

# 編輯器與 第一個 Java 程式

TYIC桃高資訊社

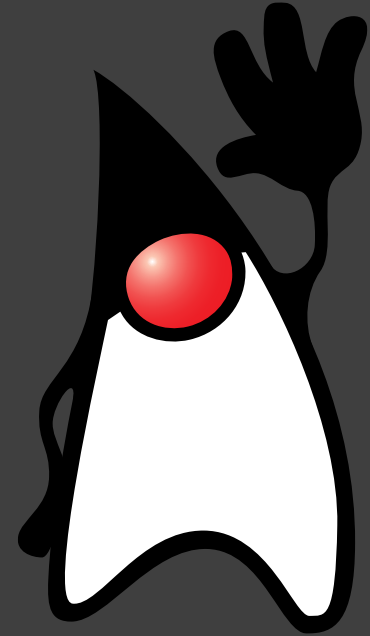
# Java 介紹



Java 之父：  
詹姆斯·高斯林  
(James Gosling)



Java 名稱由來：  
創始人想名字時拿著  
印度Java島的咖啡



Java 吉祥物：  
名叫 Duke

# Java 介紹



Java 創造公司：  
昇陽電腦(Sun)  
1995年創造 Java



Java 領導組織：  
甲骨文公司(Oracle)  
2010 收購昇陽電腦



和 JavaScript 的關係：  
只是當時 Java 很熱門  
JavaScript 就借鑑名字

# 第一個 Java 程式

學習一門程式語言，從 Hello World! 開始

```
// 類別(class)名稱必須跟檔案名稱一樣  
public class Main {
```

```
    // Java 程式需要一個主方法(main 方法)，程式從這裡開始執行  
    public static void main(String[] args) {
```

```
        // 在 Java 中，使用 System.out.println() 來輸出資料  
        System.out.println("Hello, World!");
```

```
    }
```

```
}
```



java

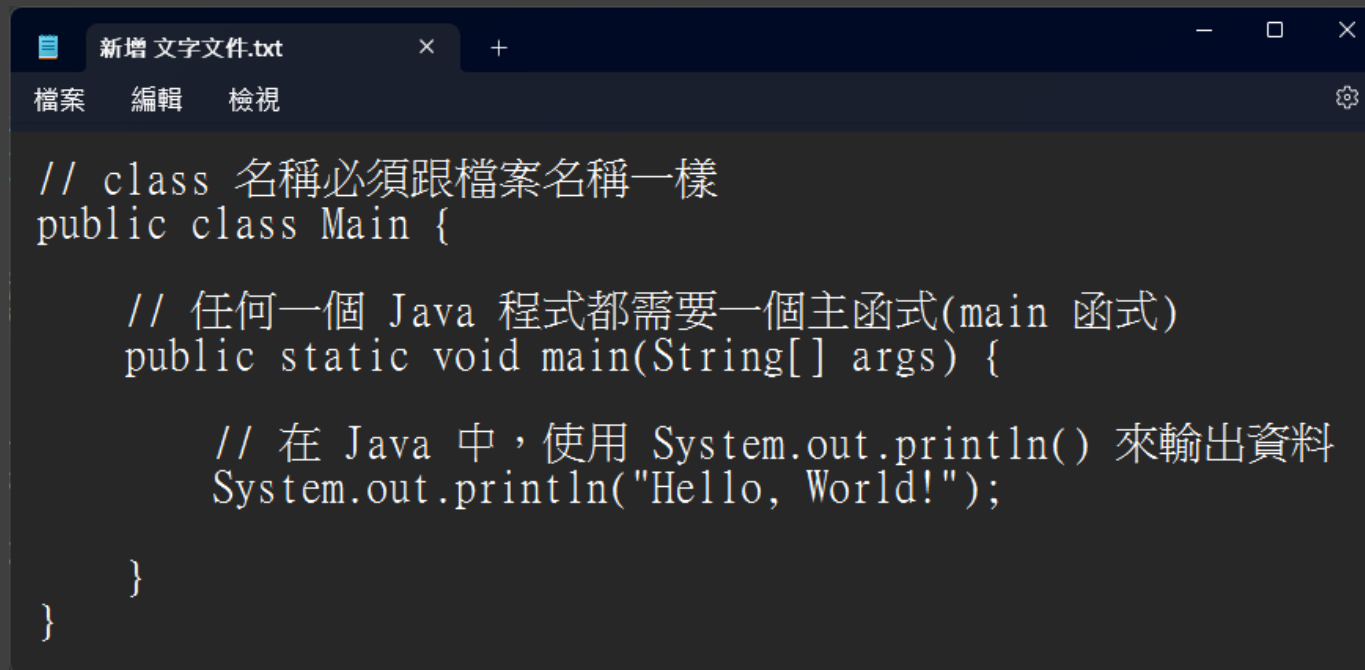
# 編輯器

要寫在哪裡？用什麼寫？

只要想辦法把程式寫在純文字檔案中即可

所以用「**Windows 記事本**」把程式寫在「新增 文字文件.txt」

也完全是可以的喔

A screenshot of a Windows Notepad window. The title bar shows '新增 文字文件.txt' (New Text Document.txt) with standard window controls. The menu bar includes '檔案' (File), '編輯' (Edit), and '檢視' (View). The text area contains the following Java code:

```
// class 名稱必須跟檔案名稱一樣
public class Main {

    // 任何一個 Java 程式都需要一個主函式(main 函式)
    public static void main(String[] args) {

        // 在 Java 中，使用 System.out.println() 來輸出資料
        System.out.println("Hello, World!");

    }
}
```

# 編輯器

功能少  
易上手

## 文字編輯器 (text editor)

顧名思義，用來編輯文字  
不限於程式語言，想寫什麼都可以  
如：`vi`、`nano`、`emacs`  
、`notepad`(Windows 記事本)

有些還可以加入插件(plugin)  
實現有自動補全(auto-complete)  
如：`Vim`、`Neovim`、`Notepad++`  
、`Python IDLE`、`Atom`、`VScode`

功能多  
不易上手

## 整合式開發環境 (Integrated Development Environment，簡稱 IDE)

通常是針對特定的程式語言設計  
並且整合了許多東西，包含：  
文字編輯器、除錯器(debugger)、  
自動組建工具(build automation)  
，部分還有版本控制系統(Version  
Control System，簡稱 VCS)  
如：`Spyder`、`Visual Studio`、  
`Code::Blocks`、`Dev-C++`、  
`Eclipse`、`IntelliJ IDEA`

# 編輯器

功能少  
易上手

## 文字編輯器 (text editor)

顧名思義，用來編輯文字  
不限於程式語言，想寫什麼都可以

如：vi、nano、emacs  
、notepad(Windows 記事本)

有些還可以加入插件(plugin)  
實現有自動補全(auto-complete)

如：Vim、Neovim、Notepad++  
、Python IDLE、Atom、VScode



```
pwsh in 01

// class 名稱必須跟檔案名稱一樣
public class Main {

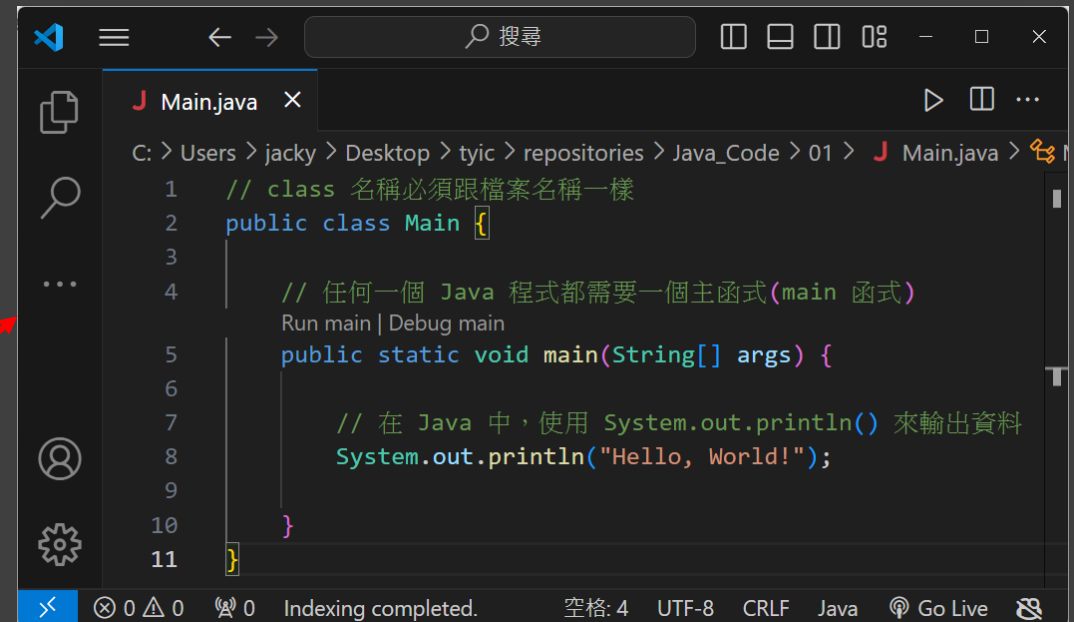
    // 任何一個 Java 程式都需要一個主函式(main 函式)
    public static void main(String[] args) {

        // 在 Java 中，使用 System.out.println() 來輸出資料
        System.out.println("Hello, World!");

    }

}

JavaSE-17 OK
```



```

Main.java x
C: > Users > jacky > Desktop > tyic > repositories > Java_Code > 01 > Main.java >
1 // class 名稱必須跟檔案名稱一樣
2 public class Main {
3
4 // 任何一個 Java 程式都需要一個主函式(main 函式)
5 Run main | Debug main
6 public static void main(String[] args) {
7
8 // 在 Java 中，使用 System.out.println() 來輸出資料
9 System.out.println("Hello, World!");
10
11 }
```

# 編輯器

功能多  
不易上手

## 整合式開發環境 (Integrated Development Environment, 簡稱 IDE)

通常是針對特定的程式語言設計  
並且整合了許多東西，包含：  
文字編輯器、除錯器(debugger)、  
自動組建工具(build automation)  
，部分還有版本控制系統(Version Control System, 簡稱 VCS)  
如：PyCharm、Visual Studio、  
Code::Blocks、Dev-C++、  
Eclipse、**IntelliJ IDEA**



```
1 // class 名稱必須跟檔案名稱一樣
2 public class Main {
3
4     // 任何一個 Java 程式都需要一個主函式(main 函式)
5     public static void main(String[] args) {
6
7         // 在 Java 中，使用 System.out.println() 來輸出資料
8         System.out.println("Hello, World!");
9
10    }
11 }
```



# 第一個 Java 程式

讓我們用 Notepad++ 撰寫我們的第一個 Java 程式

Notepad++ 是個體積小、易使用的文字編輯器

```
1 // class 名稱必須跟檔案名稱一樣
2 public class Main {
3
4     // 任何一個 Java 程式都需要一個主函式(main 函式)
5     public static void main(String[] args) {
6
7         // 在 Java 中, 使用 System.out.println() 來輸出資料
8         System.out.println("Hello, World!");
9
10    }
11 }
```

## Notepad++ 小百科



作者：侯今吾(台灣人)  
自由開源軟體  
曾被票選為最佳開發工具  
使用 C++ 編寫  
作者言論很火爆

# 第一個 Java 程式

請使用 Notepad++ 「親手」打出下列的程式碼

```
// 類別(class)名稱必須跟檔案名稱一樣
public class Main {

    // Java 程式需要一個主方法(main 方法)，程式從這裡開始執行
    public static void main(String[] args) {

        // 在 Java 中，使用 System.out.println() 來輸出資料
        System.out.println("Hello, World!");

    }

}
```



java

# 編譯(Compile)與執行(Run)

Java 是個跨平臺語言，程式編譯後便可以在所有安裝 Java 的地方執行  
Java 程式須經過編譯器(compiler)編譯成位元組碼(bytecode)  
才能被 Java 虛擬機(Java Virtual Machine，簡稱JVM)執行

```
// 類別(class)名稱必須跟檔案名稱一樣
public class Main {

    // Java 程式需要一個主方法(main 方法)，程式從這裡開始執行
    public static void main(String[] args) {

        // 在 Java 中，使用 System.out.println() 來輸出資料
        System.out.println("Hello, World!");
    }
}
```

java

程式碼

編譯



0110  
1001  
1010

Java  
bytecode

執行



# 補充：編譯與執行-C/C++

C/C++ 程式碼會直接被編譯器編譯成機器碼

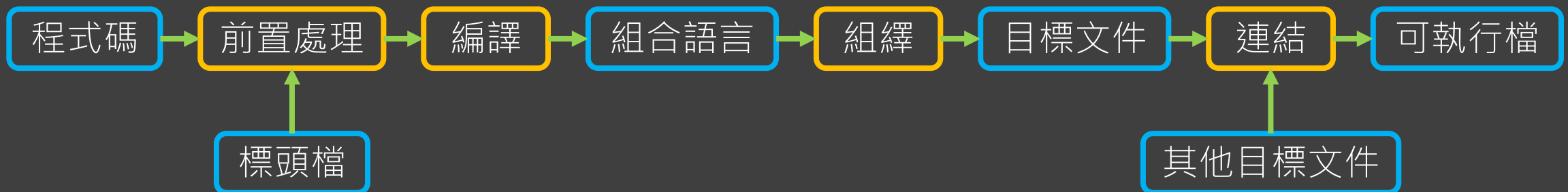
所以 C/C++ 無法一次編譯到處執行

只能在同系統、同CPU指令集(**Instruction Set**)的電腦上執行

如在 **Windows** 上編譯的可執行檔無法在 **Linux** 上執行

但在 **Intel Core i9** 上編譯的可執行檔可在 **AMD Ryzen** 執行

因為他們皆使用 **x86\_64** (**x64**、**Intel 64**、**AMD 64**) 指令集



# 補充：編譯與執行-Python

**Python** 程式不需要經過編譯

而是經由直譯器(**interpreter**)執行一行翻譯一行

所以執行起來速度與效率相較 **C/C++** 慢許多

因為執行時花很多時間在翻譯上

類似的語言還有 **JavaScript** 等腳本語言(**Scripting language**)

# 編譯與執行

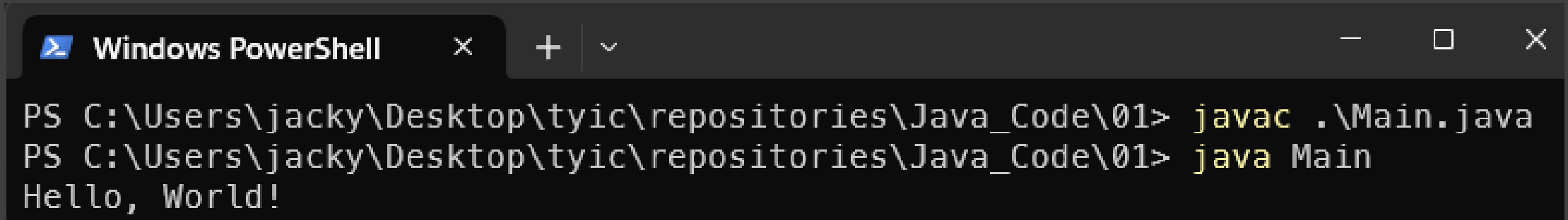
在 `bash/cmd/PowerShell` 中，依序使用下列指令來編譯和執行：

編譯 `javac Main.java`

執行 `java Main`

bash

注意第二條指令(執行程式指令)後面沒有 `".java"` 喔！



```
Windows PowerShell
PS C:\Users\jacky\Desktop\tyic\repositories\Java_Code\01> javac .\Main.java
PS C:\Users\jacky\Desktop\tyic\repositories\Java_Code\01> java Main
Hello, World!
```

`".\Main.java"` 中 `."` 指的是「當前資料夾」，`"\"` 指的是「資料夾內的東西」，所以 `".\Main.java"` 和 `"Main.java"` 等效

# 開啟 PowerShell/cmd

在資料夾位置開啟 PowerShell：

1. **shift + 右鍵** 彈出選單
2. 選擇「在這裡開啟 PowerShell 視窗」

在資料夾位置開啟 cmd：

在上方路徑欄輸入 "cmd" 然後按下 enter

