

第一章

認識Java

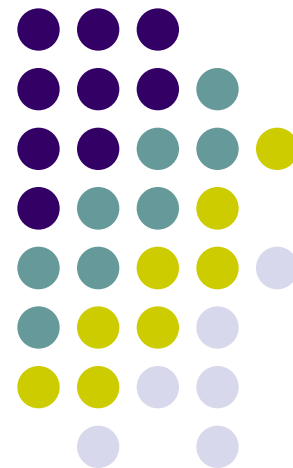
Java的發展歷史

Java的虛擬機器(JVM)

Java未來的發展

撰寫第一個Java程式

安裝、設定VSCode





Java的歷史

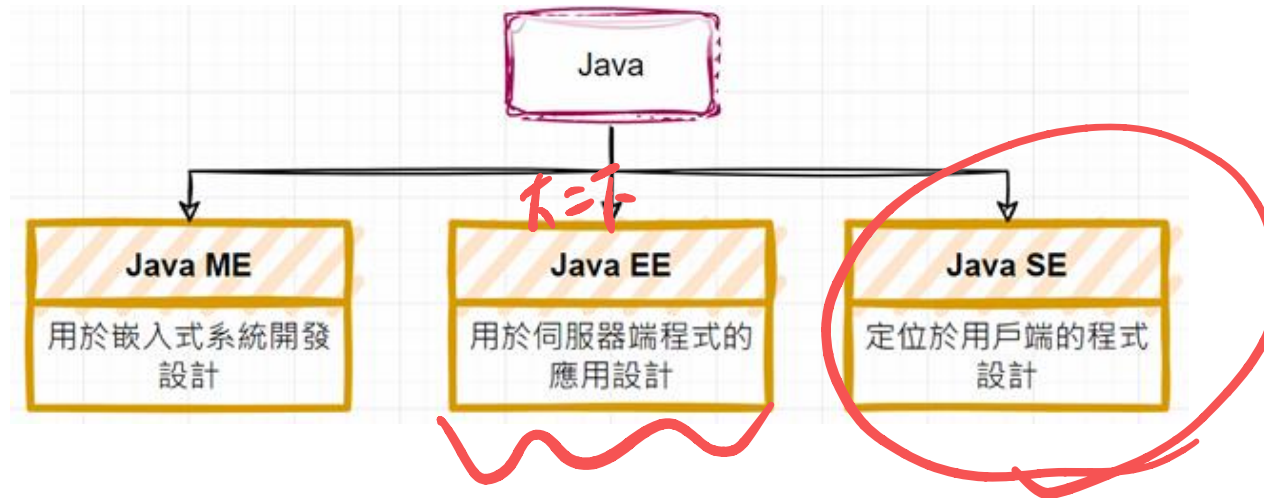
- Java於1995年誕生，由美國加州的昇陽電腦公司（Sun Microsystems, Inc.）所推出
 - Java語言的前身是Oak，即橡樹之意
 - Sun將Oak修改成為Java，並將它轉移到Web上
 - Web於1995年開始盛行，Java也隨之一炮而紅
 - 2009年4月，SUN被Oracle（甲骨文）公司併購





Java的應用領域

- 下圖為Java的三大應用領域：



Micro Edition, Enterprise Edition, Standard Edition



Java的特點 (1/2)

- 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致
- 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 (2/2)

- 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



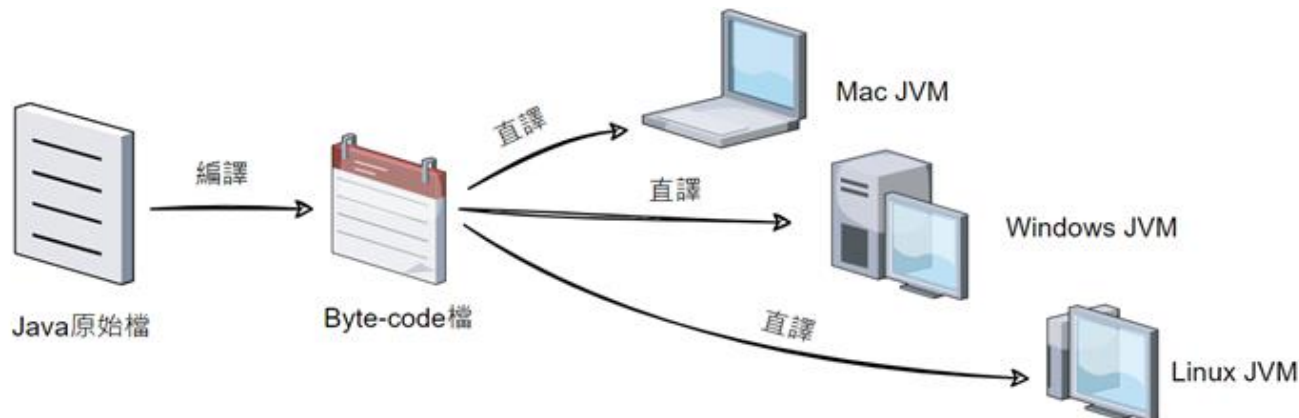
編譯與直譯

- 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、C++與FORTRAN等
 - 直譯式的語言如 Python、JavaScript、VBScript、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

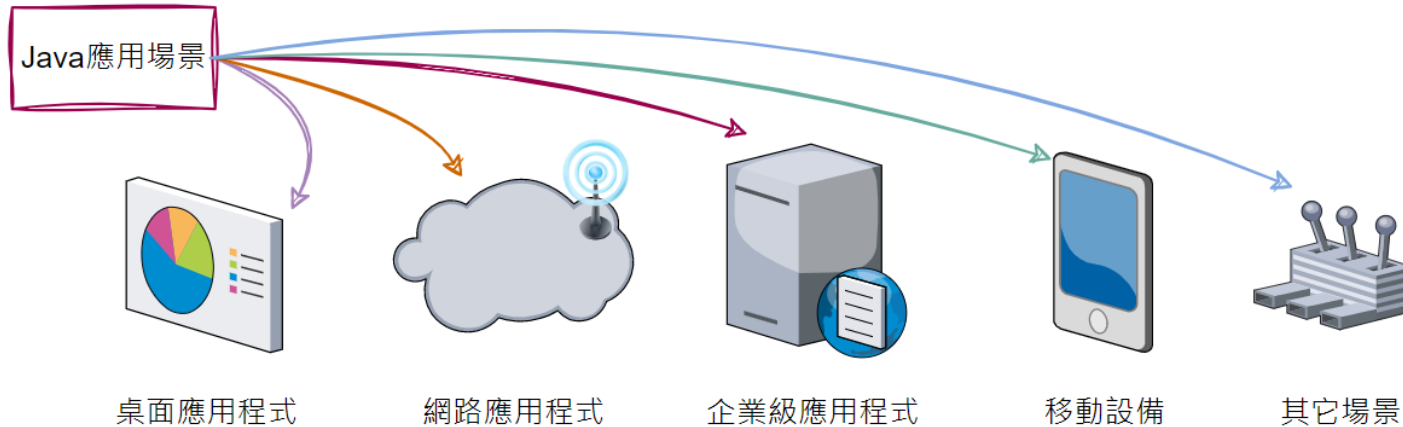
- Java的編譯與執行的程序：
 - 將程式碼編譯成與平台無關（ platform-independent ）的機器碼，稱之為「位元碼組」（ byte-codes ）
 - 經編譯後，可在裝有JVM上的平台直接執行
- 下圖說明Java相關的執行流程：





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、PDA，甚至連小朋友玩的樂高（Lego）也是使用Java來延伸





VSCode的安裝(1/3)

- VSCode提供的安裝包，請到

<https://aka.ms/vscode-java-installer-win>

取得安裝包

下載後，點選下載的檔案即可進行安裝。



VSCode的安裝(2/3)

Install Coding Pack for Java

This Setup Wizard will automatically install the JDK distribution on your system.

The JDK binary to be installed is [licensed](#). By downloading and installing it, you agree to the [license terms](#). The Microsoft Visual Studio Code is licensed under these [license terms](#).

By continuing, you agree to the terms of the license.

☒ I accept the terms in the license.

Cancel

Configure Components

Required components are listed below. Click Install to proceed.

Java Development Kit:	N/A	Not Installed
Visual Studio Code:	N/A	Not Installed

Actions:

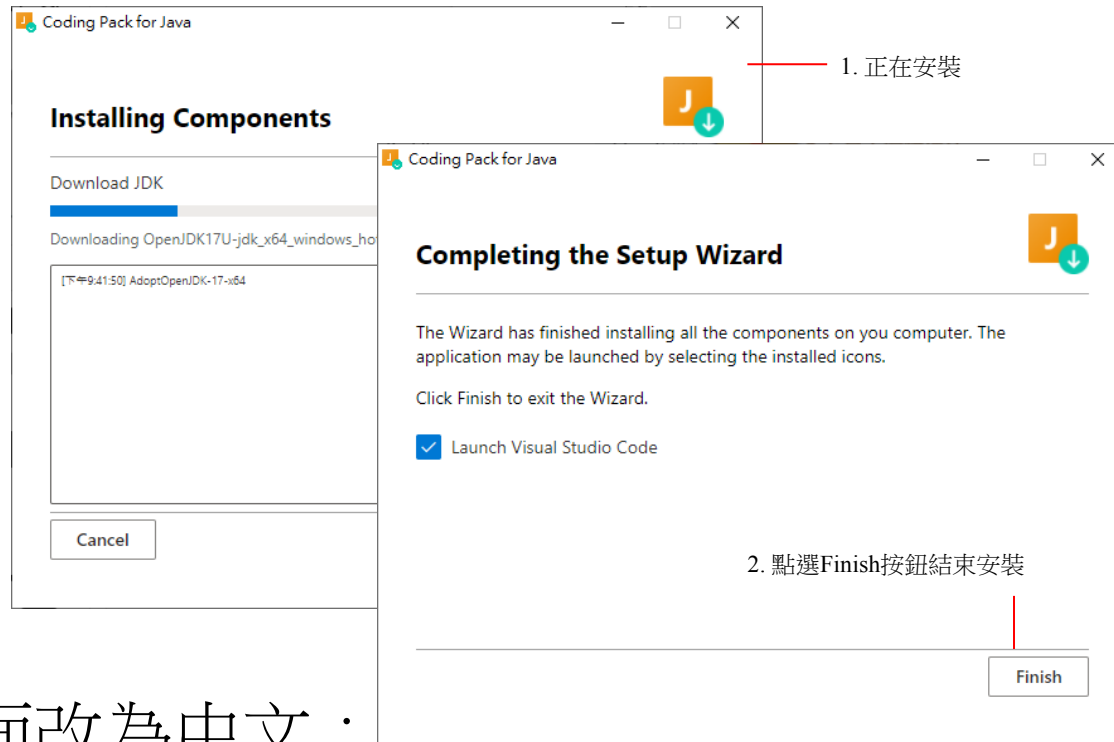
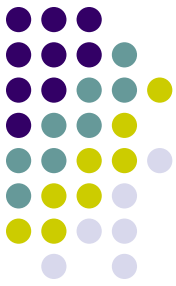
- Download JDK
- Install JDK
- Download Visual Studio Code
- Install Visual Studio Code
- Install Java Extensions

Cancel Back Install

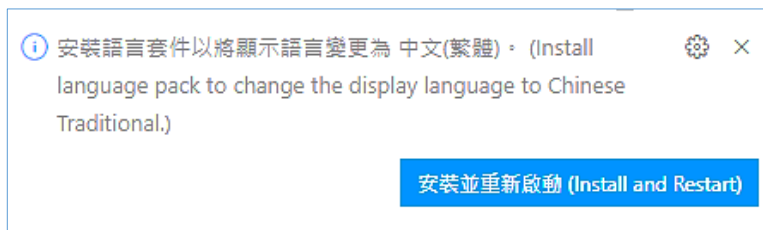
1. 點選視窗右下方Next按鈕

2. 點選Install按鈕

VSCode的安裝(3/3)



- 將界面改為中文：



點選此項，將界面改為中文版



VSCode的視窗

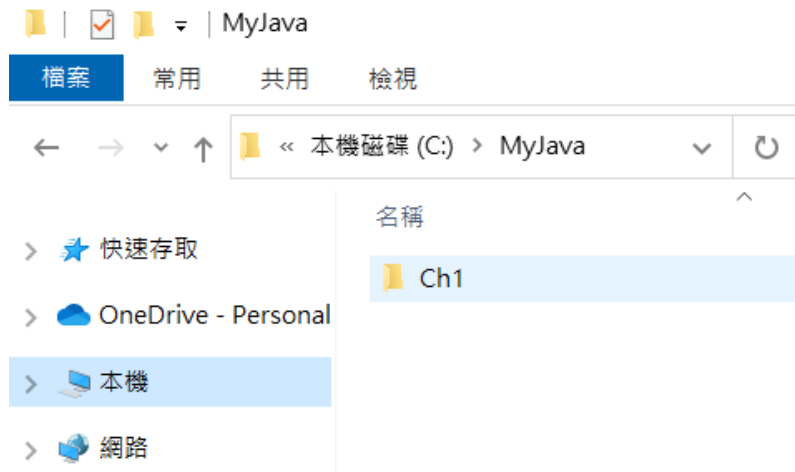
- VSCode視窗的解說：





在VSCode新建程式 (1/3)

- 在C碟的根目錄裡建立MyJava資料夾，然後在此資料夾內建子資料夾Ch1，用來存放第一章範例：

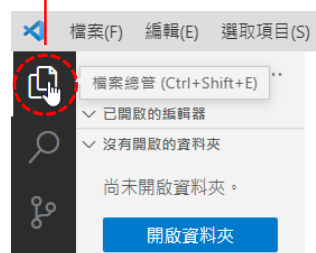




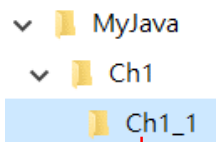
在VSCode新建程式 (2/3)

- 建一個資料夾存放Ch1_1.java :

1. 點選檔案總管



2. 點選此處



3. 選取 Ch1_1 資料夾

4. 建議勾選此項，以避免這個對話方塊

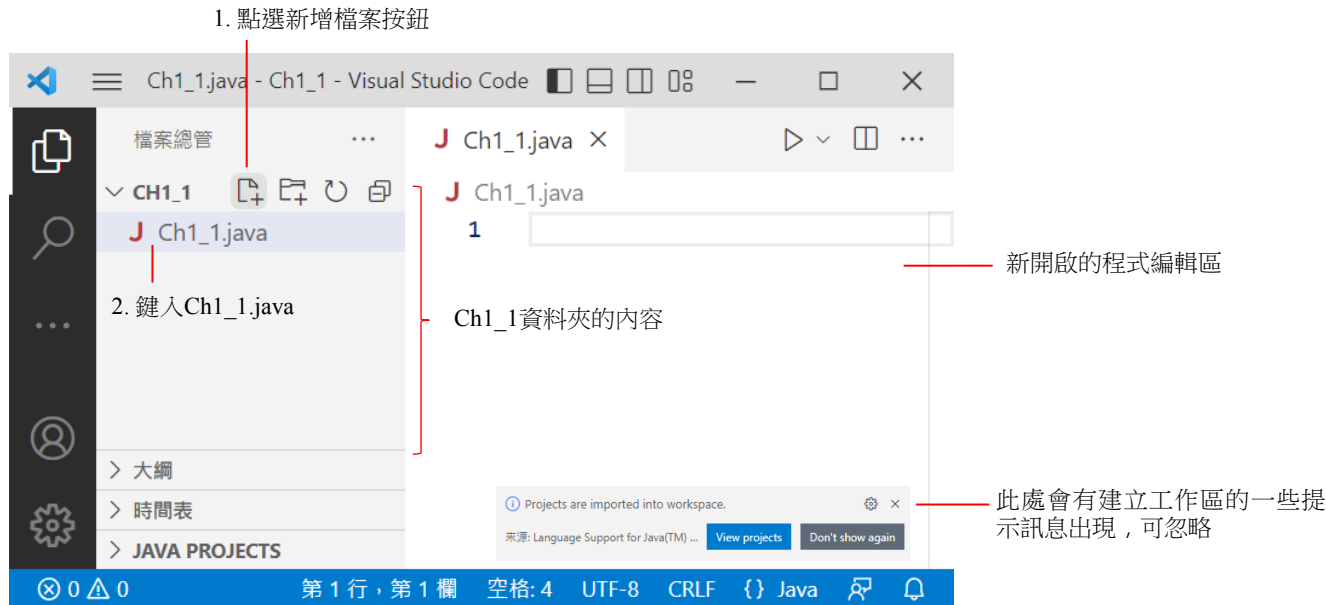


5. 點選這個按鈕



在VSCode新建程式 (3/3)

- 新增檔案Ch1_1.java :





輸入程式碼 (1/3)

- 把Ch1_1鍵入VSCode中：

```
01 // Ch1_1, 第一個 Java 程式
02 public class Ch1_1{
03     public static void main(String[] args){
04         System.out.println("Hello Java!");
05     }
06 }
```

- 當第二行輸入pu時，選單內有三個選項供選擇：





輸入程式碼 (2/3)

- 選擇第3個選項public_static_method：

```
Ch1_1 > J Ch1_1.java > Ch1_1
1 // Ch1_1, 第一個Java程式
2 public class Ch1_1{
3     pub
```

選擇這個選項

- 按下Tab鍵將輸入點移到name上，輸入main即可直接覆蓋掉name：

```
Ch1_1 > J Ch1_1.java > Ch1_1 > name()
1 // Ch1_1, 第一個Java程式
2 public class Ch1_1{
3     public static void name() {
4     }
5 }
6 }
```

1. 按下Tab鍵即會將輸入點移到name上，再改為main

2. 在括號內輸入String[] args



輸入程式碼 (3/3)

- 程式碼輸入後，畫面應如下所示：

Ch1_1 > J Ch1_1.java > Ch1_1 > main(String[])

```
1  // Ch1_1, 第一個Java程式
2  public class Ch1_1{
3      Run | Debug
4      public static void main(String[] args) {
5          System.out.println(x: "Hello Java!");
6      }
```

VSCode自動新增加的一行，它會出現在main() 函數的上面。注意它不是程式碼的一部分，所以這一行也沒有行號

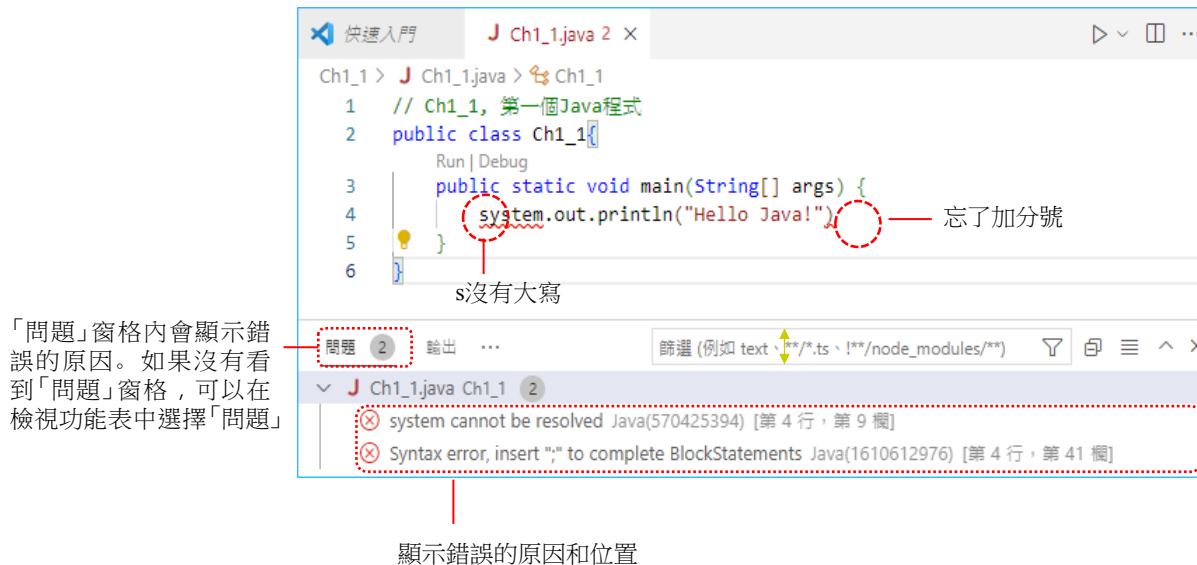
VSCode會自動加上垂直的對齊線，方便觀察程式碼是否有對齊

這個 x: 符號是VSCode自動帶出，用來標識這個位置的參數名稱。它也不是程式碼的一部分，可以先忽略它



程式碼輸入有誤

- 輸入的語法錯誤時，會顯示在「問題」窗格：

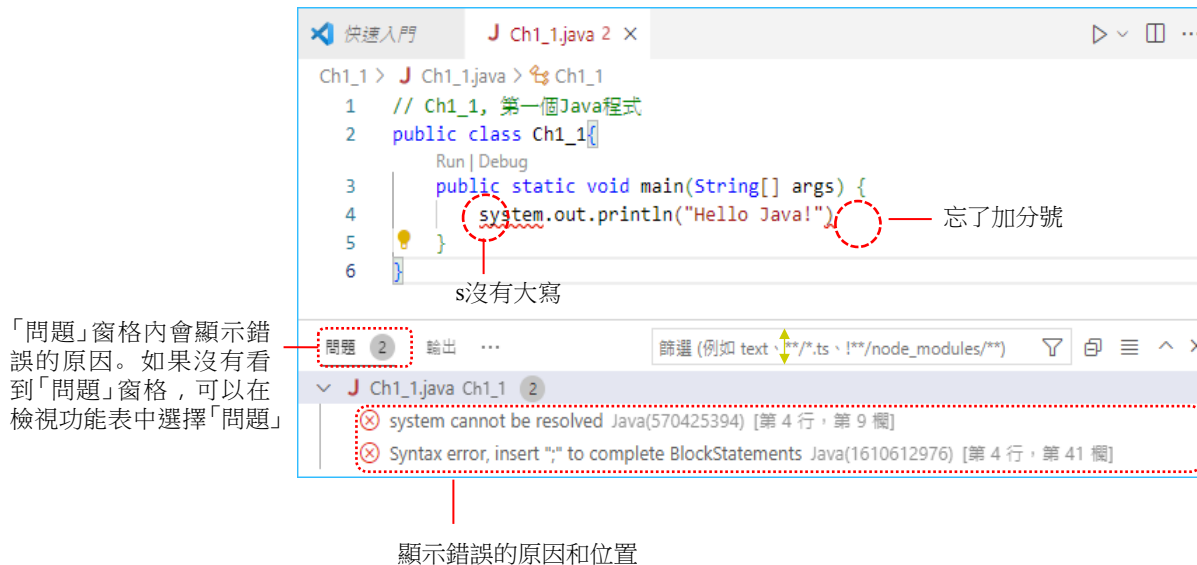


- 第一個錯誤訊息是 "system cannot be resolved"
- 第二個錯誤訊息是 "Syntax error, insert ';' to complete BlockStatements"



程式碼輸入有誤

- 輸入的語法錯誤時，會顯示在「問題」窗格：

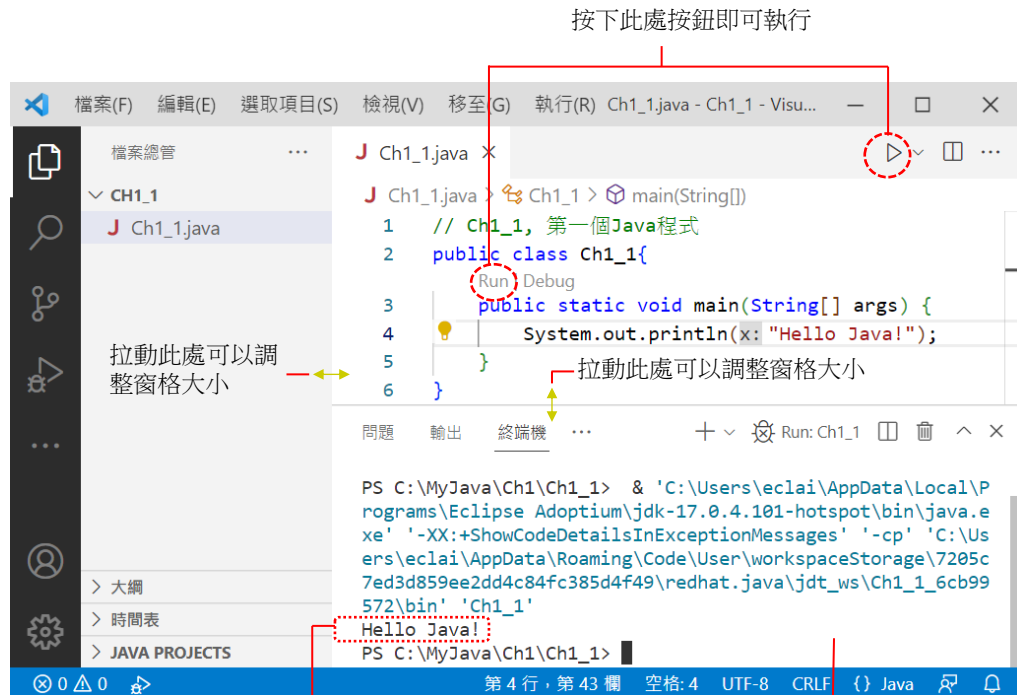


- 第一個錯誤訊息是 "system cannot be resolved"
- 第二個錯誤訊息是 "Syntax error, insert \";" to complete BlockStatements"



執行程式的方式一

- 按下VSCode自動生成那行裡的Run
- 編輯視窗上方的Run Java按鈕  編譯並執行程式：



程式執行的結果

執行過程會顯示在「終端機」窗格內



執行程式的方式二

- 在「終端機」窗格內將目錄切換到程式所在的資料夾Ch1_1，再鍵入

`java Ch1_1.java`

執行結果

```
問題 輸出 終端機 ... + - Run: Ch1_1 [ ] [ ] ^ X
PS C:\MyJava\CH1> cd Ch1_1
PS C:\MyJava\CH1\Ch1_1> java Ch1_1.java
Hello Java!
PS C:\MyJava\CH1\Ch1_1> 
```

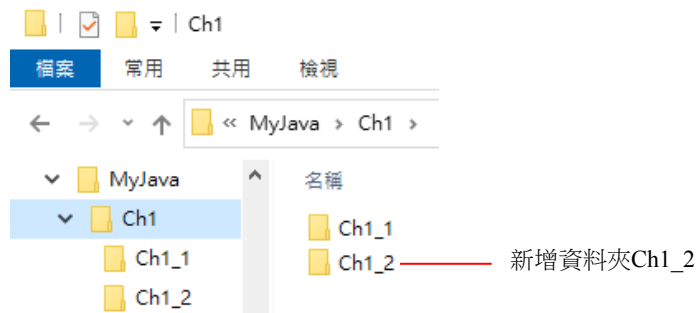
1. 切換目錄到Ch1_1

2. 鍵入java Ch1_1.java

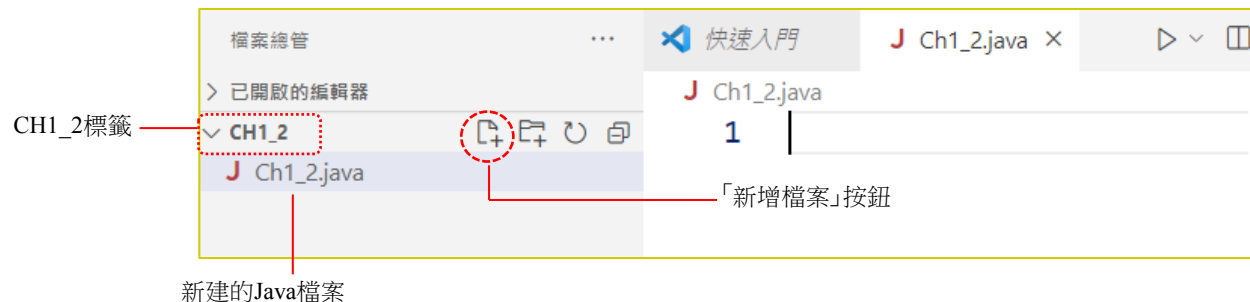


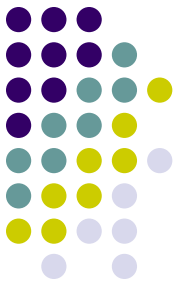
第二個程式(1/2)

- 在C:\MyJava\Ch1資料裡內建立子資料夾Ch1_2：



- 在VSCode中開啟Ch1_2資料夾，然後新增檔案Ch1_2.java：

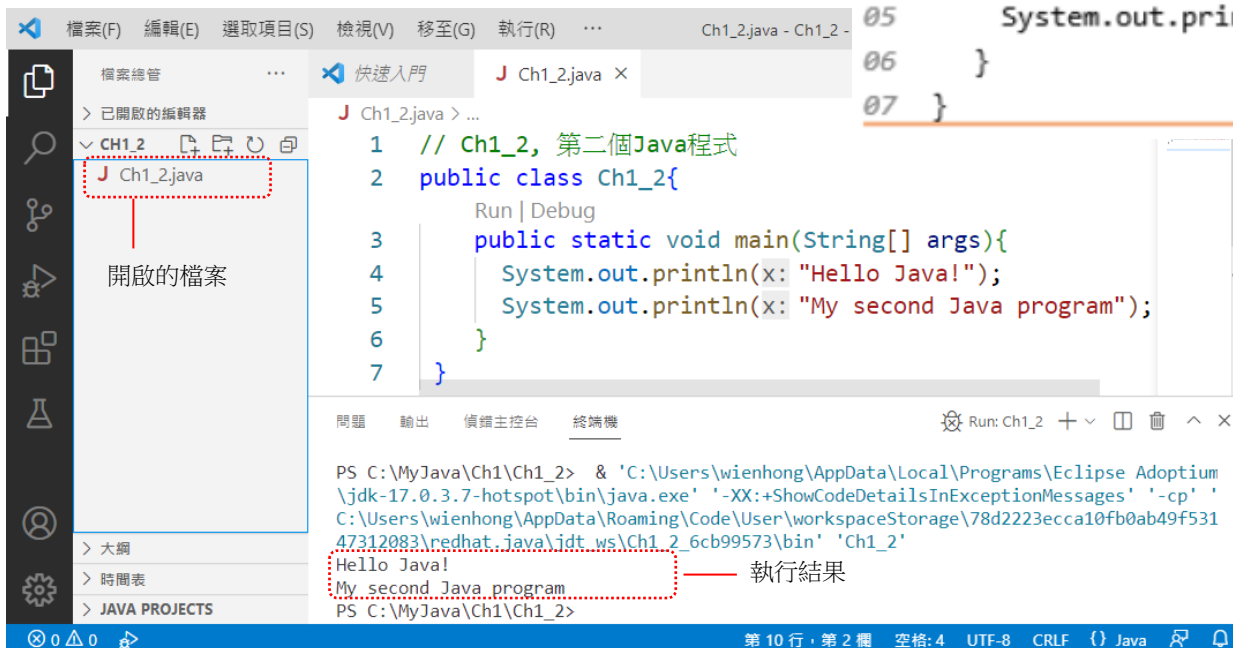




第二個程式(2/2)

- 鍵入Ch1_2.java程式內容：

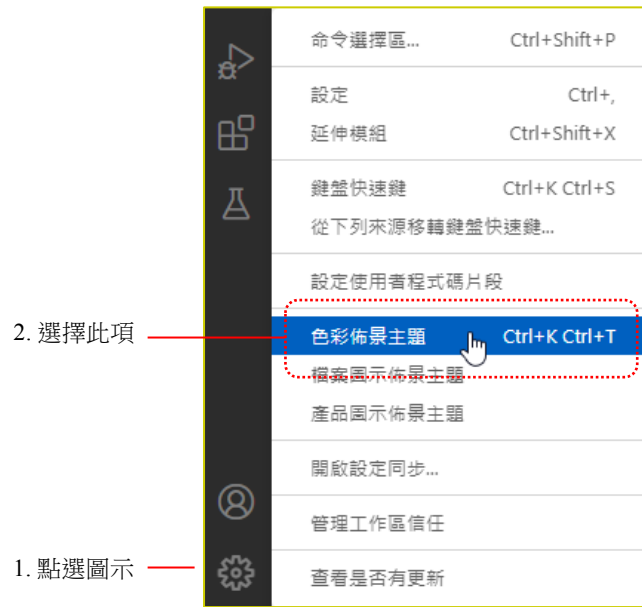
```
01 // Ch1_2, 第二個 Java 程式
02 public class Ch1_2{
03     public static void main(String[] args){
04         System.out.println("Hello Java!");
05         System.out.println("My second Java program");
06     }
07 }
```





佈景主題

- 點選左側工具列最下方的「管理」圖示，選擇「色彩佈景主題」：





改變字體大小

- 按下 `Ctrl` + `+` 將界面的字體放大一級
- 按下 `Ctrl` + `=` 將界面的字體縮小一級
- 設定編輯區字體的大小：

1. 點選圖示

2. 選擇「設定」

3. 選擇此項

4. 選擇此項



如果看不到這個窗格，把視窗拉寬一點就會出現

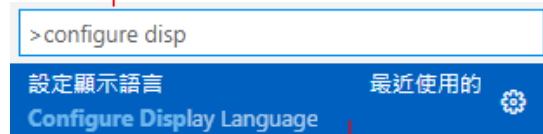


改變界面顯示的語言

- 將VSCode的界面切換成英文或中文：

1. 按下 **Ctrl** + **Shift** + **P**

2. 鍵入Configure Disp



3. 選擇「設定顯示語言」

4. 更改要顯示的語言





顯示函數參數的提示

- 程式碼有和沒有內嵌提示的比較：

```
System.out.println(x: "Hello Java");    System.out.println("Hello Java");
```

有內嵌提示

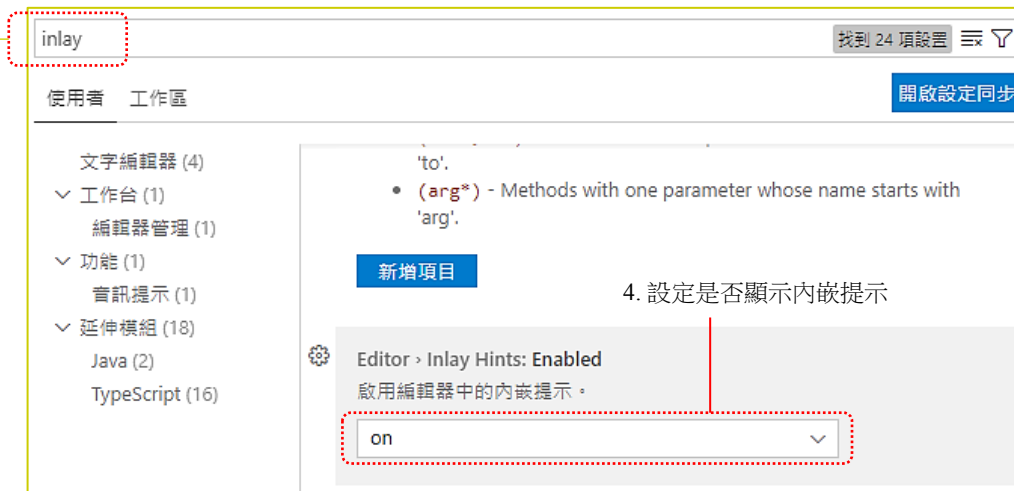
沒有內嵌提示

- 「管理」-「設定」-「搜尋設定」中鍵入inlay，於「啟用編輯中的內嵌提示」中切換內嵌提示功能：

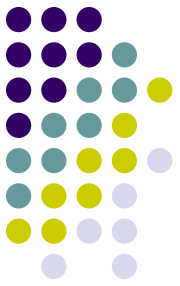
1. 點選圖示

2. 選擇「設定」

3. 鍵入inlay



4. 設定是否顯示內嵌提示



-The End-