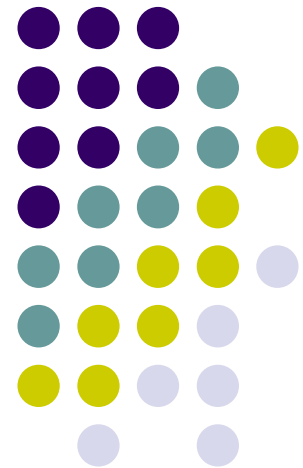


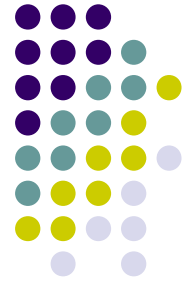
# 第一章

## 認識Java

---

Java的發展歷史  
Java的虛擬機器(JVM)  
Java未來的發展  
撰寫第一個Java程式





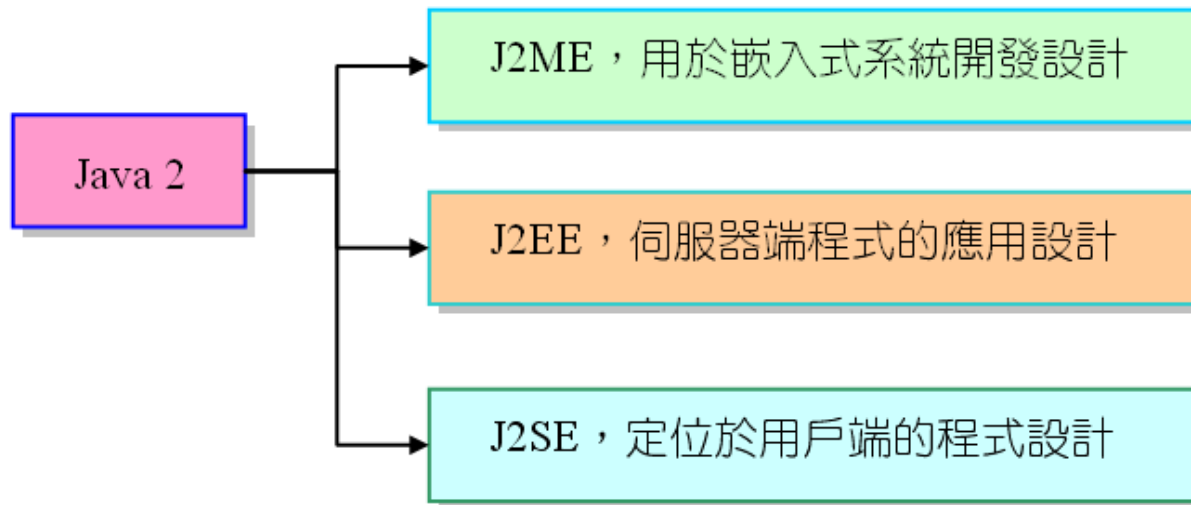
# Java的歷史

- Java於1995年誕生，由美國加州的昇陽電腦公司（Sun Microsystems, Inc.）所推出
  - Java語言的前身是Oak，即橡樹之意
  - Sun將Oak修改成為Java，並將這種技術轉移到Web上
  - Web於1995年開始盛行，Java也隨之一炮而紅



# Java的應用領域

- 下圖為Java的三大應用領域：





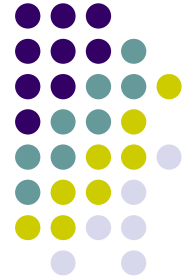
# Java的特點 (1/3)

- 嬌小且完美的語言
  - Java的語法簡潔、勻稱，容易撰寫且易於維護
- 具物件導向的功能
  - Java具有物件導向的特點
  - 它是全新的語言，因而可以將物件導向的特性發揮到極至
- 完全支援網際網路
  - Java所撰寫的程式很容易地在Browser裡呈現
  - 可藉此讓所有能夠上網的電腦都能執行Java程式



# Java的特點 (2/3)

- 通用的語言
  - 凡是C/C++所能做到的事，Java也都能做到
- 跨平台的語言
  - 可在不修改程式碼的情況下，便能在不同的作業系統執行
- 具有豐富的函數庫
  - Java語言背後有龐大的函數庫支撐
  - 這些函數庫包含了繪圖函數庫、圖形使用者介面函數庫（graphical user interface）、網路設計函數庫...等等



# Java的特點 (3/3)

- 特殊的處理機制
  - 「多執行緒」機制可在同一時間執行不同的程序
  - 「垃圾收集」機制可將無用的變數所佔用之記憶體釋放
  - 「例外處理」機制可依情況拋出例外，使得程式不會因此中斷執行



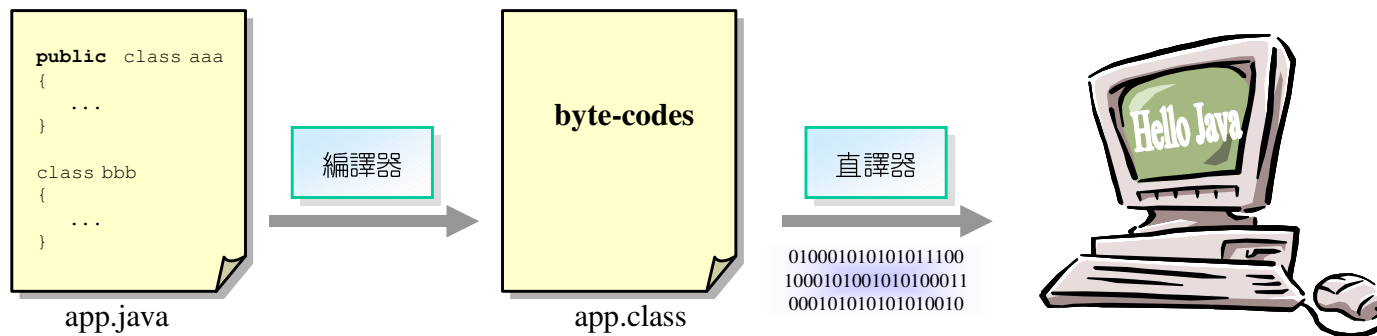
# 編譯與直譯

- 程式語言可分編譯式與直譯式兩種
  - 常見的編譯式語言如 C、FORTRAN與COBOL等均是
  - 直譯式的語言如 BASIC、JavaScript、VBScript、Python、Ruby 及其它Script語言
- Java程式的執行則是先編譯，後直譯



# Java的位元碼

- Java的編譯與執行的程序：
  - 將程式碼編譯成與平台無關（platform-independent）的機器碼，稱之為「位元碼」（byte-codes）
  - 經編譯後，可在裝有JVM上的平台直接執行
- 下圖說明Java相關的執行流程：

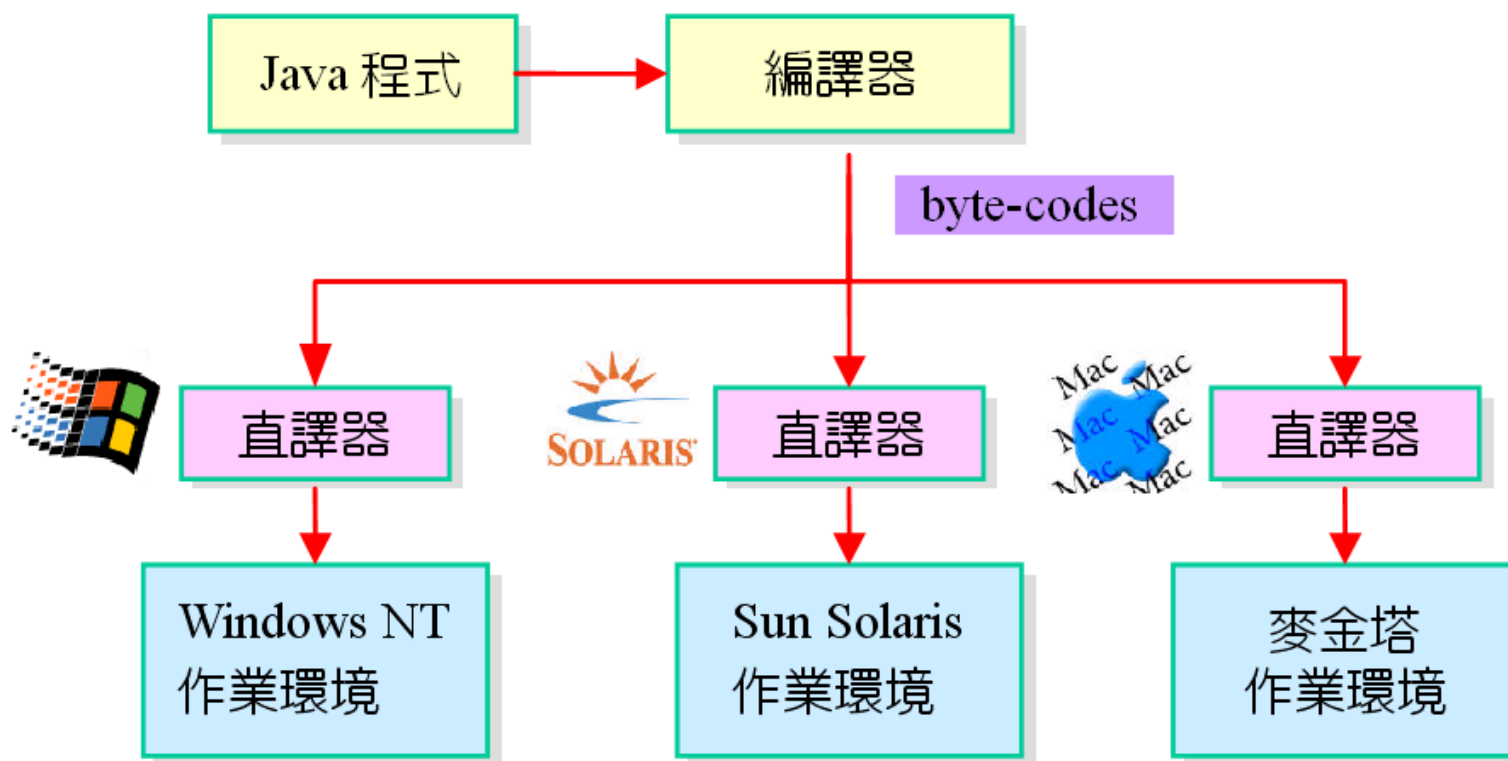


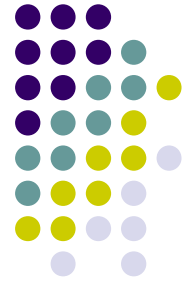




# byte-codes 的執行

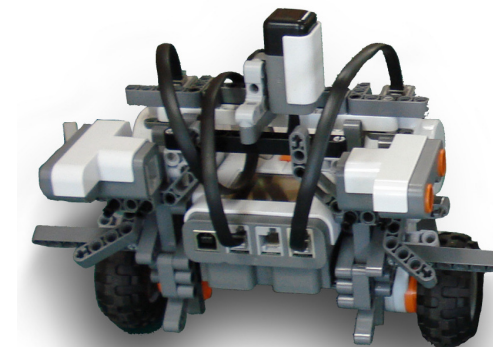
- byte-codes最大的好處是--可跨越平台來執行：





# Java的重要性

- 在學術領域裡，Java已成為最重要的教學程式語言之一
- 許多產品，如手機與PDA，也是使用Java來延伸





# Java程式的分類

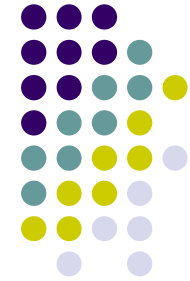
- Java程式可分為下列兩種：
  - (1) Java應用程式（Java application）
  - (2) 應用在www上的小程序（Java applet）



## 簡單的程式範例 (1/3)

- 下面是app1\_1的程式碼
  - 經編譯和執行後，會在Dos視窗上顯示Hello Java!!字串：

```
01 // app1_1, 簡單的 java application
02 public class app1_1
03 {
04     public static void main(String args[])
05     {
06         System.out.println("Hello Java!!");
07     }
08 }
```



## 簡單的程式範例 (2/3)

- 把app1\_1鍵入記事本中，檔名設為app1\_1.java：

(1) 用記事本編輯程式app1\_1

```
// app1_1, 簡單的java application
public class app1_1
{
    public static void main(String args[])
    {
        System.out.println("Hello Java !!!");
    }
}
```

(2) 存入C:\Java資料夾

(3) 鍵入檔名app1\_1.java

(4) 存檔類型請選擇「所有檔案」



## 簡單的程式範例 (3/3)

- app1\_1的執行流程如下圖所示：

```
C:\ 命令提示字元
Microsoft Windows XP [版本 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>cd c:\Java

C:\Java>javac app1_1.java — 編譯程式

C:\Java>java app1_1 — 直譯程式  切換路徑到檔案存放處
Hello Java !!
C:\Java> — 執行結果
```



# Applet的範例 (1/6)

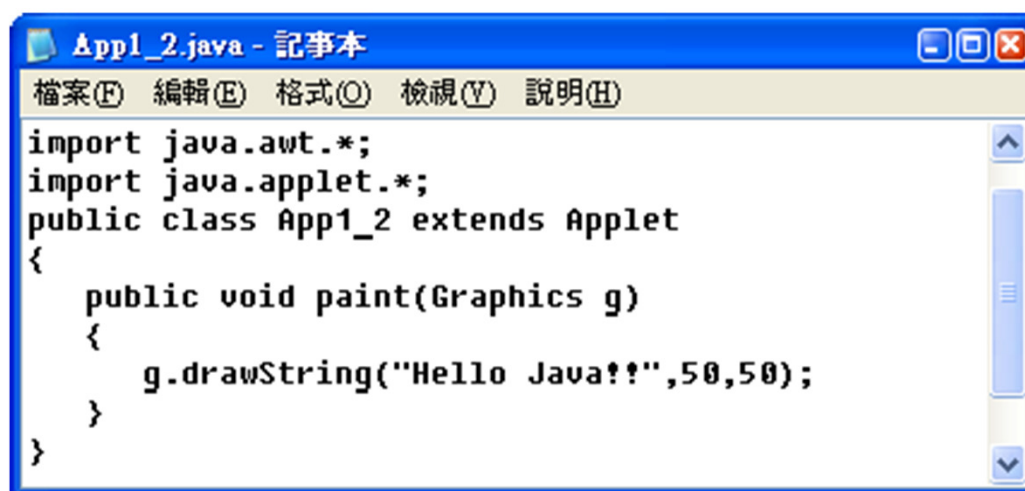
- App1\_2.java是個簡單的applet
  - 可在瀏覽器視窗上顯示 "Hello Java!!" 字串：

```
01 // App1_2, Java applet
02 import java.awt.*;
03 import java.applet.*;
04 public class App1_2 extends Applet
05 {
06     public void paint(Graphics g)
07     {
08         g.drawString("Hello Java!!", 50, 50);
09     }
10 }
```



## Applet的範例 (2/6)

- 用記事本編輯App1\_2.java :



```
App1_2.java - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

import java.awt.*;
import java.applet.*;
public class App1_2 extends Applet
{
    public void paint(Graphics g)
    {
        g.drawString("Hello Java!?",50,50);
    }
}
```

- 本書是以大寫A開頭的檔名代表applet程式，因此請把檔名存成App1\_2.java，而非app1\_2.java





# Applet的範例 (3/6)

- 編譯Java applet

- 編譯App1\_2.java :

```
c:\Java> javac App1_2.java
```

- 準備好HTML 檔案

- 下面是一個典型的html範例檔案：

```
01  <!-- App1_2.htm -->
02  <HTML>
03  <APPLET CODE    = "App1_2.class"
04              WIDTH = "200"
05              HEIGHT = "100" >
06              很抱歉，您的瀏覽器不支援 Java applet!!
07  </APPLET>
08  </HTML>
```



# Applet的範例 (4/6)

- 將html的語法編輯好，並將它存檔為App1\_2.htm：

(1) 用記事本編輯 htm 檔

```
<!-- App1_2.htm -->
<HTML>
<APPLET CODE    = "App1_2.class"
      WIDTH    = "200"
      HEIGHT   = "100" >
很抱歉，您的瀏覽器不支援Java applet!!
</APPLET>
</HTML>
```

(2) 存入 C:\Java 資料夾

(3) 鍵入檔名 App1\_2.htm

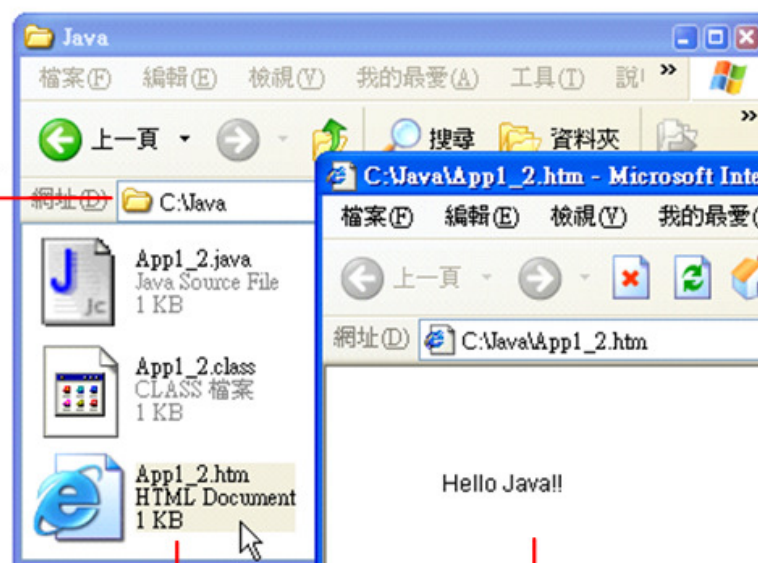
(4) 存檔類型請選擇「所有檔案」



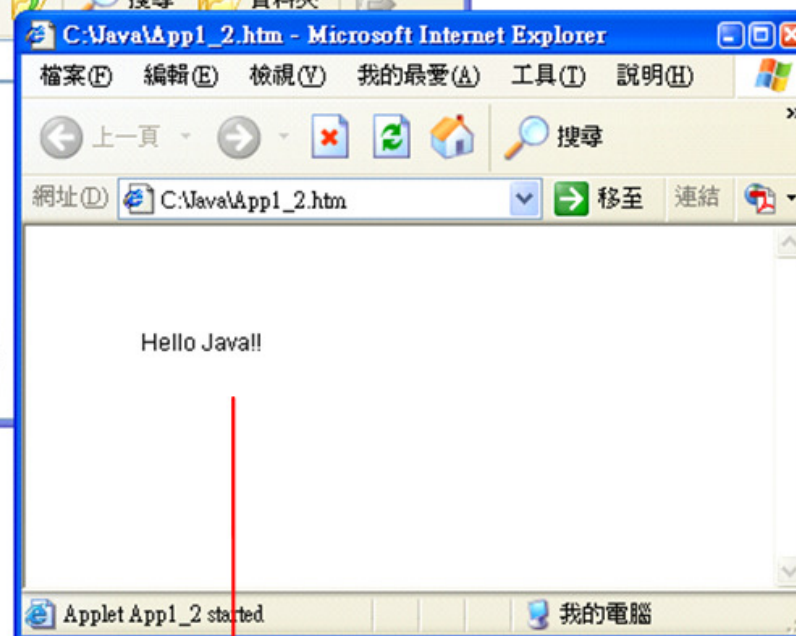
# Applet的範例 (5/6)

- 執行Java applet

(1) 找到 C:\Java  
資料夾



(2) 開啟 App1\_2.htm



(3) applet 的執行結果



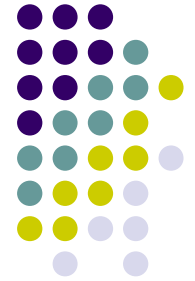
## Applet的範例 (6/6)

- 利用applet檢視器執行Java applet

```
C:\Java> appletviewer App1_2.htm
```

- 「Applet檢視器」視窗與執行結果：





-The End-