDK 在 2023 年 9 月停止免費更新	不適用
Java SE 21	2023 年 9 月	OpenJDK 在 2024 年 3 月停止免費更新	不適用


格式： 舊版本 舊版本，仍被支援 目前版本

Java is Everywhere



TIOBE Programming Community Index

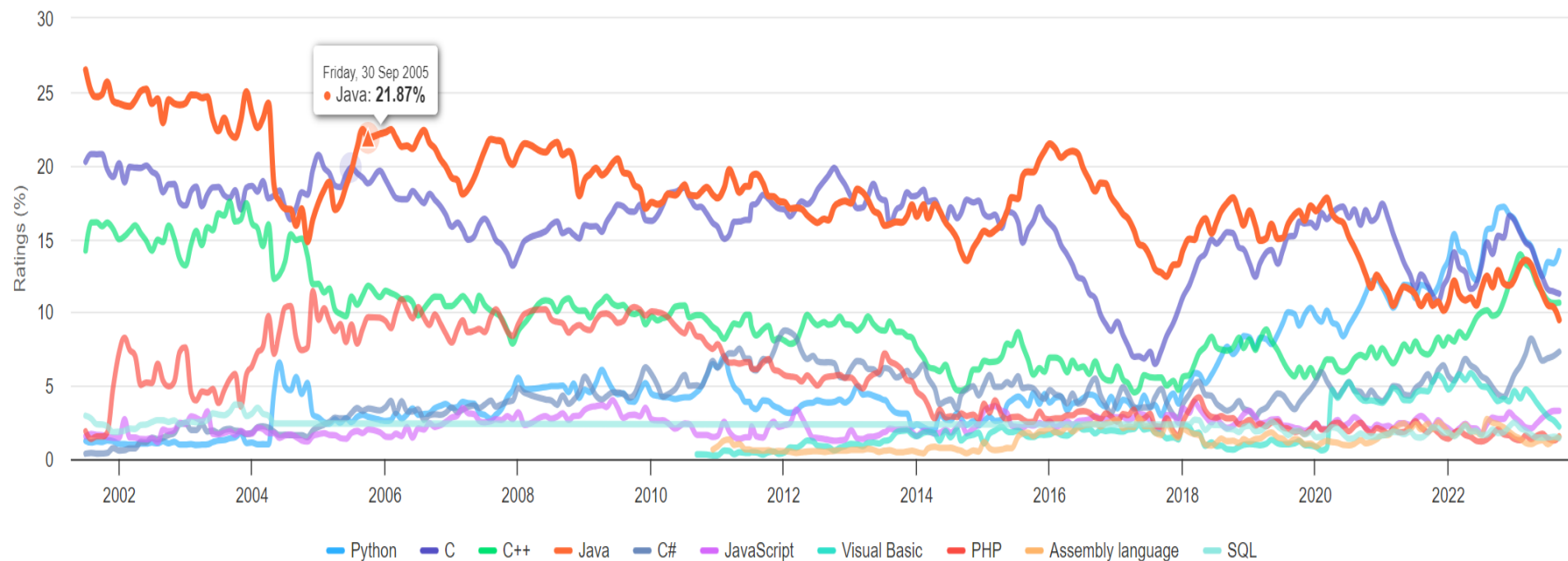
■ 程式語言普及性的指標- 你的程式技能符合時代趨勢及需要嗎？

Sep 2023	Sep 2022	Change	Programming Language		Ratings	Change
1	1			Python	14.16%	-1.58%
2	2			C	11.27%	-2.70%
3	4	▲		C++	10.65%	+0.90%
4	3	▼		Java	9.49%	-2.23%
5	5			C#	7.31%	+2.42%
6	7	▲		JavaScript	3.30%	+0.48%
7	6	▼		Visual Basic	2.22%	-2.18%
8	10	▲		PHP	1.55%	-0.13%
9	8	▼		Assembly language	1.53%	-0.96%
10	9	▼		SQL	1.44%	-0.57%

TIOBE PCI 趨勢

TIOBE Programming Community Index

Source: www.tiobe.com



Java 程式語言特性

- 物件導向
- 跨平台
- 簡單
- 安全性
- 多執行緒

跨平台 Platform Independent

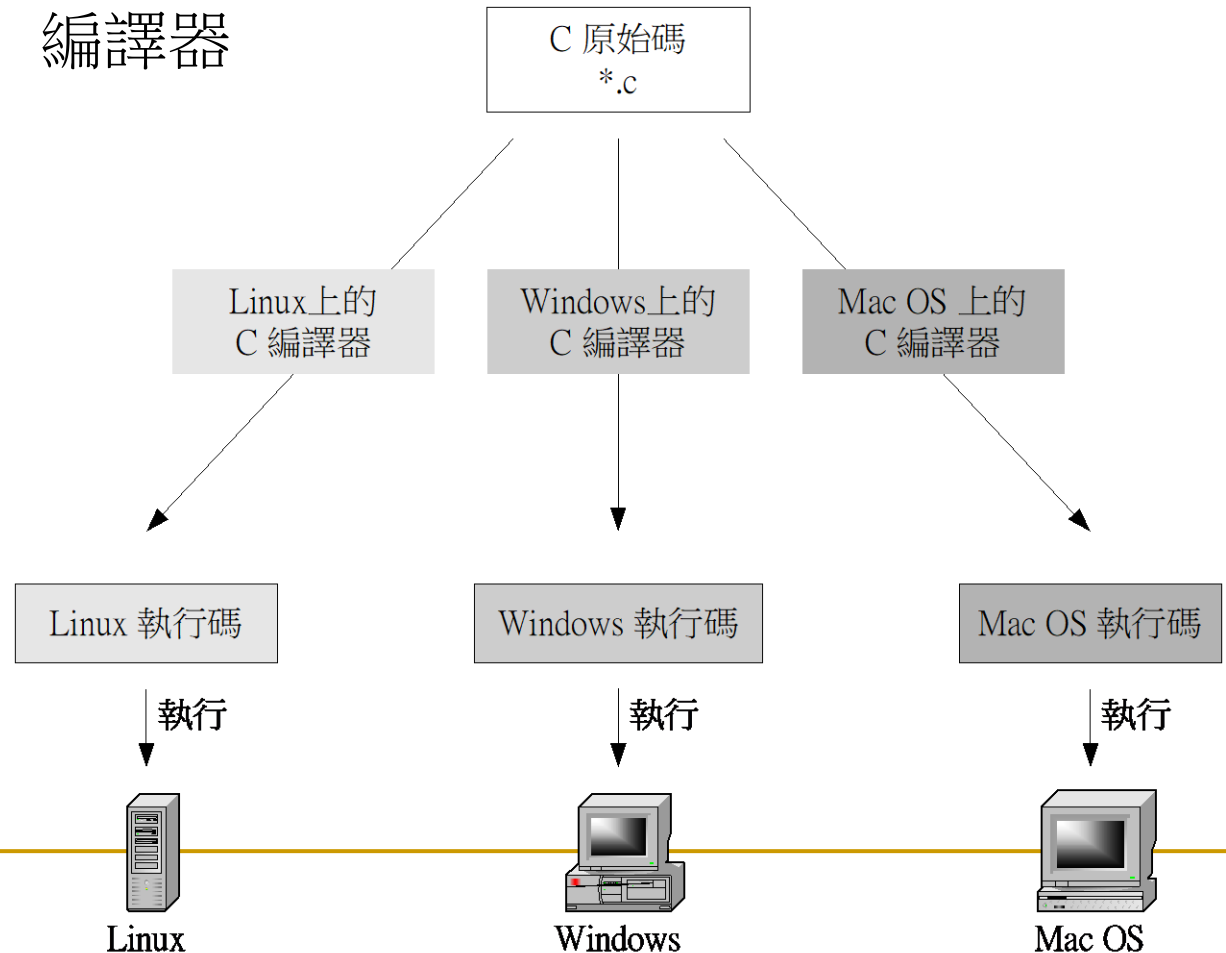
■ 傳統程式編譯流程

原始程式碼



平台相關的機器碼

編譯器



跨平台 Platform Independent

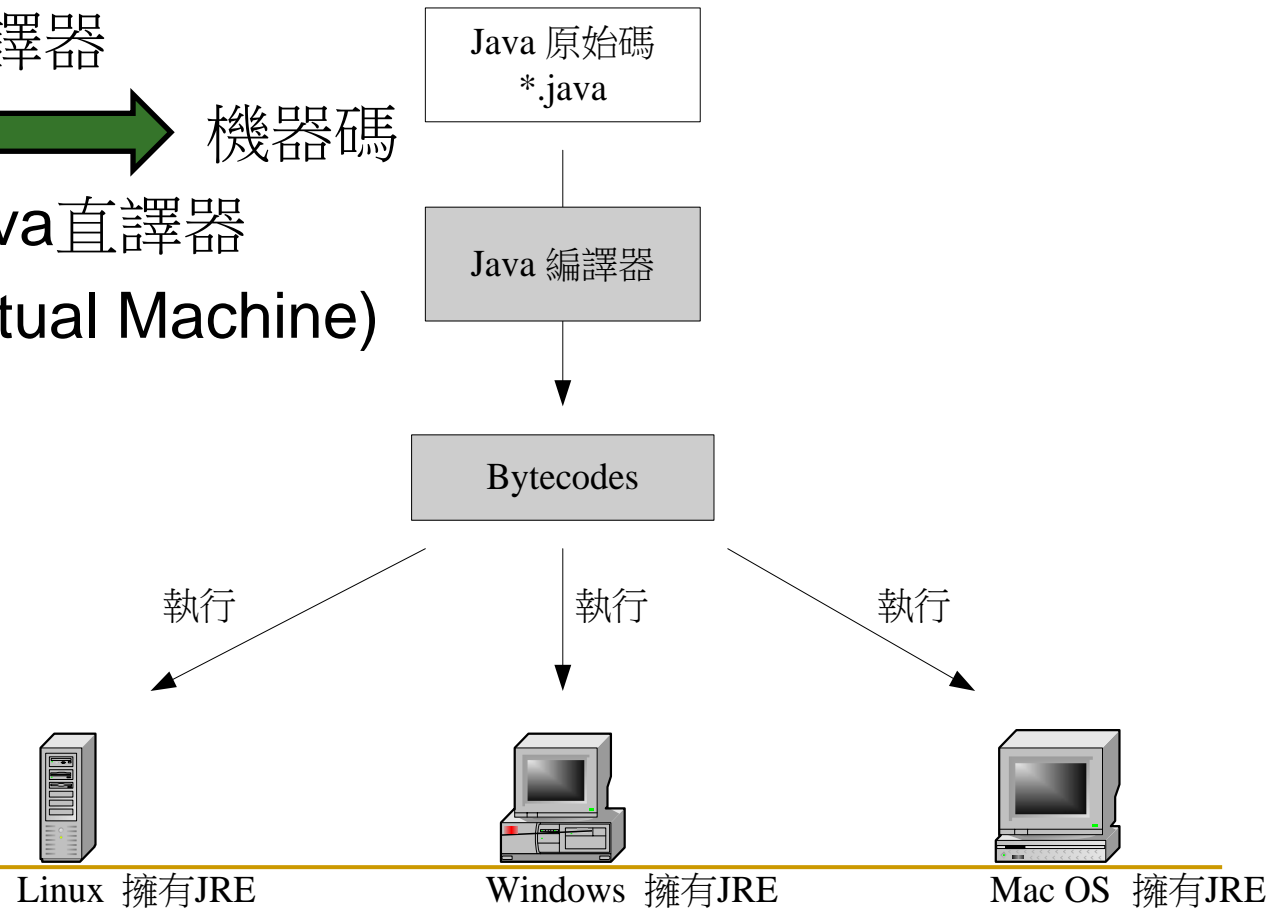
■ Java 程式編譯流程

原始程式碼  虛擬機器碼(**Byte Code**)

編譯器

虛擬機器碼  機器碼

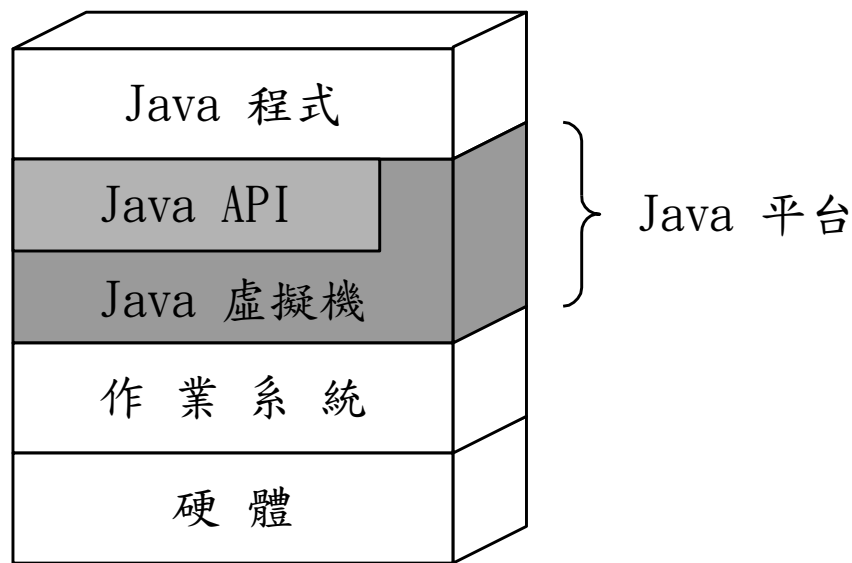
Java直譯器
(Java Virtual Machine)



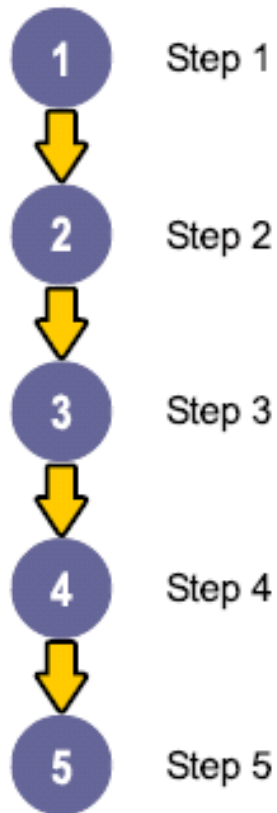
跨平台 Platform Independent

■ Java Platform

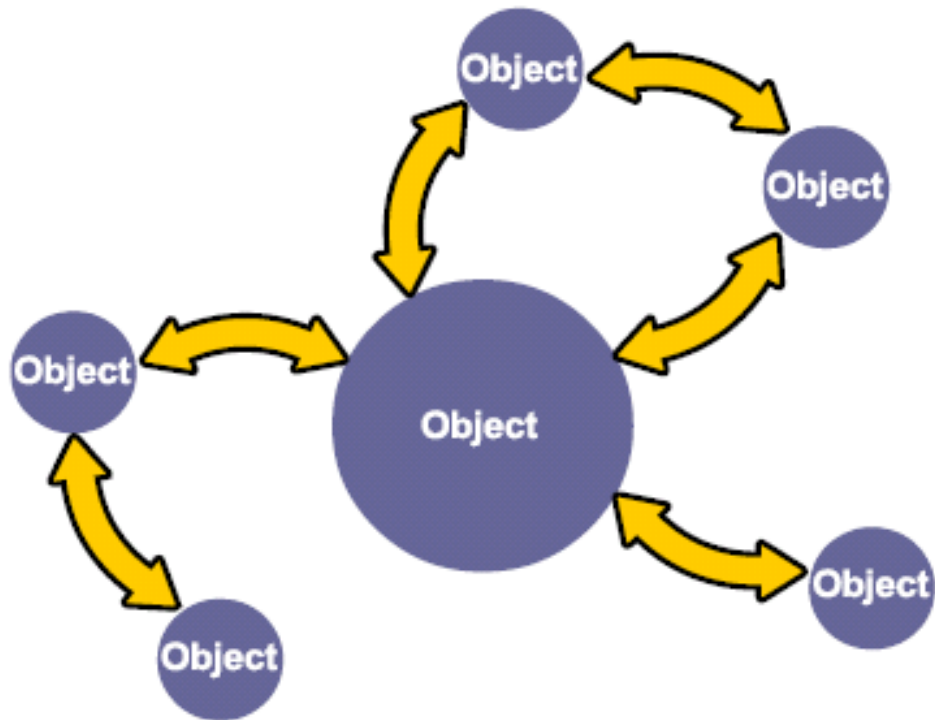
- 只要電腦上有java平台，無論作業系統是Windows、Mac OS或 Linux，Java程式都可以執行



物件導向 Object Oriented



Procedural Language



Object-oriented Language

物件導向程式 Object Oriented

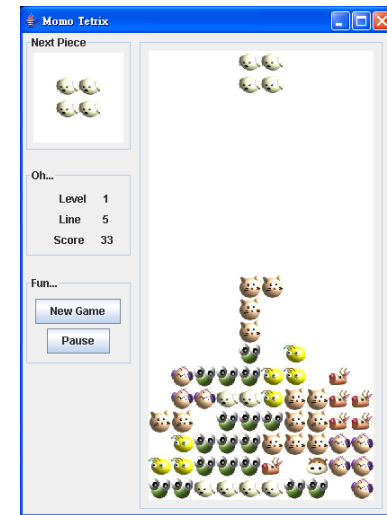
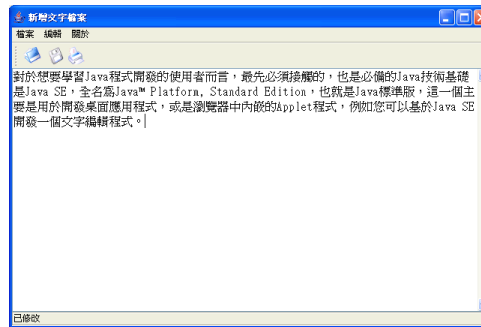
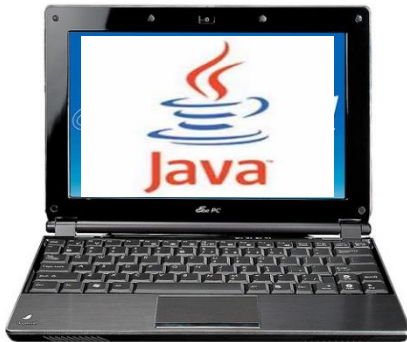
- 傳統程序式程式設計
 - 一系列對電腦下達的指令(procedure 流程)
- 物件導向程式設計
 - 一種抽象且擬人化的程式設計的方法
 - 物件作為程式的基本單元
 - 將資料與操作封裝其中
 - 透過物件之間的交互作用來完成工作
 - 物件能夠接受資料、處理資料並傳遞資料給其它物件

Java 三大平台

- Java SE(Java Platform, Standard Edition)
 - Java 技術的核心和基礎，提供一般程式開發，也是各應用平台的基礎
- Java EE(Java Platform, Enterprise Edition)
 - 提供企業伺服器端應用，如JSP、Servlet技術
- Java ME(Java Platform, Micro Edition)
 - 適用於小型數位設備應用程式開發，如PDA、手機、股票機

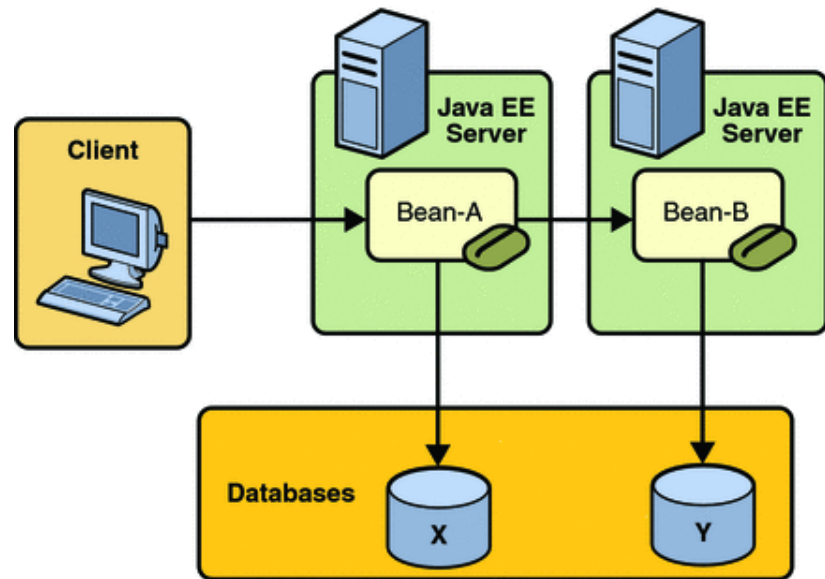
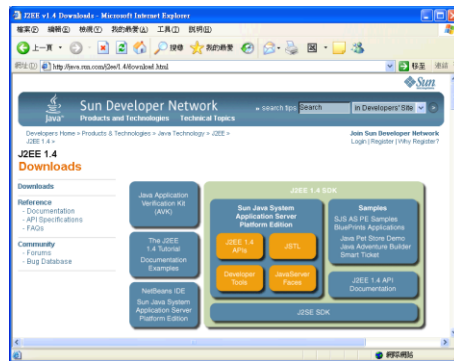
Java Standard Edition

- Java™ Standard Edition：Java 標準版
 - 主要用於開發單機桌面應用程式



Java Enterprise Edition

- Java™ Enterprise Edition : Java 企業版
 - 包含 JSP、Servlet、EJB 等多種技術
 - 分散式架構
 - 以 Java SE 為基礎

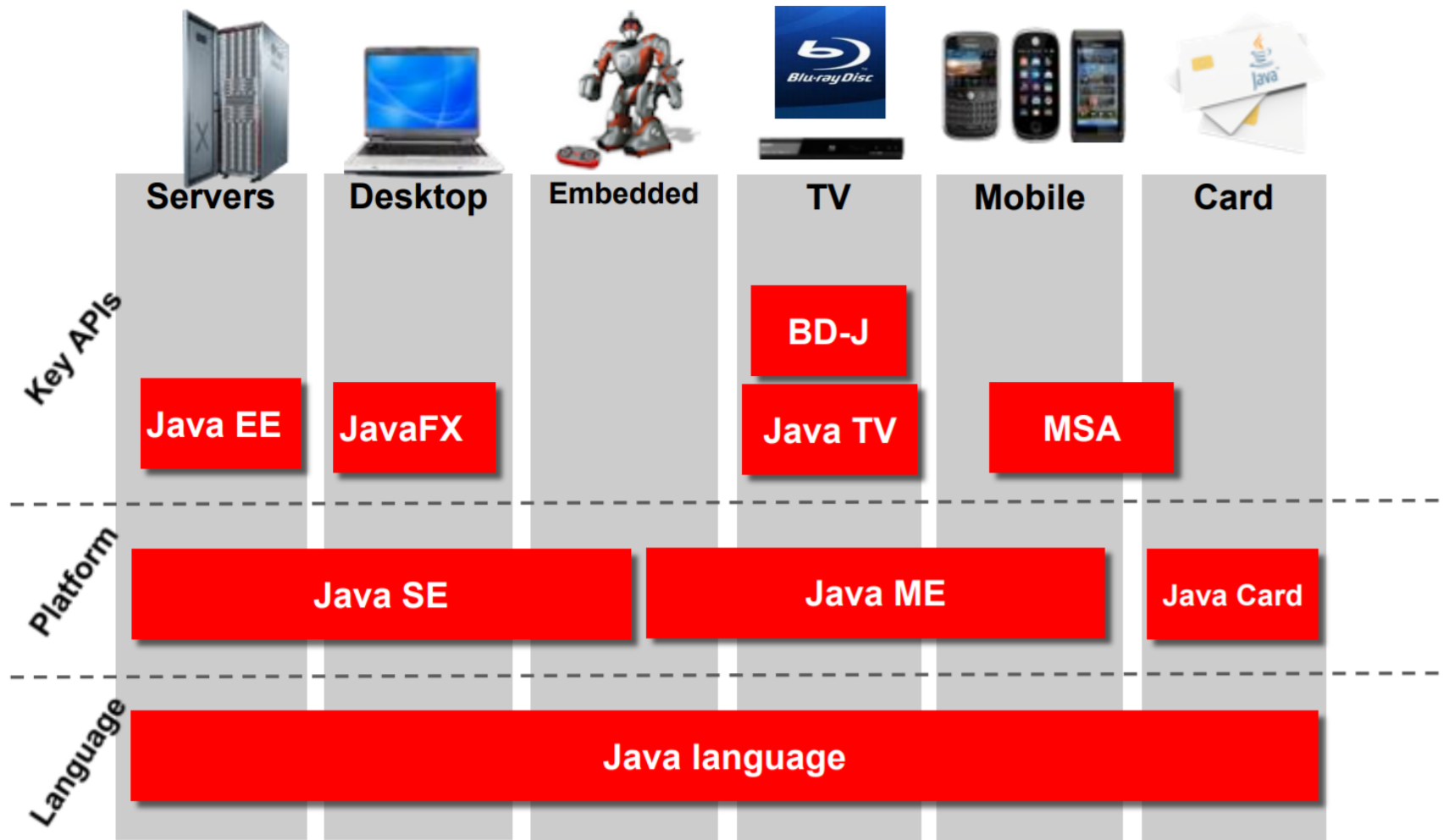


Java Micro Edition

- Java™ Micro Edition : Java 微型版
 - 針對消費性行動電子產品，例如手機或是 PDA
 - 物聯網嵌入式系統



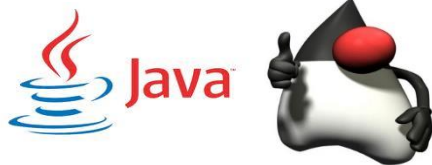
Oracle Java 光譜



Java 行動開發

■ Android 作業系統

- 以 Linux 為核心的手機作業系統
- 應用程式開發語言：Java



Java SE 四個基本架構

JDK Java標準版開發工具：

提供編譯器，及相關的開發、執行、測試等工具

JRE Java執行環境：

提供Java應用程式相關資源的一個環境

如標準程式庫、Java虛擬機器

JVM Java虛擬機器：

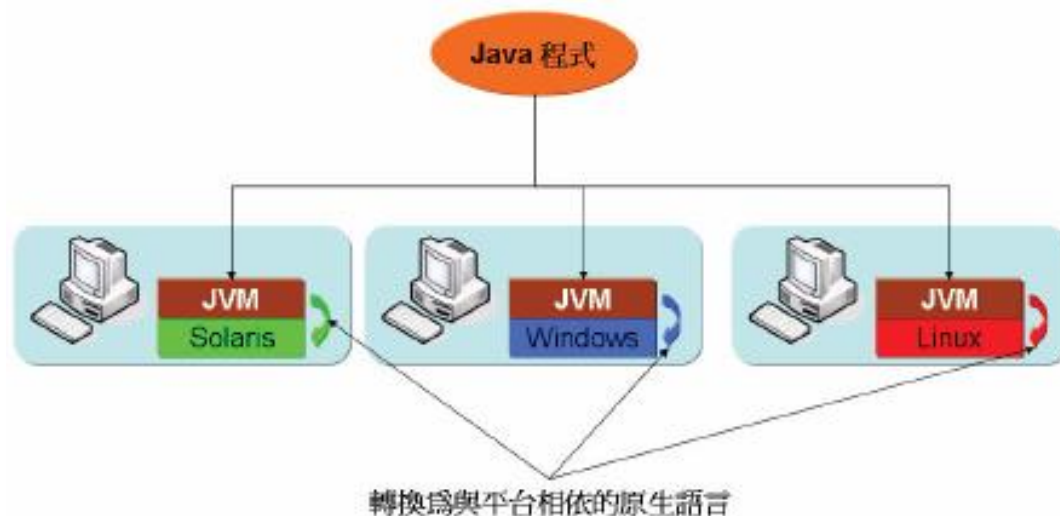
將編譯過的Java程式轉換為平台相依的原生碼

平台

Java Language						
java	javac	javadoc	jar	javap	JPDA	
JConsole	Java VisualVM	JMC	JFR	Java DB	Int'l	JVM T
IDL	Deploy	Security	Troubleshoot	Scripting	Web Services	RMI
Java Web Start			Applet / Java Plug-in			
JavaFX						
Swing		Java 2D		AWT	Accessibility	
Drag and Drop		Input Methods		Image I/O	Print Service	Sound
IDL	JDBC	JNDI	RMI	RMI-IIOP		Scripting
Beans	Int'l Support		Input/Output		JMX	
JNI	Math		Networking		Override Mechanism	
Security	Serialization		Extension Mechanism		XML JAXP	
lang and util	Collections		Concurrency Utilities		JAR	
Logging	Management		Preferences API		Ref Objects	
Reflection	Regular Expressions		Versioning		Zip	Instrumentation
Java HotSpot VM						
Solaris		Linux		Windows		Other

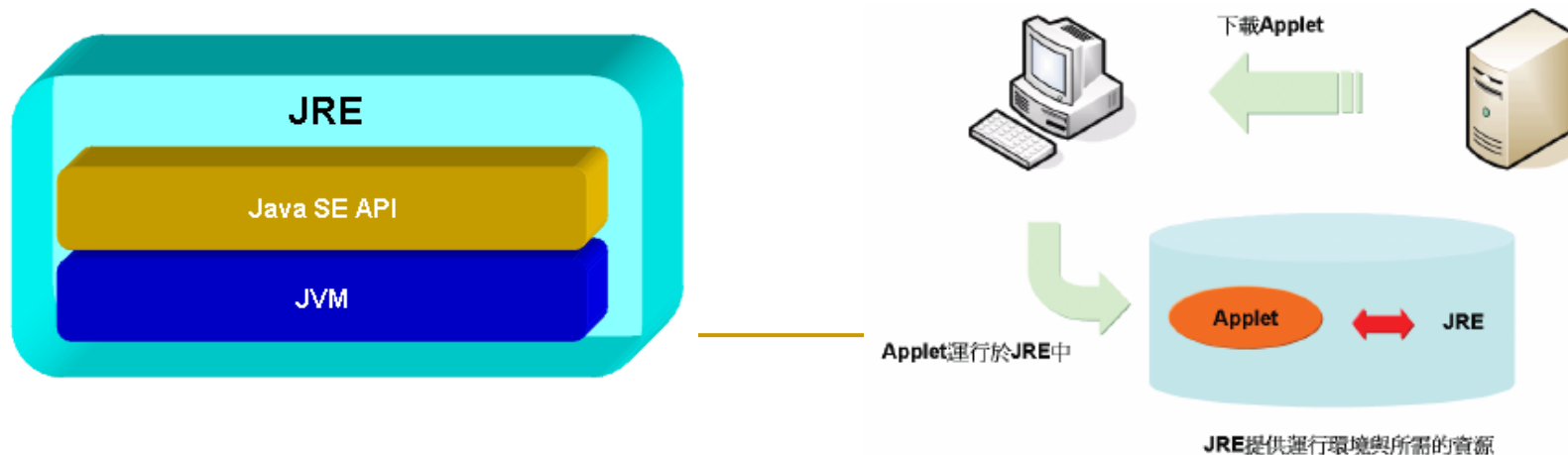
■ Java Virtual Machine (JVM)

□ ByteCode 解譯器



■ Java Runtime Environment (JRE)

□ Java Class Libraries (API) + JVM



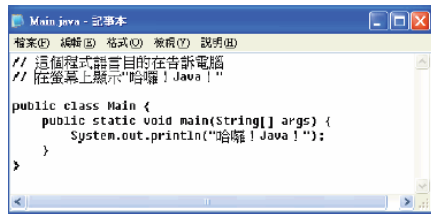
■ Java Development Kit (JDK)

□ 開發工具程式 + JRE

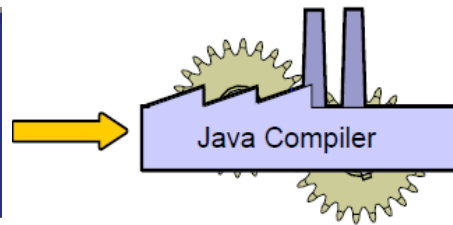
- javac.exe, java.exe, javadoc.exe, jar.exe

■ Java 程式語言

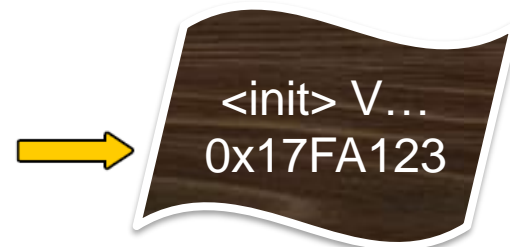
□ 副檔名為*.java的文字檔案



Hello.java

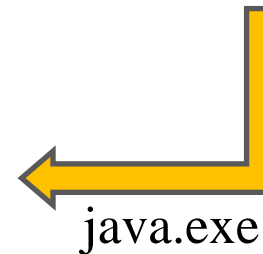
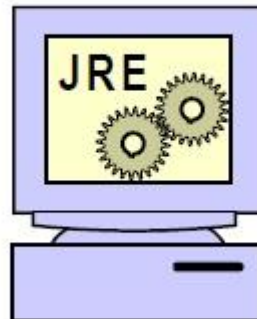


javac.exe



Hello.class

Windows Workstation



OpenJDK

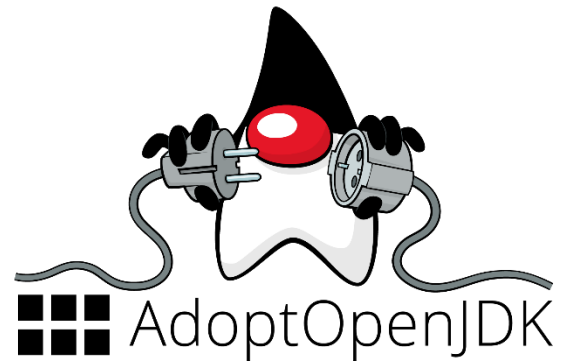
■ OpenJDK

- ❑ <https://openjdk.java.net/>
- ❑ 2006年的JavaOne大會上，Sun宣告其參考實作將開放原始碼
- ❑ 2009年4月15日正式發佈 OpenJDK
- ❑ Oracle接管Sun之後發佈的JDK7，指定OpenJDK7為官方參考實作

■ AdoptOpenJDK

- ❑ 預先建構好的OpenJDK LTS
- ❑ <https://adoptopenjdk.net/>

OpenJDK



Oracle JDK 與 OpenJDK

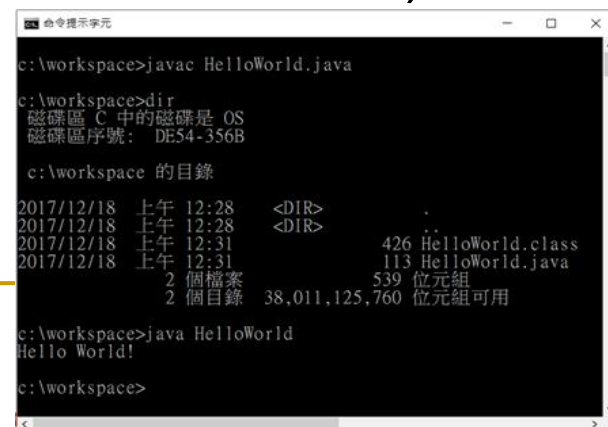
- Oracle JDK 採用 BCL (Binary Code License) 授權
 - 只能用於個人一般用途，不能用於商業用途
 - 可用於開發、測試、原型、應用程式展示
 - 不可用於資料處理、商業、產品、內部企業使用等目的
- OpenJDK 採用GNU General Public License v2 授權
 - 開放原始碼
 - 可以作為商業用途
 - 不能使用Oracle提供的安全性修補服務
 - 修補原始碼會回饋至OpenJDK的原始碼庫
 - 可以自行取得原始碼進行建構

課程大綱

- 1) Java 簡介
- 2) 建置 Java 開發環境
 - 下載與安裝 JDK
 - 查詢與安裝 Java 文件
 - 安裝文字編輯器
 - 測試Java環境
- 3) Java程式初體驗

終端機開發環境

- 終端機模式的開發環境
 - 程式執行環境資料輸入和輸出都是命令列模式(Command-line Interface)
 - 文字模式的鍵盤輸入或單純文字內容的輸出。
- 開發環境建置十分容易
 - 安裝Java開發工具Java Development Kit(JDK)
 - 配合純文字編輯工具(nano、記事本或UltraEdit)，就可以開發Java應用程式。



```
c:\workspace>javac HelloWorld.java
c:\workspace>dir
磁碟區 C 中的磁碟是 OS
磁碟區序號: DE54-356B
c:\workspace 的目錄
2017/12/18 上午 12:28 <DIR>
2017/12/18 上午 12:28 <DIR>
2017/12/18 上午 12:31 426 HelloWorld.class
2017/12/18 上午 12:31 113 HelloWorld.java
2 個檔案 539 位元組
2 個目錄 38,011,125,760 位元組可用
c:\workspace>java HelloWorld
Hello World!
c:\workspace>
```

整合開發環境

■ 整合開發環境

- 輔助程式開發人員開發程式的應用軟體
- 提供編輯、編譯、測試執行、除錯及布署等功能
- 提供圖形介面開發工具及資料庫伺服器管理
- 仍需安裝JDK

■ 常見的Java整合開發環境：

- Eclipse
- NetBeans IDE
- IntelliJ (Android Studio)

下載、安裝Java 終端機開發環境

- 下載、安裝 Java SE SDK
- 查詢、安裝 Java SE 文件
- 安裝文字編輯器 (Notepad++)
- 測試

下載與安裝 JDK

- JDK(Java SE Development Kit ,Java標準版開發套件)

<https://www.oracle.com/java/technologies/javase-downloads.html>

The screenshot shows the Oracle Java Downloads page for JDK 17 on Windows. The page features the Oracle logo and navigation links. A section titled "Java 17 available now" states that JDK 17 LTS is the latest long-term support release, free to use in production and free to redistribute. Below this, a section titled "Java SE Development Kit 17.0.2 downloads" provides a table of download options for Windows.

Product/file description	File size	Download
x64 Compressed Archive	171.34 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.zip (sha256 [2])
x64 Installer	152.43 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.exe (sha256 [2])
x64 MSI Installer	151.32 MB	https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.msi (sha256 [2])

The screenshot shows the Oracle Java Downloads page for JDK 11 on Windows. The page features the Oracle logo and navigation links. A section titled "Java SE subscribers have more choices" states that JDK 11 is also available for development, personal use, and to run other licensed Oracle products. Below this, a section titled "Java SE Development Kit 11.0.14" provides a table of download options for Windows.

Product/file description	File size	Download
x64 Installer	140.24 MB	jdk-11.0.14_windows-x64_bin.exe
x64 Compressed Archive	157.79 MB	jdk-11.0.14_windows-x64_bin.zip

JDK 文件

The screenshot shows the Oracle Java SE 11 & JDK 11 API Specification Overview page. The browser address bar shows the URL `docs.oracle.com/en/java/javase/11/docs/api/index.html`. The page has a navigation bar with tabs: OVERVIEW (selected), MODULE, PACKAGE, CLASS, USE, TREE, DEPRECATED, INDEX, and HELP. The title is "Java® Platform, Standard Edition & Java Development Kit Version 11 API Specification". Below the title, it states: "This document is divided into two sections: Java SE" and "JDK". The Java SE section describes the core Java platform APIs, and the JDK section describes APIs specific to the JDK. At the bottom, there is a table with columns "Module" and "Description". The table lists various modules including `java.base`, `java.compiler`, `java.datatransfer`, `java.desktop`, `java.instrument`, `java.logging`, `java.management`, `java.management.rmi`, `java.naming`, `java.net.http`, and `java.prefs`.

Overview (Java SE 11 & JDK 11) x +

docs.oracle.com/en/java/javase/11/docs/api/index.html

OVERVIEW MODULE PACKAGE CLASS USE TREE DEPRECATED INDEX HELP Java SE 11 & JDK 11

ALL CLASSES SEARCH: Search

Java® Platform, Standard Edition & Java Development Kit Version 11 API Specification

This document is divided into two sections:

Java SE

The Java Platform, Standard Edition (Java SE) APIs define the core Java platform for general-purpose computing. These APIs are in modules whose names start with `java`.

JDK

The Java Development Kit (JDK) APIs are specific to the JDK and will not necessarily be available in all implementations of the Java SE Platform. These APIs are in modules whose names start with `jdk`.

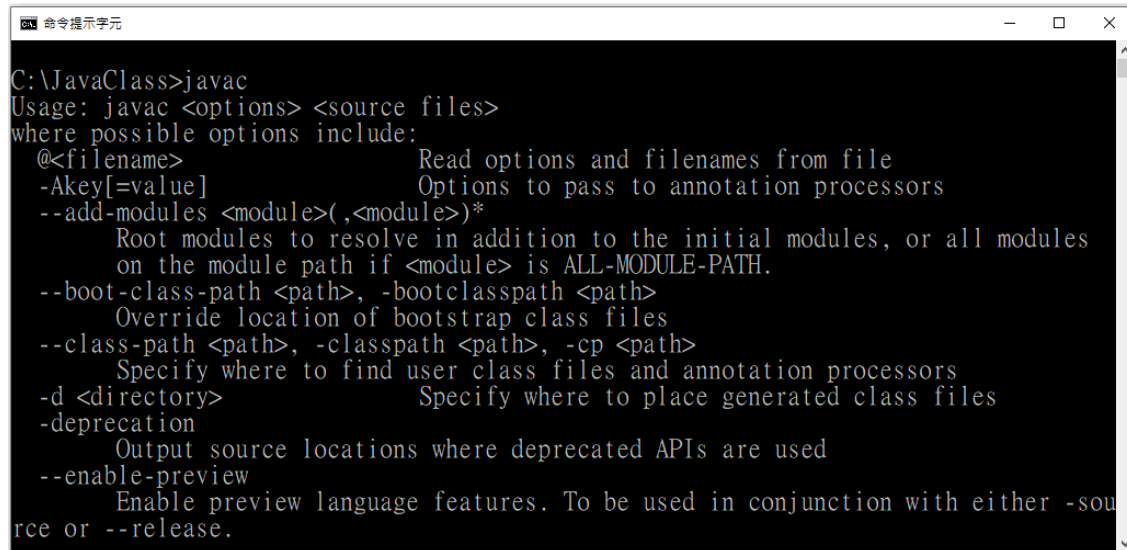
All Modules	Java SE	JDK	Other Modules
Module	Description		
<code>java.base</code>	Defines the foundational APIs of the Java SE Platform.		
<code>java.compiler</code>	Defines the Language Model, Annotation Processing, and Java Compiler APIs.		
<code>java.datatransfer</code>	Defines the API for transferring data between and within applications.		
<code>java.desktop</code>	Defines the AWT and Swing user interface toolkits, plus APIs for accessibility, audio, imaging, printing, and JavaBeans.		
<code>java.instrument</code>	Defines services that allow agents to instrument programs running on the JVM.		
<code>java.logging</code>	Defines the Java Logging API.		
<code>java.management</code>	Defines the Java Management Extensions (JMX) API.		
<code>java.management.rmi</code>	Defines the RMI connector for the Java Management Extensions (JMX) Remote API.		
<code>java.naming</code>	Defines the Java Naming and Directory Interface (JNDI) API.		
<code>java.net.http</code>	Defines the HTTP Client and WebSocket APIs.		
<code>java.prefs</code>	Defines the Preferences API.		

測試

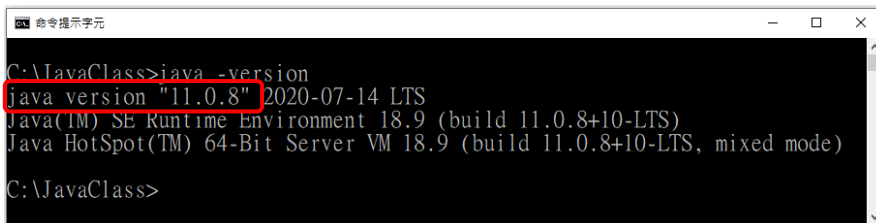
■ 測試Java環境

□ java -version

□ javac



```
C:\JavaClass>javac
Usage: javac <options> <source files>
where possible options include:
  @<filename>                Read options and filenames from file
  -Akey[=value]              Options to pass to annotation processors
  --add-modules <module>(<module>)*
                             Root modules to resolve in addition to the initial modules, or all modules
                             on the module path if <module> is ALL-MODULE-PATH.
  --boot-class-path <path>, -bootclasspath <path>
                             Override location of bootstrap class files
  --class-path <path>, -classpath <path>, -cp <path>
                             Specify where to find user class files and annotation processors
  -d <directory>             Specify where to place generated class files
  -deprecation               Output source locations where deprecated APIs are used
  --enable-preview           Enable preview language features. To be used in conjunction with either -source
                             or --release.
```



```
C:\JavaClass>java -version
java version "11.0.8" 2020-07-14 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.8+10-LTS)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.8+10-LTS, mixed mode)

C:\JavaClass>
```

課程大綱

- 1) Java 簡介
- 2) 建置 Java 開發環境
- 3) Java 程式初體驗
 - Java 程式初體驗
 - 常見編譯與執行錯誤
 - main 方法傳入參數

程式初體驗 - HelloWorld

■ 編輯

- 純文字編輯器：記事本、UltraEdit、NotePad++
- 副檔名為 .java



A screenshot of a Notepad window titled "HelloWorld.java - 記事本". The window contains the following Java code:

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

定義類別

- 類別是Java程式的基本單位
- 語法

```
[public] class <類別名稱> {  
    ..... //類別內容  
}
```

- **public**是類別的存取權限，宣告為**public**的類別稱為公開的類別，可以被其他任何其他Java類別使用
- 公開的類別檔案名稱與類別名稱相同
 - 儲存在 <類別名稱>.java的檔案中
- 一個Java原始檔案，可定義一個以上的類別，但只能有一個公開的類別

定義程式的主流程

- Java程式的執行由main方法開始

- main方法語法：

```
public static void main (String args[ ]) {  
    ..... //程式的主流程  
}
```

- 符合標準的main方法，才可被JRE當程式的進入點
- public與static是main方法的修飾字
- void表示main方法執行完畢不傳回任何值
- (String[] args) 定義main方法的傳入參數列

■ java classname ABC XYZ 123

args [0] args [1] args [2]

定義陳述句 Statement

- main方法中陳述句
 - 在命令提示字元上印文字

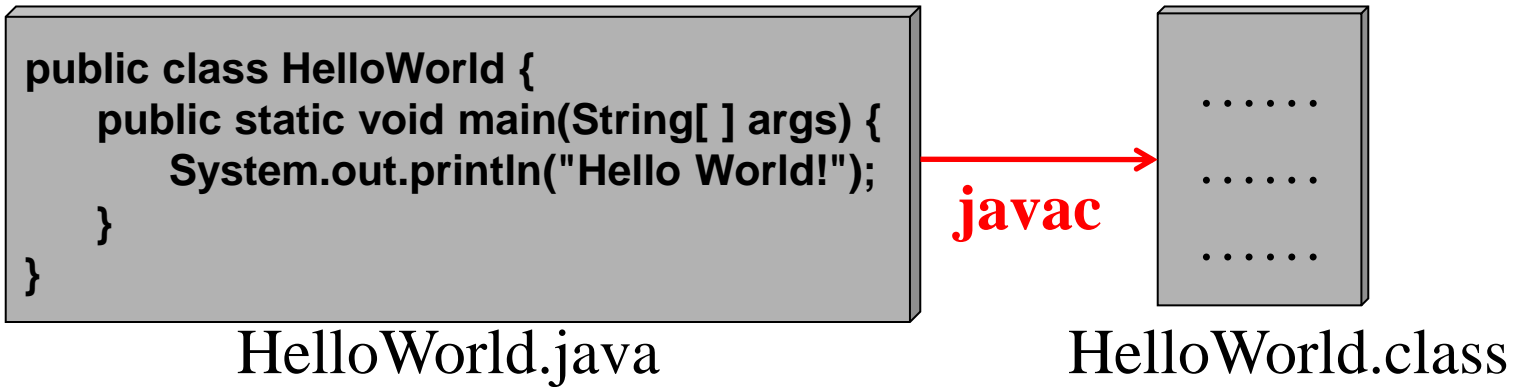
```
System.out.println("Hello World");
```

- System 類別的標準輸出成員 out (PrintStream型態)所提供的 println() 方法
- 輸出文字用雙引號 "" 包括
- 陳述句結束時要記得加分號；

編譯Java原始碼為位元碼

■ 編譯

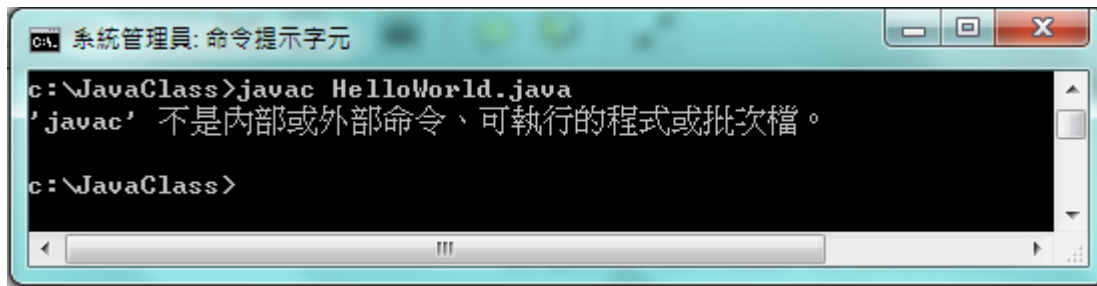
javac HelloWorld.java



The screenshot shows a Windows command prompt window titled "系統管理員: 命令提示字元". The user has executed the following commands:

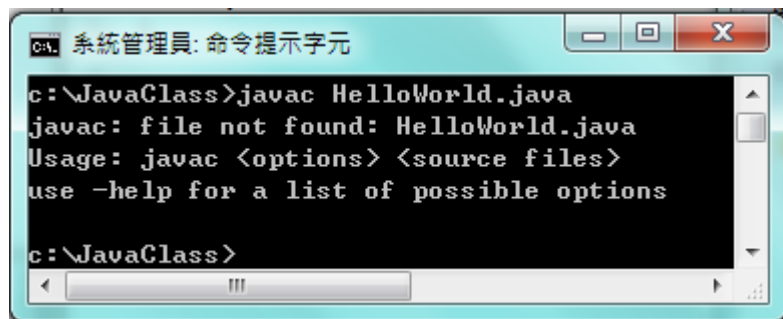
```
c:\JavaClass>javac HelloWorld.java  
  
c:\JavaClass>dir  
磁碟區 C 中的磁碟是 OS  
磁碟區序號: 7278-9798  
  
c:\JavaClass 的目錄  
  
2012/07/01 下午 09:01 <DIR> .  
2012/07/01 下午 09:01 <DIR> ..  
2012/07/01 下午 09:00 426 HelloWorld.class  
2012/07/01 下午 09:00 115 HelloWorld.java  
2 個檔案 541 位元組  
2 個目錄 21,502,488,576 位元組可用  
  
c:\JavaClass>
```

常見編譯錯誤訊息



```
系統管理員: 命令提示字元
c:\JavaClass>javac HelloWorld.java
'javac' 不是內部或外部命令、可執行的程式或批次檔。
c:\JavaClass>
```

Path環境變數設定有誤，作業系統找不到 **javac.exe** 程式



```
系統管理員: 命令提示字元
c:\JavaClass>javac HelloWorld.java
javac: file not found: HelloWorld.java
Usage: javac <options> <source files>
use -help for a list of possible options
c:\JavaClass>
```

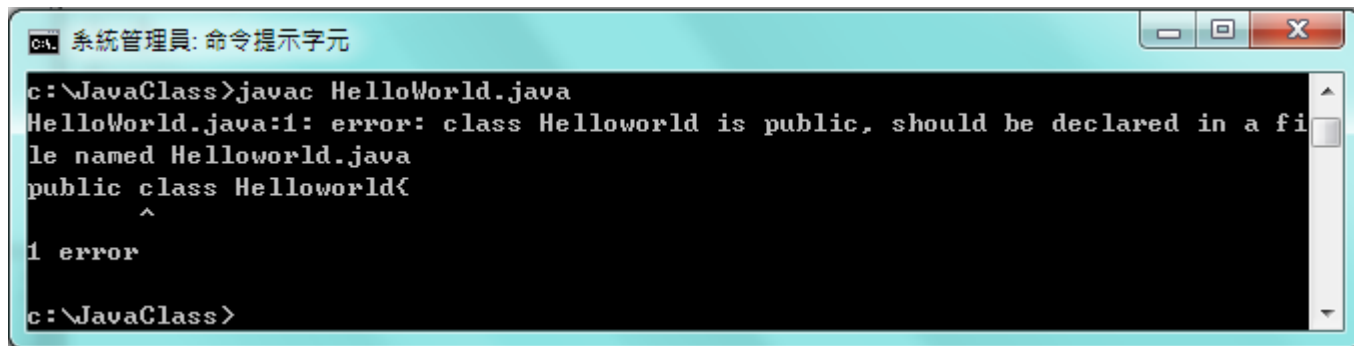
Java編譯器找不到指定的檔案

- 檔案不存在
- 路徑有誤
- 副檔名

常見編譯錯誤訊息

HelloWorld.java

```
01 public class Helloworld {  
02     public static void main(String[ ] args) {  
03         System.out.println("Hello World!");  
04     }  
05 }
```

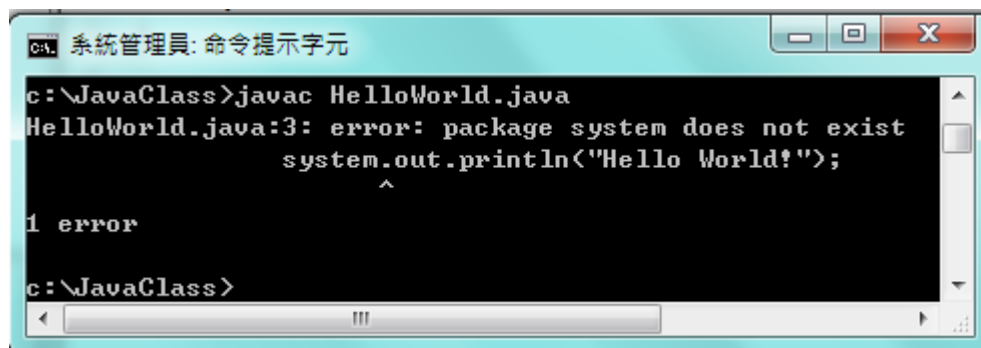


The screenshot shows a Windows command prompt window titled "系統管理員: 命令提示字元". The command prompt shows the following text:

```
c:\JavaClass>javac HelloWorld.java  
HelloWorld.java:1: error: class Helloworld is public, should be declared in a fi  
le named Helloworld.java  
public class Helloworld{  
    ^  
1 error  
c:\JavaClass>
```

常見編譯錯誤訊息

```
01 public class HelloWorld {  
02     public static void main(String[ ] args) {  
03         system.out.println("Hello World!");  
04     }  
05 }
```



The screenshot shows a Windows Command Prompt window titled "系統管理員: 命令提示字元". The command prompt shows the following text:

```
c:\JavaClass>javac HelloWorld.java  
HelloWorld.java:3: error: package system does not exist  
    system.out.println("Hello World!");  
        ^  
1 error  
  
c:\JavaClass>
```

The error message indicates that the package `system` does not exist, which is a common mistake for beginners who are not familiar with the `java.lang` package.

常見編譯錯誤訊息

```
01 public class HelloWorld {  
02     public static void main(String[ ] args) {  
03         System.Out.println("Hello World!");  
04     }  
05 }
```



The screenshot shows a Windows command prompt window titled "系統管理員: 命令提示字元". The command executed is `c:\JavaClass>javac HelloWorld.java`. The output shows a compilation error: `HelloWorld.java:3: error: cannot find symbol`, followed by the code snippet `System.Out.println("Hello World!");` with an arrow pointing to `Out`. Below this, it says `symbol: variable Out` and `location: class System`. The prompt then shows `1 error` and the command prompt returns to `c:\JavaClass>`.

常見編譯錯誤訊息

```
01 public class HelloWorld {  
02     public static void main(String[ ] args) {  
03         System.out.println("Hello World!")  
04     }  
05 }
```



The screenshot shows a Windows Command Prompt window titled "系統管理員: 命令提示字元". The command prompt shows the following text:

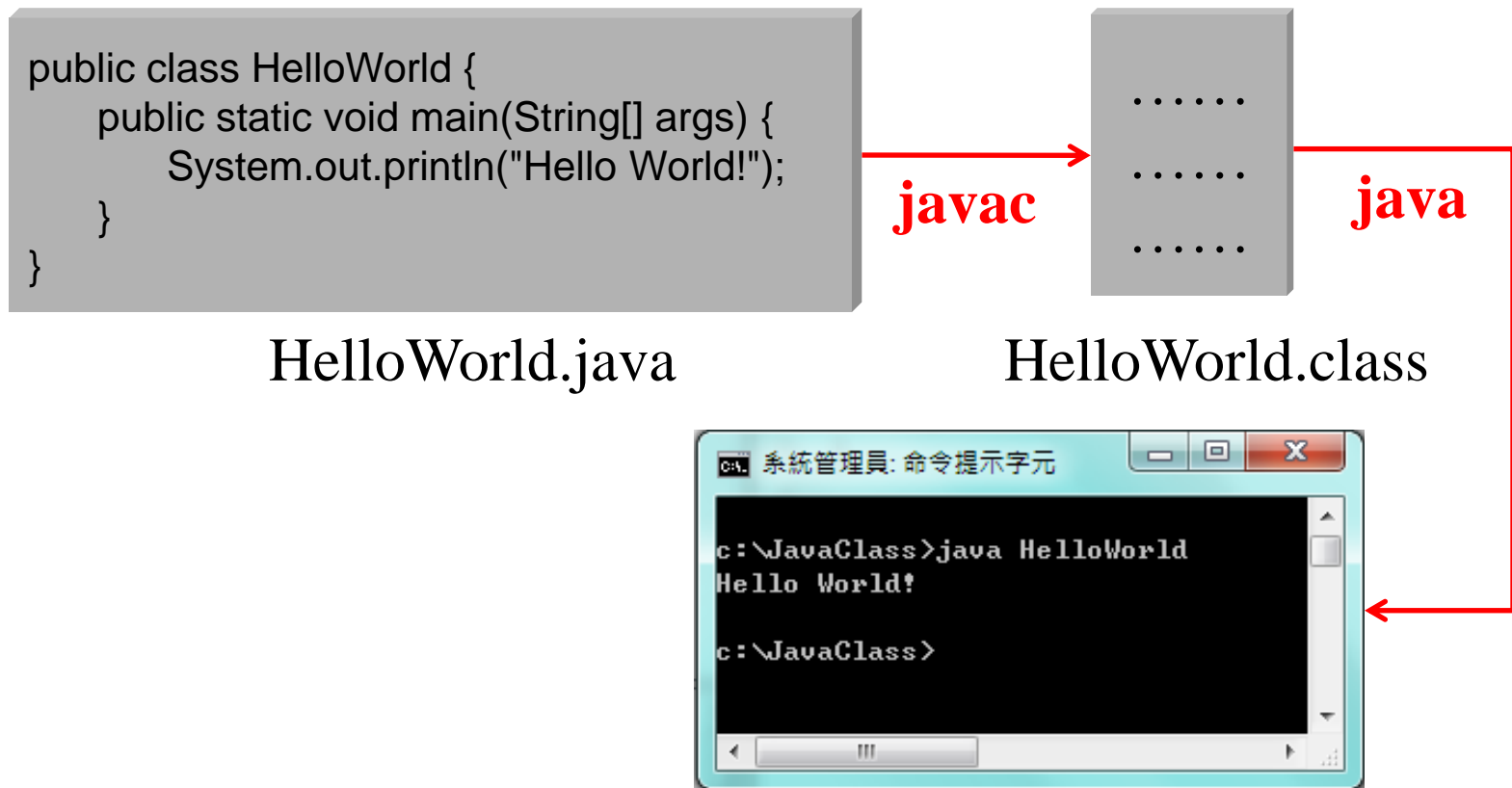
```
c:\JavaClass>javac HelloWorld.java  
HelloWorld.java:3: error: ';' expected  
        System.out.println("Hello World!")  
                           ^  
1 error  
  
c:\JavaClass>
```

The error message indicates that a semicolon is expected at the end of the `System.out.println("Hello World!")` statement on line 3.

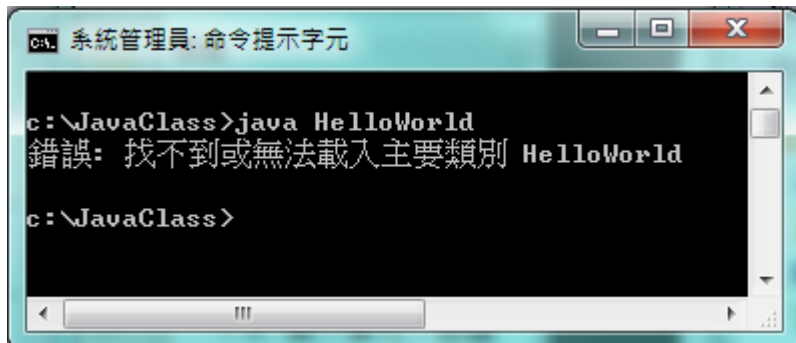
執行Java位元碼

■ 執行

java HelloWorld



常見執行錯誤訊息



A screenshot of a Windows Command Prompt window titled "系統管理員: 命令提示字元". The prompt shows the command `c:\JavaClass>java HelloWorld` being entered. The output is an error message in Chinese: "錯誤: 找不到或無法載入主要類別 HelloWorld". The prompt then returns to `c:\JavaClass>`.

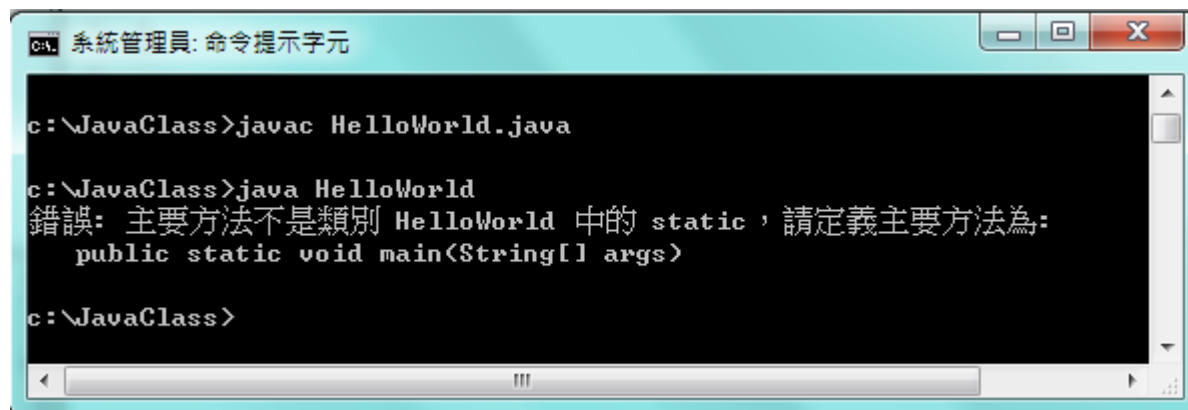
```
系統管理員: 命令提示字元
c:\JavaClass>java HelloWorld
錯誤: 找不到或無法載入主要類別 HelloWorld
c:\JavaClass>
```

Java執行程式找不到指定的檔案

- 類別檔案不存在
- ClassPath路徑有誤

常見執行錯誤訊息

```
01 public class HelloWorld {  
02     public void main(String[] args) {  
03         System.out.println("Hello World!");  
04     }  
05 }
```



The screenshot shows a Windows Command Prompt window titled "系統管理員: 命令提示字元". The user has entered the following commands:

```
c:\JavaClass>javac HelloWorld.java  
  
c:\JavaClass>java HelloWorld
```

The output shows an error message in Chinese:

```
錯誤: 主要方法不是類別 HelloWorld 中的 static, 請定義主要方法為:  
    public static void main<String[] args>
```

The prompt is now at the same directory:

```
c:\JavaClass>
```