



巨匠線上真人

Java 先修基礎程式設計

www.pcschoolonline.com.tw

同學，歡迎你參加本課程

- ☑ 請關閉你的FB、Line等溝通工具，以免影響你上課。
- ☑ 考量頻寬，請預設關閉麥克風、攝影機，若有需要再打開。
- ☑ 隨時準備好，老師會呼叫你的名字進行互動。
- ☑ 如果有緊急事情，你必需離開線上教室，請用聊天室私訊給老師，以免老師癡癡呼喚你的名字。
- ☑ 先倒好水、上個洗手間，準備上課囉^^

課程檔案下載

The screenshot displays the PC School Online website interface. The top navigation bar includes links for '巨匠電腦線上真人' (Giant Master Computer Online Real Person), '開課查詢' (Class Inquiry), '免費體驗專區' (Free Experience Area), '課程總覽' (Course Overview), '專業師資' (Professional Faculty), '學員專區' (Student Area), '講師專區' (Teacher Area), and '最新消息' (Latest News). Social media icons for Line, Facebook, and YouTube are also present. A user is logged in, indicated by '您好!' and a '登出' (Logout) button.

The main banner features the text: **程式語言好難學?** (Programming Language is so hard to learn?), **那是因為你還沒學過Python!** (That's because you haven't learned Python!), and **線上老師 LIVE 直播教學 · 搶先看** (Online Teacher LIVE Broadcast Teaching · Preview).

A dropdown menu is open from the '學員專區' (Student Area) link, listing various options. The '課程檔案下載' (Course Archive Download) option is highlighted with an orange box and a callout bubble. Other options in the menu include '點數卡產品兌換', 'APCS檢測專區', '公告專區', '我的課表', 'IT真人課程劃位', '電腦分校課程劃位', '外語真人課程劃位', '美語分校課程劃位', '取消劃位', '上課權益查詢', '教學平台測試', '學習諮詢', '常見問題', '個資維護', '忘記密碼', and '登出'.

The background of the banner shows a stylized image of a person's head with circuitry and data lines, suggesting a focus on technology and learning.

ZOOM 學員操作說明

The screenshot shows the Zoom interface with several callouts:

- 5 查看選項/共同註記/筆 (連連看)**: Points to the '查看選項' (View Options) dropdown menu, which includes '原始大小' (Original Size), '請求遠端控制' (Request Remote Control), '共同註記' (Annotate), and '退出全螢幕' (Exit Full Screen). The '共同註記' option is highlighted with an orange box.
- 筆**: Points to the '筆' (Pen) icon in the toolbar, which is also highlighted with an orange box.
- 2 共享螢幕 (指導演練; 點評作品)**: Points to the '共享螢幕' (Share Screen) button in the bottom toolbar. The text below it says: '老師須先停止共享螢幕 才能請學生共享螢幕' (The teacher must first stop sharing the screen before asking the student to share the screen).
- 1 聊天**: Points to the '聊天' (Chat) button in the bottom toolbar.
- 3 與會者/舉手**: Points to the '與會者' (Participants) button in the bottom toolbar, which has a small '1' next to it.
- 4 解除靜音**: Points to the '解除靜音' (Unmute) button in the bottom toolbar.

Other visible elements include the Zoom logo, the URL 'www.pcschool.com.tw', a green status bar at the top saying '您正在觀看綠世界的螢幕' (You are watching the screen of the Green World), and a list of participants on the left side of the screen.

本課程各堂教學主題

- ◆ 第一堂：認識 Java 技術
- ◆ 第二堂：建立 Java 類別、宣告、初始化和使用變數
- ◆ 第三堂：資料傳遞、運算式
- ◆ 第四堂：流程控制(一)
- ◆ 第五堂：流程控制(二)



巨匠線上真人

Java 先修基礎程式設計

第一堂：認識 Java 技術

本堂教學重點

1. Java 簡介

1-1. Java 發展簡史

1-2. Java 三大平台

2. Java 語言程式系統架構及特色

2-1. Java SE 架構

Q&A

2-2. Java SE 程式運作

Q&A

3. 安裝和建置 Java 環境

3-1. 下載與安裝 JDK 8

3-2. 環境變數設定

練習1：環境變數設定

4. Java 程式的編譯與執行

4-1. 使用記事本撰寫

練習2：撰寫第一個 Java 程式

4-2. 編譯與執行 Java 程式

練習3：編譯與執行程式

5. Java 程式結構

5-1. 註解

5-2. package (套件)

5-3. import (引用類別)

5-4. class 類別

5-5. 主方法 main()

5-6. 敘述句

Q&A

◆ 下堂教學重點

本堂教學重點

1. Java 簡介

1-1. Java 發展簡史

1-2. Java 三大平台

2. Java 語言程式系統架構及特色

2-1. Java SE 架構

Q&A

2-2. Java SE 程式運作

Q&A

3. 安裝和建置 Java 環境

3-1. 下載與安裝 JDK 8

3-2. 環境變數設定

練習1：環境變數設定

4. Java 程式的編譯與執行

4-1. 使用記事本撰寫

練習2：撰寫第一個 Java 程式

4-2. 編譯與執行 Java 程式

練習3：編譯與執行程式

5. Java 程式結構

5-1. 註解

5-2. package (套件)

5-3. import (引用類別)

5-4. class 類別

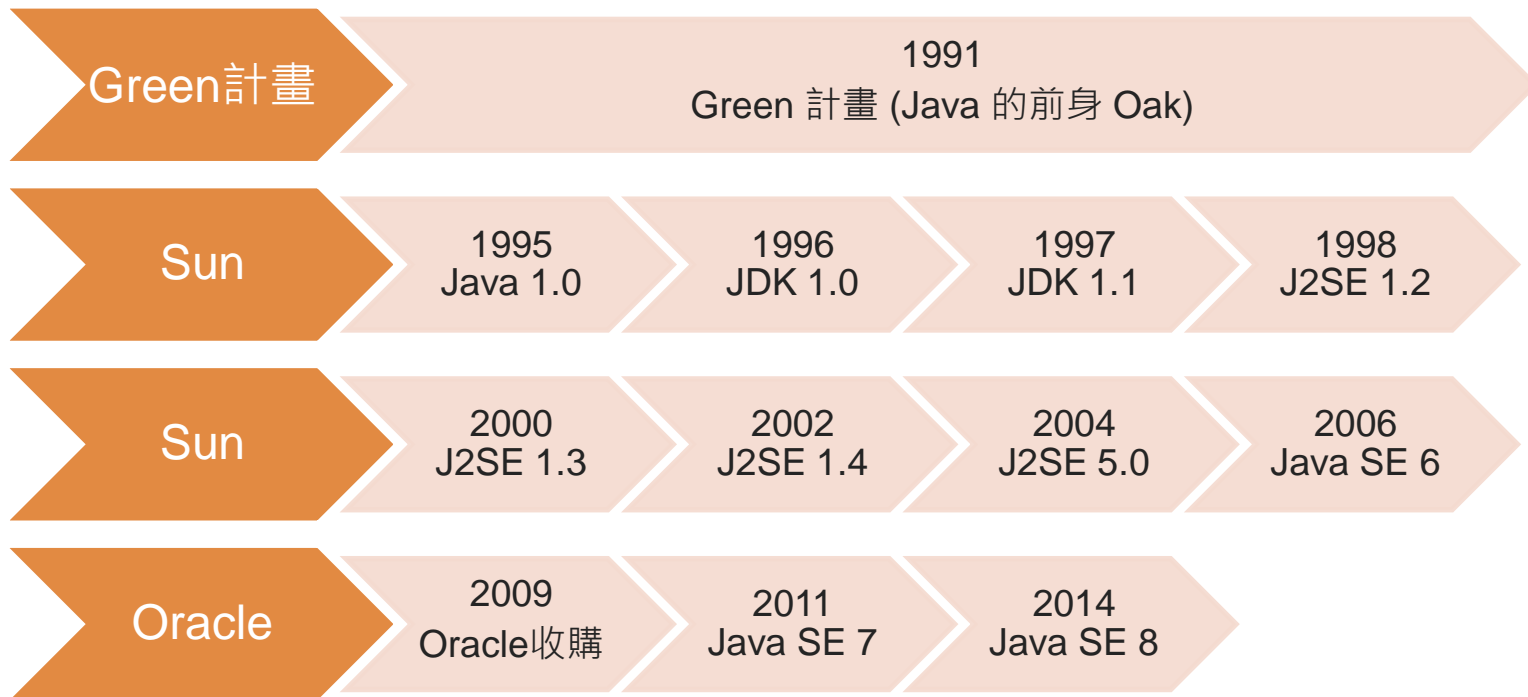
5-5. 主方法 main()

5-6. 敘述句

Q&A

◆ 下堂教學重點

1-1. Java 發展簡史



1-2. 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

本堂教學重點

1. Java 簡介

1-1. Java 發展簡史

1-2. Java 三大平台

2. Java 語言程式系統架構及特色

2-1. Java SE 架構

Q&A

2-2. Java SE 程式運作

Q&A

3. 安裝和建置 Java 環境

3-1. 下載與安裝 JDK 8

3-2. 環境變數設定

練習1：環境變數設定

4. Java 程式的編譯與執行

4-1. 使用記事本撰寫

練習2：撰寫第一個 Java 程式

4-2. 編譯與執行 Java 程式

練習3：編譯與執行程式

5. Java 程式結構

5-1. 註解

5-2. package (套件)

5-3. import (引用類別)

5-4. class 類別

5-5. 主方法 main()

5-6. 敘述句

Q&A

◆ 下堂教學重點

2-1. Java SE 架構

JDK Java 標準版開發工具：

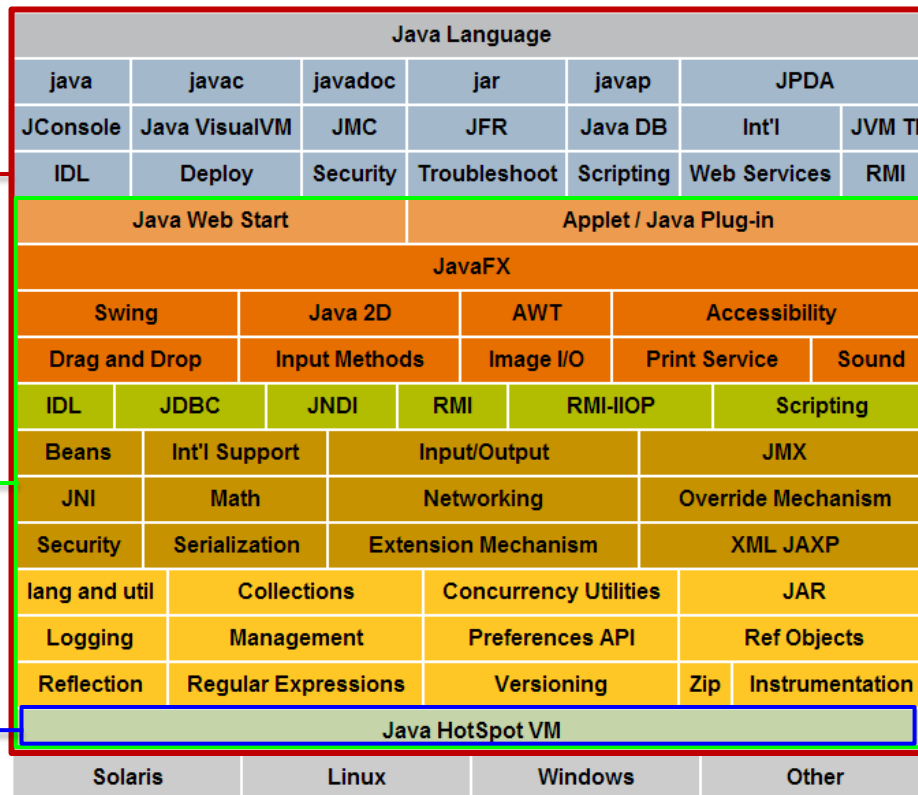
提供編譯器，及相關的開發、執行、測試等工具

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

解釋名詞

- ◆ Java SE (Java Platform, Standard Edition) 平台名稱，如：Java SE 8
- ◆ JDK (Java SE Development Kits) 是基於平台的開發程式發行版本，如：JDK8
- ◆ JRE (Java SE Runtime Environment, JRE) Java 執行環境
- ◆ JVM (Java Virtual Machine) Java 虛擬機器
- ◆ API (Application Programming Interface) 應用程式介面，讓開發者得以憑藉著這組介面，而更便利地開發應用程式
- ◆ IDE (Integrated Develop Environment) 整合開發環境



Q. 對下列英文縮寫，了解幾個？

1. JDK

2. JRE

3. IDE

4. JVM

5. API

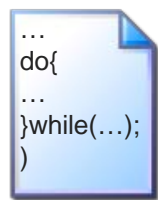
6. Java SE

7. Java EE

8. Java ME

2-2. Java SE 程式運作

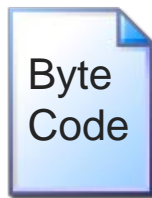
- ◆ Java 為了達到跨平台，採用編譯與直譯混合模式



*.java



Javac 編譯



*.class



Java 直譯執行



Solaris



Java 直譯執行



Linux



Java 直譯執行



Windows

撰寫一次，到處執行 (write once, run anywhere)



Q ■ Java 程式開發與執行順序 (連連看)

1

A 儲存 .java

2

B 產生 .class

3

C 撰寫程式

4

D java 執行程式

5

E javac 編譯

本堂教學重點

1. Java 簡介

1-1. Java 發展簡史

1-2. Java 三大平台

2. Java 語言程式系統架構及特色

2-1. Java SE 架構

Q&A

2-2. Java SE 程式運作

Q&A

3. 安裝和建置 Java 環境

3-1. 下載與安裝 JDK 8

3-2. 環境變數設定

練習1：環境變數設定

4. Java 程式的編譯與執行

4-1. 使用記事本撰寫

練習2：撰寫第一個 Java 程式

4-2. 編譯與執行 Java 程式

練習3：編譯與執行程式

5. Java 程式結構

5-1. 註解

5-2. package (套件)

5-3. import (引用類別)

5-4. class 類別

5-5. 主方法 main()

5-6. 敘述句

Q&A

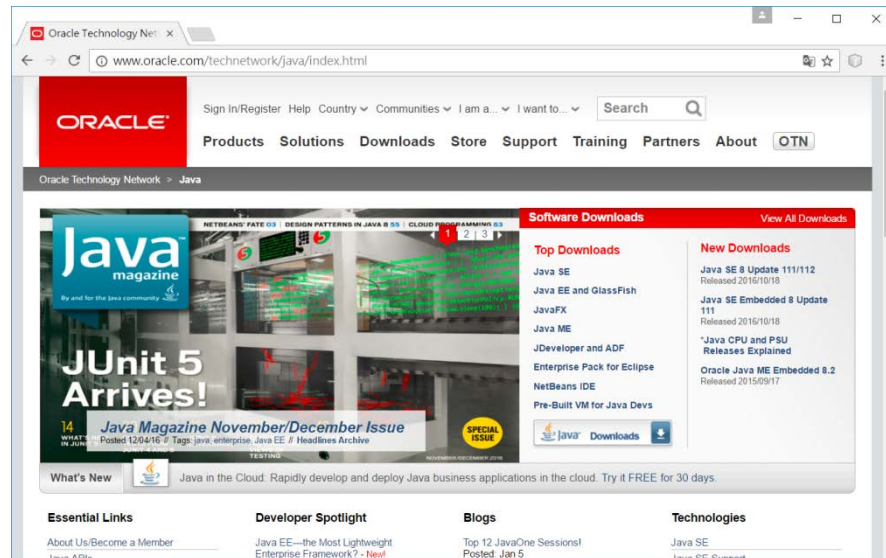
◆ 下堂教學重點



3-1. 下載與安裝 JDK 8

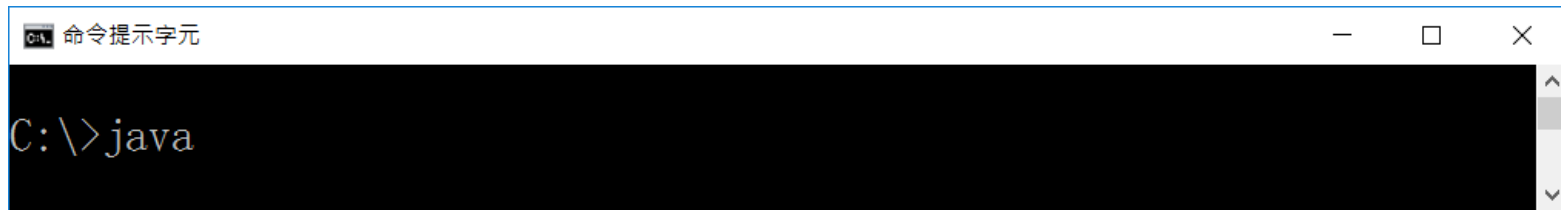
- JDK (Java SE Development Kit, Java 標準版開發套件)

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



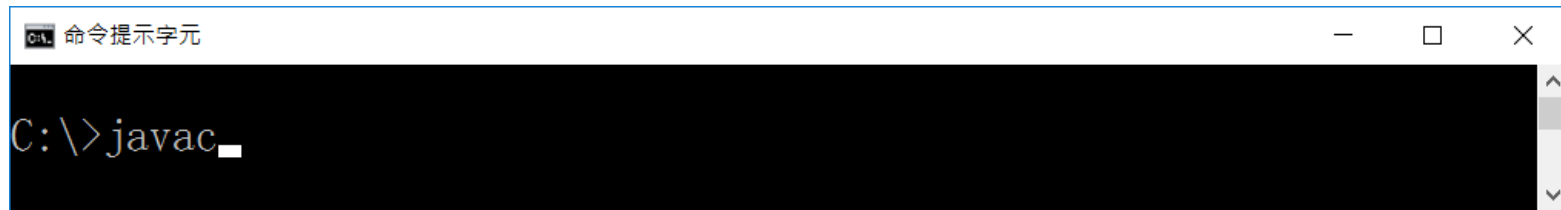
3-2. 環境變數設定

- ◆ 開啟「命令提示字元」測試 Java 執行指令：`java`



```
C:\>java
```

- ◆ 測試 Java 編譯指令：`javac`



```
C:\>javac_
```

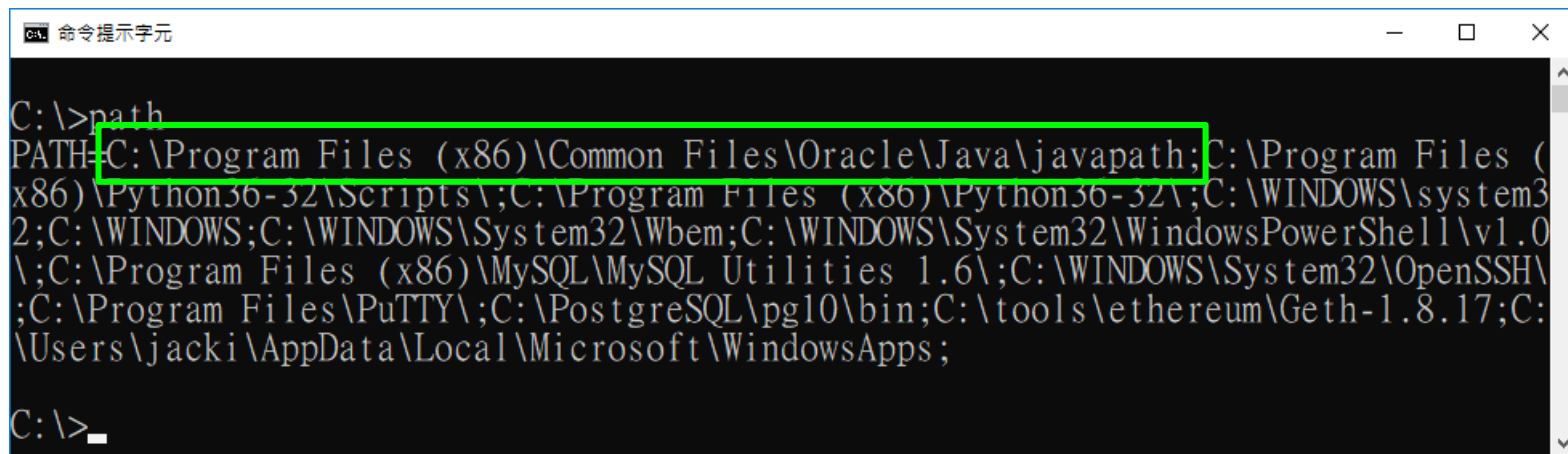
指令測試

```
C:\> javac  
'javac' 不是內部或外部命令、可執行的程式或批次檔。
```



指令測試

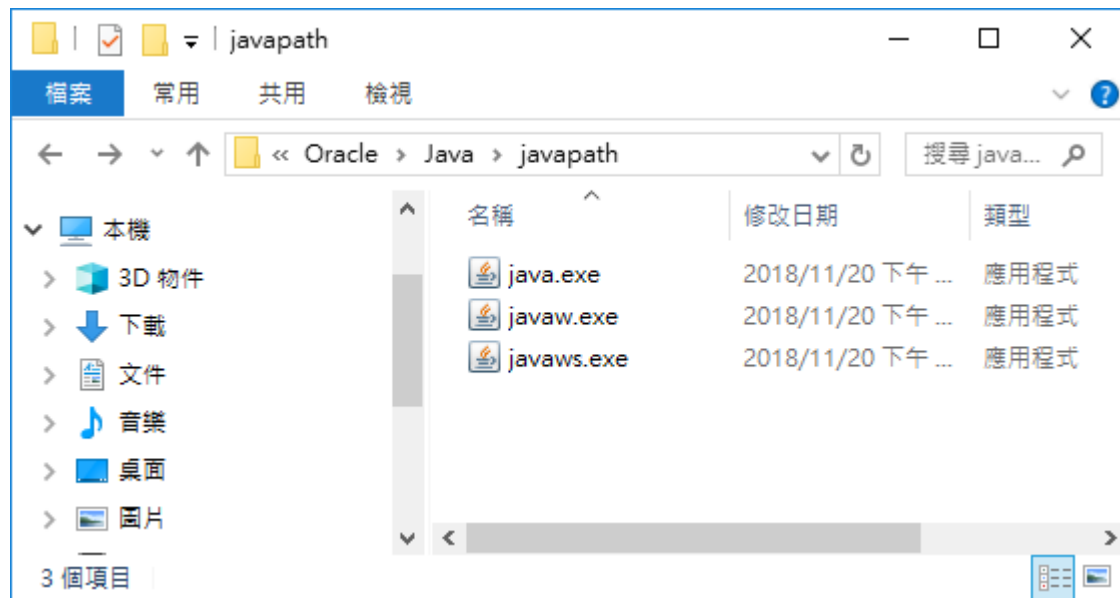
- ◆ 查看設定執行檔的搜尋路徑：[path](#)



```
C:\>path
PATH=C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\Program Files (x86)\Python36-32\Scripts\;C:\Program Files (x86)\Python36-32\;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\WINDOWS\System32\WindowsPowerShell\v1.0\;C:\Program Files (x86)\MySQL\MySQL Utilities 1.6\;C:\WINDOWS\System32\OpenSSH\;C:\Program Files\PuTTY\;C:\PostgreSQL\pg10\bin;C:\tools\ethereum\Geth-1.8.17;C:\Users\jacki\AppData\Local\Microsoft\WindowsApps;
C:\>
```

指令測試

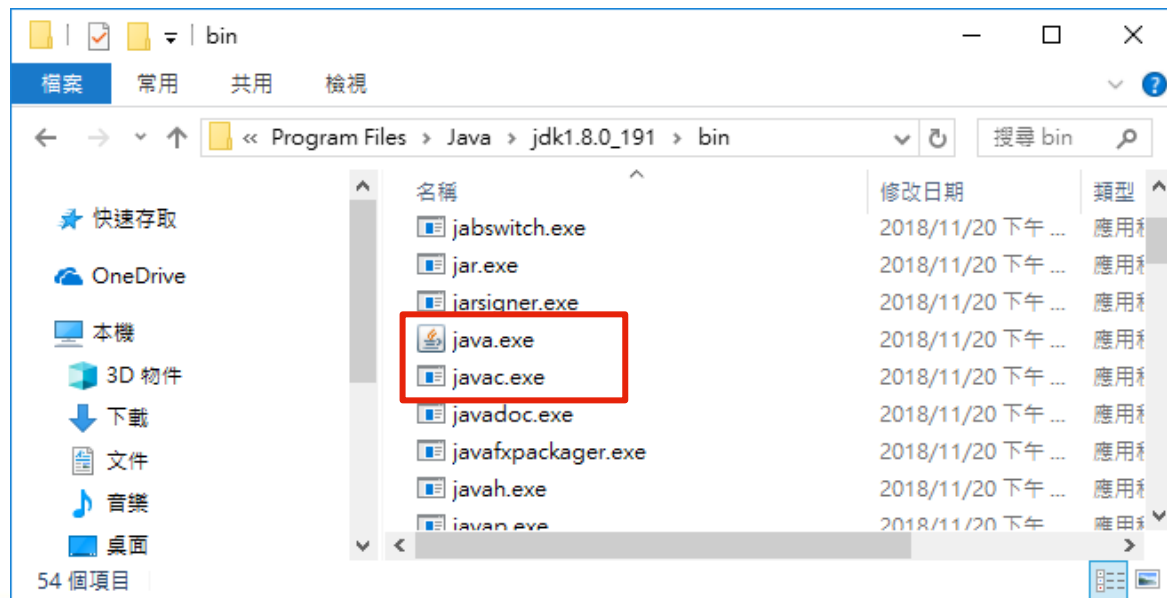
哪個指令
沒看到？



指令測試

◆ JDK 安裝路徑

C:\Program Files\Java\jdk1.8.0_xxx

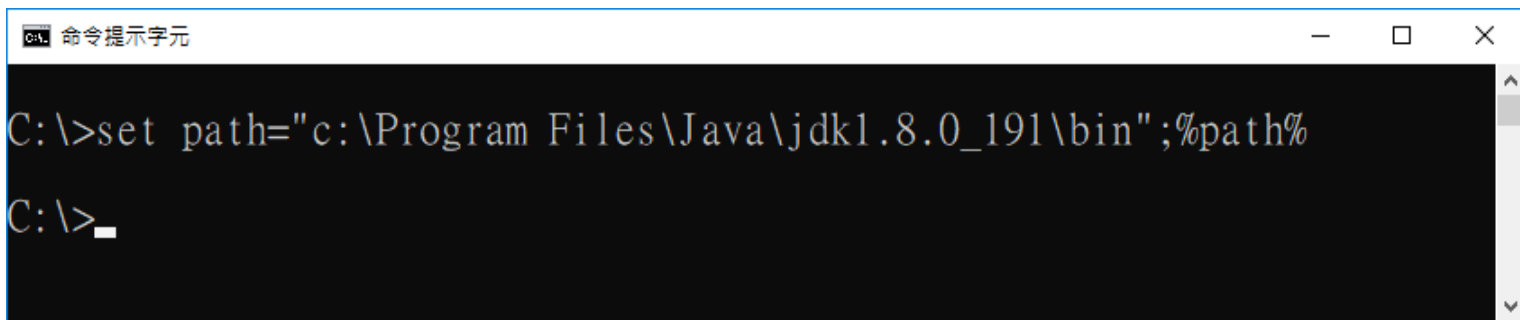


指令測試

- ◆ 開啟「命令提示字元」輸入Java安裝“bin”路徑，設定環境變數：

`set path="c:\Program Files\Java\jdk1.8.0_191\bin";%path%`

- ◆ 再測試：`javac` 與 `java`



```
C:\>set path="c:\Program Files\Java\jdk1.8.0_191\bin";%path%
C:\>_
```

- ◆ 「命令提示字元」視窗關閉，路徑即失效，若關閉「命令提示字元」需在重新開啟後，再輸入一次環境變數

練習1：環境變數設定

● 操作重點

★開啟「命令提示字元」

1. 輸入 Java 執行指令：`java`
2. 輸入 Java 編譯指令：`javac` (會有錯誤訊息)
3. 輸入輸入路徑設定：`set path="c:\Program Files\Java\jdk1.8.0_191\bin";%path%`
4. 再測試 `java` 與 `javac`

本堂教學重點

1. Java 簡介

1-1. Java 發展簡史

1-2. Java 三大平台

2. Java 語言程式系統架構及特色

2-1. Java SE 架構

Q&A

2-2. Java SE 程式運作

Q&A

3. 安裝和建置 Java 環境

3-1. 下載與安裝 JDK 8

3-2. 環境變數設定

練習1：環境變數設定

4. Java 程式的編譯與執行

4-1. 使用記事本撰寫

練習2：撰寫第一個 Java 程式

4-2. 編譯與執行 Java 程式

練習3：編譯與執行程式

5. Java 程式結構

5-1. 註解

5-2. package (套件)

5-3. import (引用類別)

5-4. class 類別

5-5. 主方法 main()

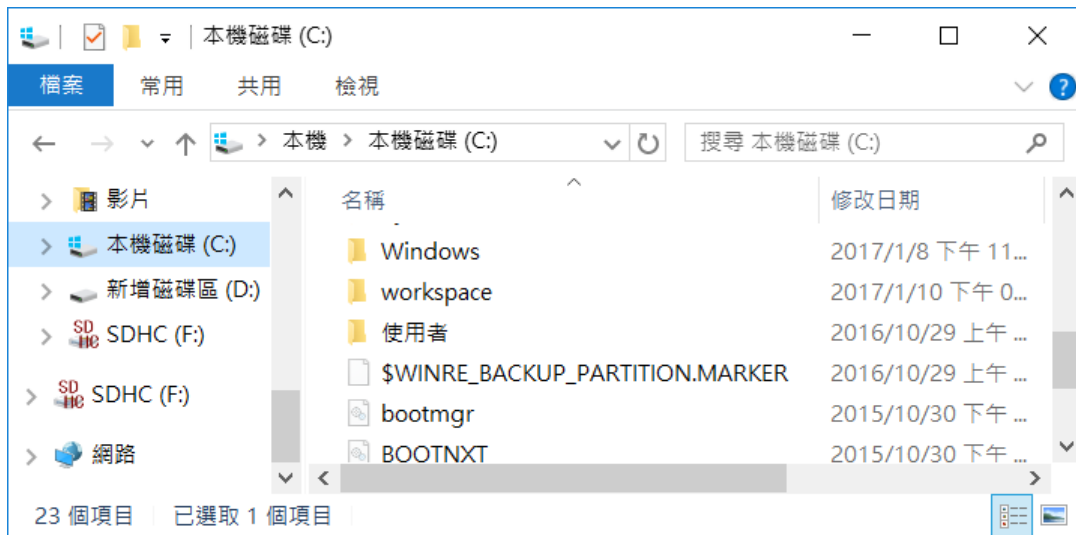
5-6. 敘述句

Q&A

◆ 下堂教學重點

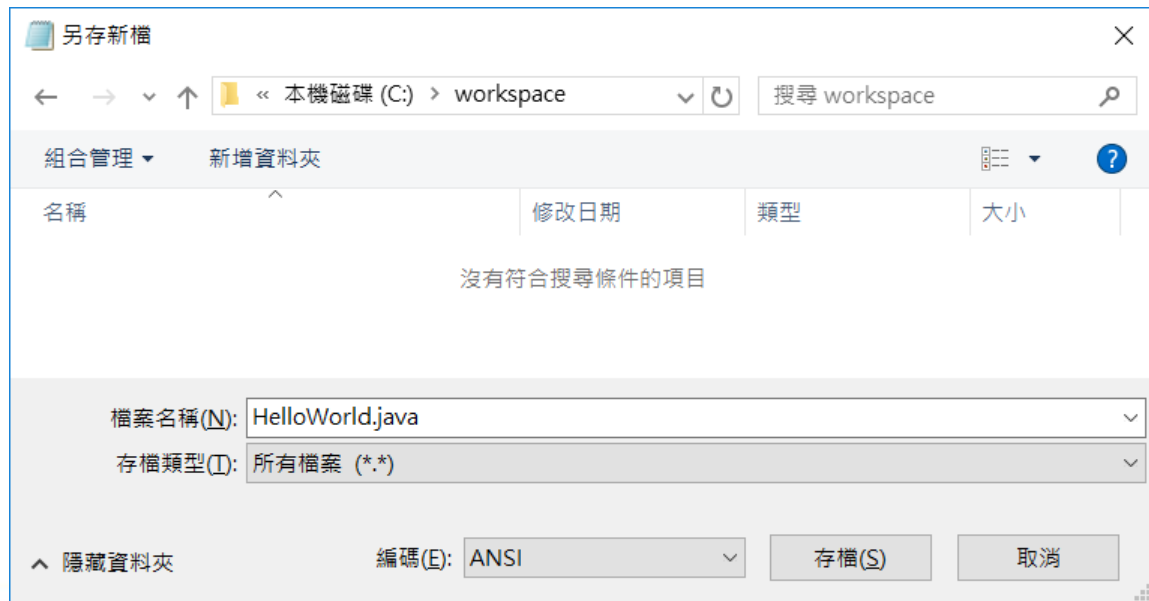
新增工作目錄

- ◆ 建立工作目錄，在 C 磁碟新增資料夾，並命名為：[workspace](#)



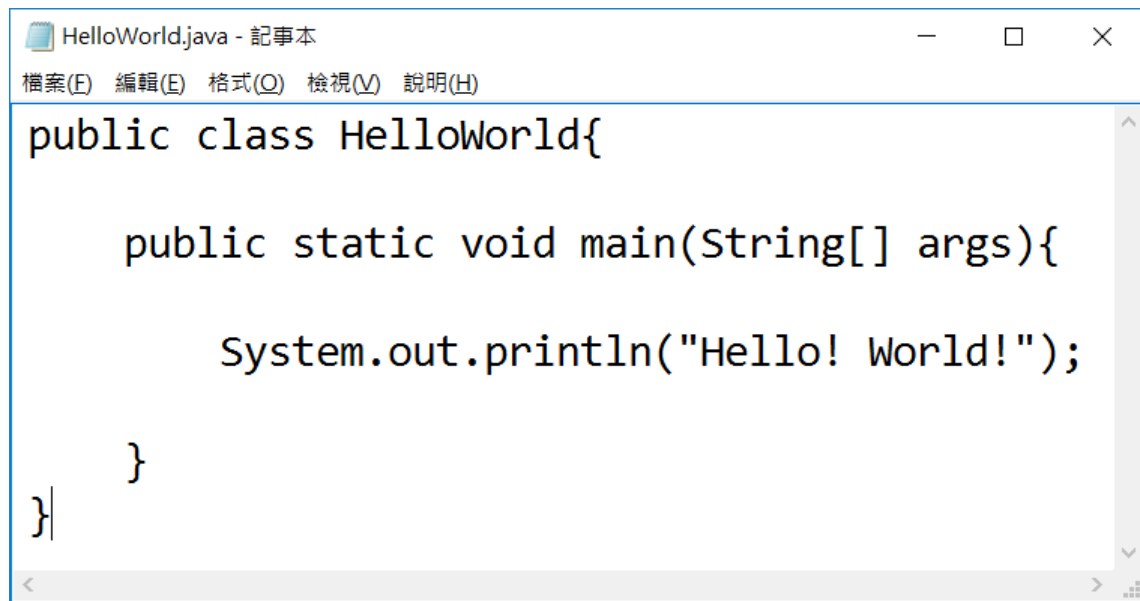
4-1. 使用記事本撰寫

- ◆ 開啟史上最強的程式編輯器：記事本
- ◆ 先另存新檔，檔名為：[HelloWorld.java](#)



HelloWorld.java

◆ 注意英文大小寫

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!");  
  
    }  
}
```

The code is displayed in a monospaced font. The window has a standard menu bar with options: 檔案(E), 編輯(E), 格式(O), 檢視(V), and 說明(H). The window also features standard Windows window controls (minimize, maximize, close) and a scrollbar on the right side.

練習2：撰寫第一個 Java 程式

● 操作重點

★建立工作目錄

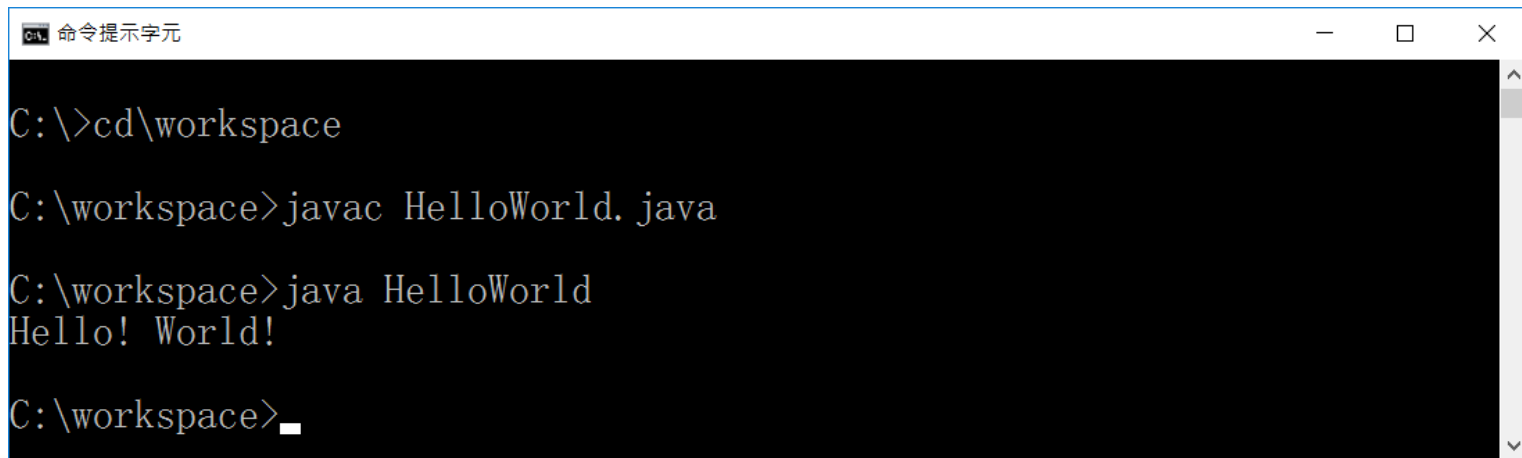
1. 在 C 磁碟新增資料夾，並命名為：`workspace`

★開啟「記事本」

1. 另存先檔，檔名為：`HelloWorld.java` (注意副檔名為 `.java`)
2. 撰寫程式(注意英文大小寫)
3. 存檔

4-2. 編譯與執行 Java 程式

- ◆ 開啟命令提示，切換到 C:\workspace 工作目錄：`cd\workspace`
- ◆ 執行編譯：`javac HelloWorld.java`
- ◆ 執行程式：`java HelloWorld`



```
C:\>cd\workspace

C:\workspace>javac HelloWorld.java

C:\workspace>java HelloWorld
Hello! World!

C:\workspace>_
```

練習3：編譯與執行程式

● 操作重點

★開啟「命令提示字元」

1. 切換工作目錄：`cd\workspace`
2. 輸入編譯指令：`javac HelloWorld.java`
3. 輸入執行指令：`java HelloWorld`

本堂教學重點

1. Java 簡介

1-1. Java 發展簡史

1-2. Java 三大平台

2. Java 語言程式系統架構及特色

2-1. Java SE 架構

Q&A

2-2. Java SE 程式運作

Q&A

3. 安裝和建置 Java 環境

3-1. 下載與安裝 JDK 8

3-2. 環境變數設定

練習1：環境變數設定

4. Java 程式的編譯與執行

4-1. 使用記事本撰寫

練習2：撰寫第一個 Java 程式

4-2. 編譯與執行 Java 程式

練習3：編譯與執行程式

5. Java 程式結構

5-1. 註解

5-2. package (套件)

5-3. import (引用類別)

5-4. class 類別

5-5. 主方法 main()

5-6. 敘述句

Q&A

◆ 下堂教學重點

Java 程式結構

- ◆ 此階段先認識 Java
主類別結構與撰寫

```
/*
 * 檔名：MainClass.java
 * 用途：認識 Java 程式結構
 */
package com.sample;

import java.lang.*;

//主類別
public class MainClass {
    /** 主方法，程式進入點 */
    public static void main(String[] args) {
        System.out.println("Hello!World!");
    }
}
```

5-1. 註解

多行註解
不可以寫成巢狀註解

```
/*  
 * 檔名：MainClass.java  
 * 用途：認識 Java 程式結構  
 */
```

```
package com.sample;
```

```
import java.lang.*;
```

單行註解

```
//主類別
```

文件註解

```
public class MainClass {  
    /** 主方法，程式進入點 */  
    public static void main(String[] args) {  
        System.out.println("Hello!World!");  
    }  
}
```

5-1. 註解

◆ 單行註解：單一行註解或說明文字

◇ //註解文字

```
class Test{           //一般類別  
}
```

◆ 多行註解：多行註解或說明文字，不可寫巢狀註解

◇ /* 註解文字1

註解文字2

註解文字3

*/

```
/*  
    檔名：MainClass.java  
    用途：認識 Java 程式結構  
*/
```


5-1. 註解

- ◆ 文件註解：將註解資訊透過 javadoc.exe 產生 Java API 說明文件

◆ `/** ... */`

```
/**主方法，程式進入點*/  
public static void main(String[] args){  
    System.out.println("Hello! World!");  
}
```

All Methods

Static Methods

Concrete Methods

Modifier and Type

Method and Description

static void

`main(java.lang.String[] args)`

主方法，程式進入點

5-2. package (套件)

關鍵字 package (套件)

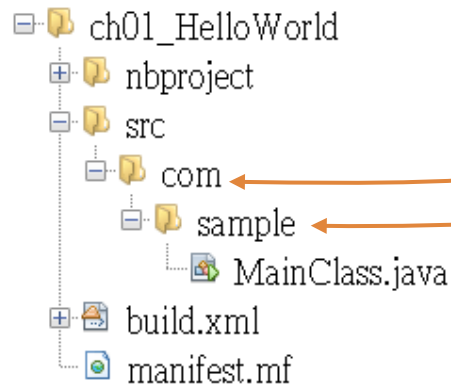
```
/*
 * 檔名：MainClass.java
 * 用途：認識 Java 程式結構
 */
package com.sample;

import java.lang.*;

//主類別
public class MainClass {
    /** 主方法，程式進入點 */
    public static void main(String[] args) {
        System.out.println("Hello!World!");
    }
}
```

5-2. package (套件)

- ◆ Java 使用 package 來管理類別，用來宣告該檔案中的類別是屬於哪一個 package，必須寫在程式的第一行。
- ◆ package 套件的敘述可以和檔案系統結構相對應。



```
package com.sample;
```

5-3. import (引用類別)

關鍵字 import (引用類別)

```
/*
 * 檔名：MainClass.java
 * 用途：認識 Java 程式結構
 */
package com.sample;

import java.lang.*;

//主類別
public class MainClass {
    /** 主方法，程式進入點 */
    public static void main(String[] args) {
        System.out.println("Hello!World!");
    }
}
```

5-3. import (引用類別)

- ◆ 引用現成 **API** 或廠商或自己開發的類別時，可以先在 **import** 區宣告，告知編譯器要使用的類別是位於哪一個套件，讓編譯器找到這些類別所在的位置。
- ◆ **Java.lang** 是預設引用的套件，日後使用此套件下的類別可以直接使用，因此可以省略 **import** 的敘述。
- ◆ **import** 敘述寫在 **package** 後方，類別宣告的前方，可以有多個。

5-4. class 類別

類別 class

```
/*
 * 檔名：MainClass.java
 * 用途：認識 Java 程式結構
 */
package com.sample;

import java.lang.*;

//主類別
public class MainClass {
    /** 主方法，程式進入點 */
    public static void main(String[] args) {
        System.out.println("Hello!World!");
    }
}
```

5-4. class 類別

- ◆ Java 程式由類別組成，因此 Java 程式至少須包含一組類別的宣告。

```
public class MainClass{ }
```

- ◆ public 存取修飾字，表示公開使用
- ◆ class 宣告類別關鍵字
- ◆ MainClass 類別名稱(自訂)
- ◆ [注意]：類別若宣告public，則存檔時的主檔名必須與類別名稱相同
一個 java 檔案可以有多的類別宣告，但只有一個可以宣告 public。


5-5. 主方法 main()

```
/*
 * 檔名：MainClass.java
 * 用途：認識 Java 程式結構
 */
package com.sample;

import java.lang.*;

//主類別
public class MainClass {
    /** 主方法，程式進入點 */
    public static void main(String[] args) {
        System.out.println("Hello!World!");
    }
}
```

main() 程式進入點



5-5. 主方法 main()


- ◆ Java 應用程式執行時的進入點。

```
public static void main(String[] args){ }
```

- ◆ `public` 存取修飾字，表示公開使用
- ◆ `static` 代表 `main` 方法是類別成員，不需要產生物件就能被執行
- ◆ `void` 表示 `main()` 執行結束後不傳回任何值
- ◆ `main` 方法名稱(不可以修改)
- ◆ `String[] args` 方法參數列：`String[]` 字串陣列，`args` 陣列名稱，在此用來接收命令列傳入的參數

5-6. 敘述句

敘述句



```
/*
 * 檔名：MainClass.java
 * 用途：認識 Java 程式結構
 */
package com.sample;

import java.lang.*;

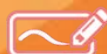
//主類別
public class MainClass {
    /** 主方法，程式進入點 */
    public static void main(String[] args) {
        System.out.println("Hello!World!");
    }
}
```

5-6. 敘述句

- ◆ 敘述句代表程式中的一行指令，敘述句的結束要用分號「；」

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

- ◇ 在上面的敘述句中，使用了 `java.lang` 套件下的 `System` 類別的公開成員 `out` 物件，`out` 是一個 `PrintStream` 物件，在此使用了 `PrintStream` 所提供的 `println()` 方法，將指定的字串“Hello!World!”從螢幕輸出顯示。



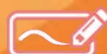
Q . 何者宣告正確？

A. `import java.io.*;`
`import java.lang.*;`
`class MyClass{...}`
`package com.sample;`

B. `import java.io.*;`
`import java.lang.*;`
`package com.sample;`
`class MyClass{...}`

C. `package com.sample;`
`class MyClass{...}`
`import java.io.*;`
`import java.lang.*;`

D. `package com.sample;`
`import java.io.*;`
`import java.lang.*;`
`class MyClass{...}`



Q . 第2行需插入哪一個項目的程式

```
1    public class Test{  
2        // 插入程式  
3        System.out.println("Java程式開發!");  
4    }  
5 }
```

- A. Static void main(String[] args){
- B. public static void Main(String[] args){
- C. public static void main(string[] args){
- D. public static void main(){
- E. public static void main(String[] args){

下堂教學重點

- ◆ 物件導向程式設計概念
- ◆ 建立類別與方法
- ◆ Java 名稱規則
- ◆ Java 資料型別與變數宣告
- ◆ 基本型別
- ◆ 使用常數

問卷

<http://www.pcschoolonline.com.tw>

開課查詢

免費體驗專區

課程總覽

專業師

1

學員專區

講師專區



➤ 課程檔案下載：

學員的「上課教材」，下載檔案為壓縮檔 ([解壓縮操作步驟](#))。
如無法觀看上課教材，請安裝 [PDF閱讀軟體](#)。

公告專區

我的課表

課程劃位

取消劃位

2

課程檔案下載

自107年1月1日起，課程錄影檔由180天改為365天(含)內無限次觀看 (上課隔日18:00起)。

問
卷

上課日期	課程名稱	課程節次	教材下載		
2017/12/27 2000 ~ 2200	線上真人-ZBrush 3D動畫造型設計	18	上課教材	錄影檔	課堂問卷
2017/12/20 2000 ~ 2200	線上真人-ZBrush 3D動畫造型設計	17	上課教材	錄影檔	
2017/12/18 2000 ~ 2200	線上真人-ZBrush 3D動畫造型設計	16	上課教材	錄影檔	



巨匠線上真人

www.pcschoolonline.com.tw