

Java

輸出語句



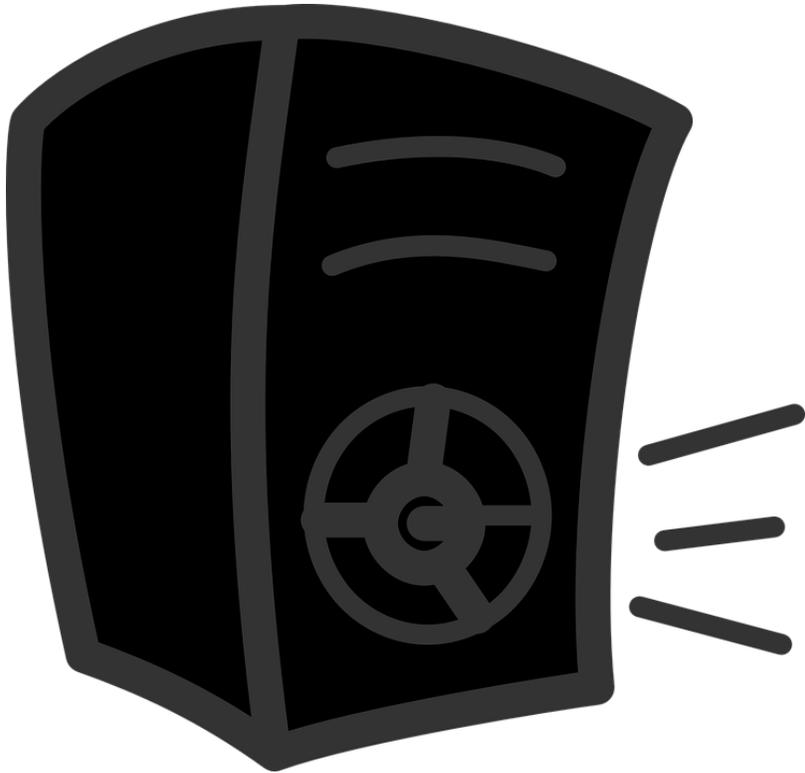
電腦是被製造出來幫忙人類的工具，
讓電腦把它所知道的東西告訴我們是非常重要的



電腦把它想說的告訴我們，有兩種方法

讓電腦用喇叭發出聲音

用螢幕輸出文字



請電腦印出
Hello, World!

問題



先提供一個 **框架**

```
class web {  
    public static void main(String[] args) {  
        → 程式碼寫在這 ←  
    }  
}
```



使用**印**的指令

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

組合『**框架** + **印**的指令』

```
class web {  
    public static void main(String[] args) {  
        → 程式碼寫在這 ←  
    }  
}
```

將印的指令放
到這個位置

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

框架 + 印的指令 = 輸出 Hello, World!

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



執行結果

Hello, World!

一些注意事項

- 紅色的字，會被印出來
 - `System.out.println("Hello, World!");`

如果印你好
該怎麼辦？



一些注意事項

- 紅色的字，會被印出來
 - `System.out.println("Hello, World!");`

```
class web{  
    public static void main(String[] args) {  
        System.out.println("你好");  
    }  
}
```



如果印你好
該怎麼辦？



一些注意事項

- 一句話的結束時要有一個結束符號

- 中文用 。

你好，程式語言。

- 英文用 。

Hi, program.

- 電腦語言用 ;

System.out.println("Hi, Java");

需要一個**框架**

- `System.out.println("Hello, World!");`
只寫這樣是印不出資料的！
- 所有的 Java 語言都需要**框架**
 - 如同網球比賽，球要掉在界內才算得分
 - 只有放在框架裡面的 Java 程式才是可以執行的

JAVA

延伸的概念

概念1：換行不換行？(1)

- 每一個 `System.out.println();` 印完都會自動換行
- 不想換行？
 - `System.out.print("Hello");`



In 就如同 line 的縮寫，
加上 In 的 print 就會
自己變成一行喔！

不換行：`System.out.print("Hello");`

換行：`System.out.println("Hello");`

概念1：換行不換行？(2)

- 每一個 `System.out.println();` 印完都會自動換一行
- 一次想換兩行呢？
 - `System.out.println("Hello\n");`
 - `System.out.print("Hello\n\n");`
- **\n 是換行符號！**
 - 使用 `\n` 是在跟程式說請先換一行之後再繼續顯示。
 - 兩個 `\n` 就會換兩行。

概念1：換行不換行？(3)

- 不換行
 - System.out.**print**("Hello");
- 換一行
 - System.out.**println**("Hello");
 - System.out.**print**("Hello\n");
- 換兩行
 - System.out.**println**("Hello\n");
 - System.out.**print**("Hello\n\n");

\n 是一種換行符號，可以增加 println() 與 print() 的使用彈性。

print() 與 println() 是一種顯示工具。

兩者千萬不要搞混了喔！



概念2：不說出的話(1)

```
class web{  
    public static void main(String[] args) {  
        System.out.println("Hello, World!"); // 這是單行註解: 這行會印出  
        /*  
        這是多行註解，這裡面的都不會印出  
        System.out.println("Hello, Java!"); 這行不會印出  
        */  
        System.out.println("Hello, Dice!"); // 這是單行註解: 這行會印出  
    }  
}
```

不是所有的 System.out.println 都會說話，放在註解區內的 System.out.println 是不說話的唷！



執行結果

```
dice$ java web  
Hello, World!  
Hello, Dice!
```

概念2：不說出的話(2)

註解是給人看的，
不是給電腦看的！

```
class web{  
    public static void main(String[] args) {  
        System.out.println("Hello, World!"); //這是單行註解:這行會印出  
        /*  
            這是多行註解，這裡面的都不會印出  
            System.out.println("Hello, Java!"); 這行不會印出  
        */  
        System.out.println("Hello, Dice!"); //這是單行註解:這行會印出  
    }  
}
```

註解就像閱讀者黏貼在書中內頁的便條紙筆記一樣，不會影響原書的表達，但可以讓閱讀者更快速的了解該書要傳達的意思。



概念2：不說出的話(3)

- 註解有兩種：
 - 多行註解: 前面先放 `/*`，後面再放 `*/`
 - 包在 `/* */` 之間的所有文字都是註解
 - 單行註解: 使用兩個雙斜線 `//` 開頭
 - `//` 後面的文字都是註解
 - `//` 前面的文字會當作正常的程式碼執行
 - `//` 前面也可以沒有文字！

概念3：執行 java 程式 (1)

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

Java 程式檔案: web.java

第一步: 編輯 java 程式

- Java 程式附檔名: **java**
- Java 程式檔名必須和檔案內的 class 後面接的**名稱一致** (**大小寫**也要一致!)
 - class web
 - web.java

Dice 程式學習系統內預設會將檔名存為 web.java，因此在練習程式題目時務必使用 class web {}，若在其他編譯環境就可隨意命名。

如：Hello.java: class Hello {}
test.java: class test{}

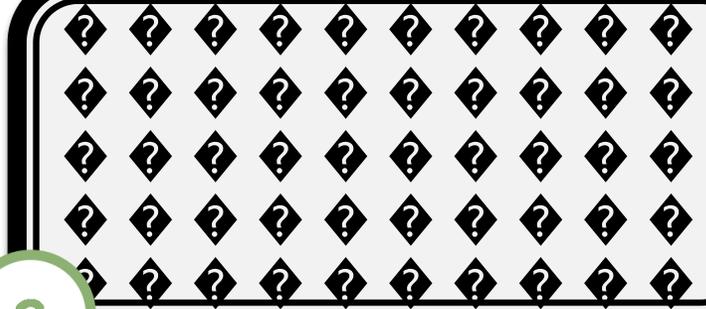


概念3：執行 java 程式 (2)

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

1 Java 程式檔案: web.java

javac web.java



2

二進位檔: web.class

第二步: 編譯 java 程式

– 使用 **javac** 指令產生 **.class** 檔案

- .class 檔案

- .class 檔案是一種 binary 檔案(二進位檔)，是給電腦看得，不是給人讀的！
- .class 的檔名會與 .java 檔案內的 class 名稱一樣，加上第一個步驟所提的『Java 程式檔名必須和檔案內的 class 後面接的名稱一致』，web.java 經過 javac 指令會產生 web.class，Hello.java 經過 javac 指令會產生 Hello.class。

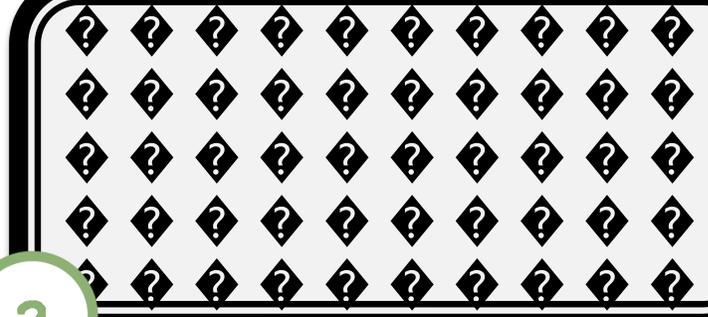
- javac 指令

- 要執行 javac 指令，需先在系統安裝 [JDK](#)(Java Development Kit)，JDK 是用來開發 Java 程式的開發工具組。

概念3：執行 java 程式 (3)

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

javac web.java



二進位檔: web.class

java
web

Hello, World!

1 Java 程式檔案: web.java

第三步: 執行 class 檔案

- 使用 **java** 指令執行 **.class** 檔案
- 執行第二步驟產生的 **.class** 檔案就可以執行編寫的 java 程式了！

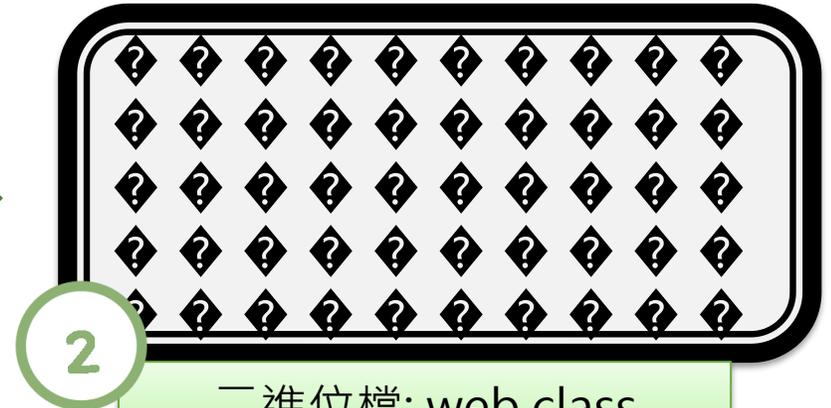
3 執行畫面

概念3：執行 java 程式 (4)

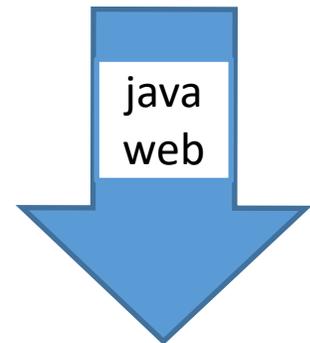
```
class web{  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

1

Java 程式檔案: web.java



二進位檔: web.class



Hello, World!

3

執行畫面



第一步: 編輯 **java** 程式

第二步: 編譯 **java** 程式

第三步: 執行 **class** 檔案