IntelliJ IDEA

TYIC 桃高資訊社

IDE

功能多
不易上手

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

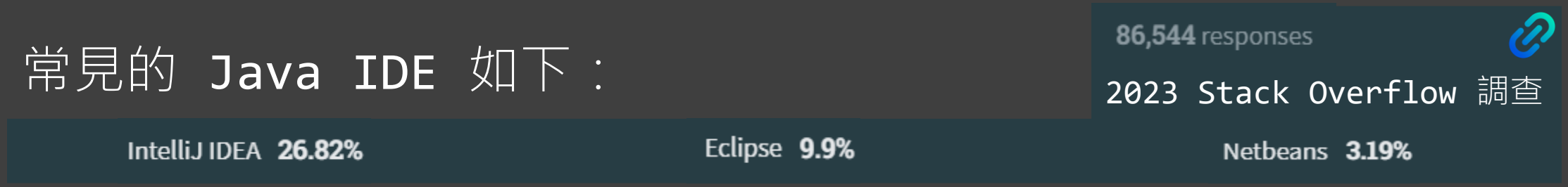
通常是針對特定的程式語言設計並且整合了許多東西，包含：文字編輯器、除錯器(debugger)、自動組建工具(build automation)，部分還有版本控制系統(Version Control System，簡稱 VCS)

如：PyCharm、Visual Studio、Code::Blocks、Dev-C++、Eclipse、IntelliJ IDEA

一款好的 IDE 能很大程度加速開發

Java IDE

常見的 Java IDE 如下：



IntelliJ IDEA

IntelliJ IDEA
JETBRAINS IDE

作者：JetBrains 公司
免費版：自由開源軟體
旗艦版：專有軟體
使用 Java 編寫

Eclipse

eclipse

作者：Eclipse 基金會
自由開源軟體
使用 Java 編寫

Netbeans

Apache NetBeans IDE

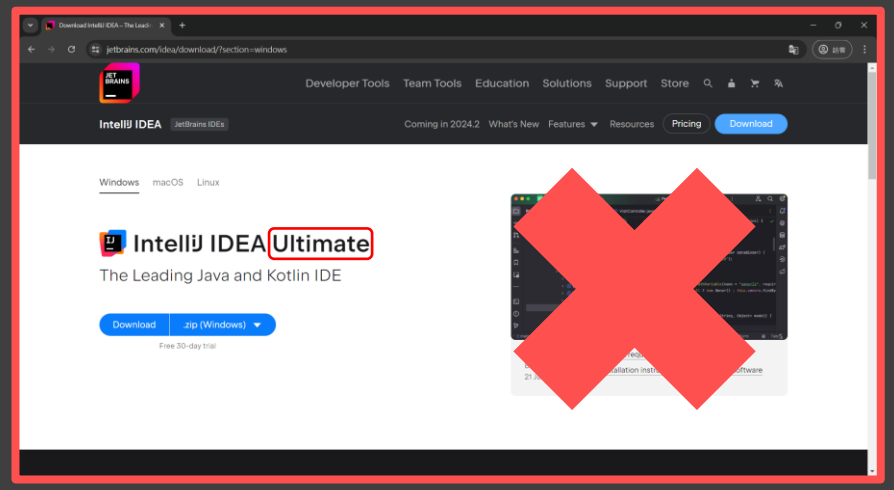
作者：Apache 基金會
自由開源軟體
使用 Java 編寫

IntelliJ IDEA 是這學年會用的

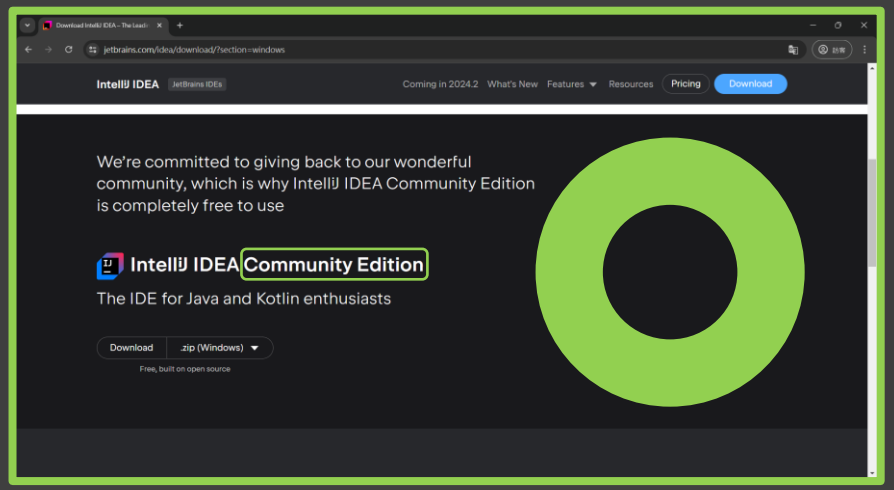
還有許多 Java IDE，但極為少見

下載/安裝

https://www.jetbrains.com/idea/download/



往下捲動



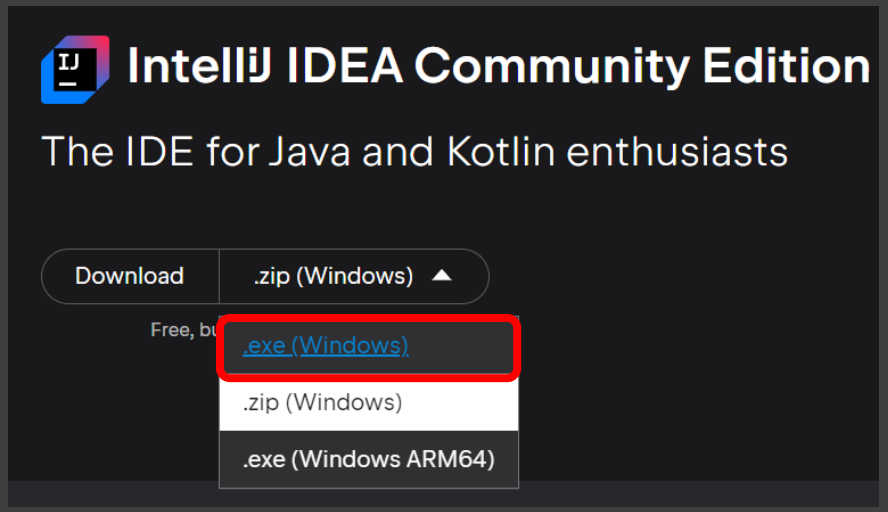
付費版(旗艦版)

免費版(社區版)

選擇 .exe 後就會開始下載安裝程式

安裝過程非常簡單

按照安裝程式的說明即可



新增專案

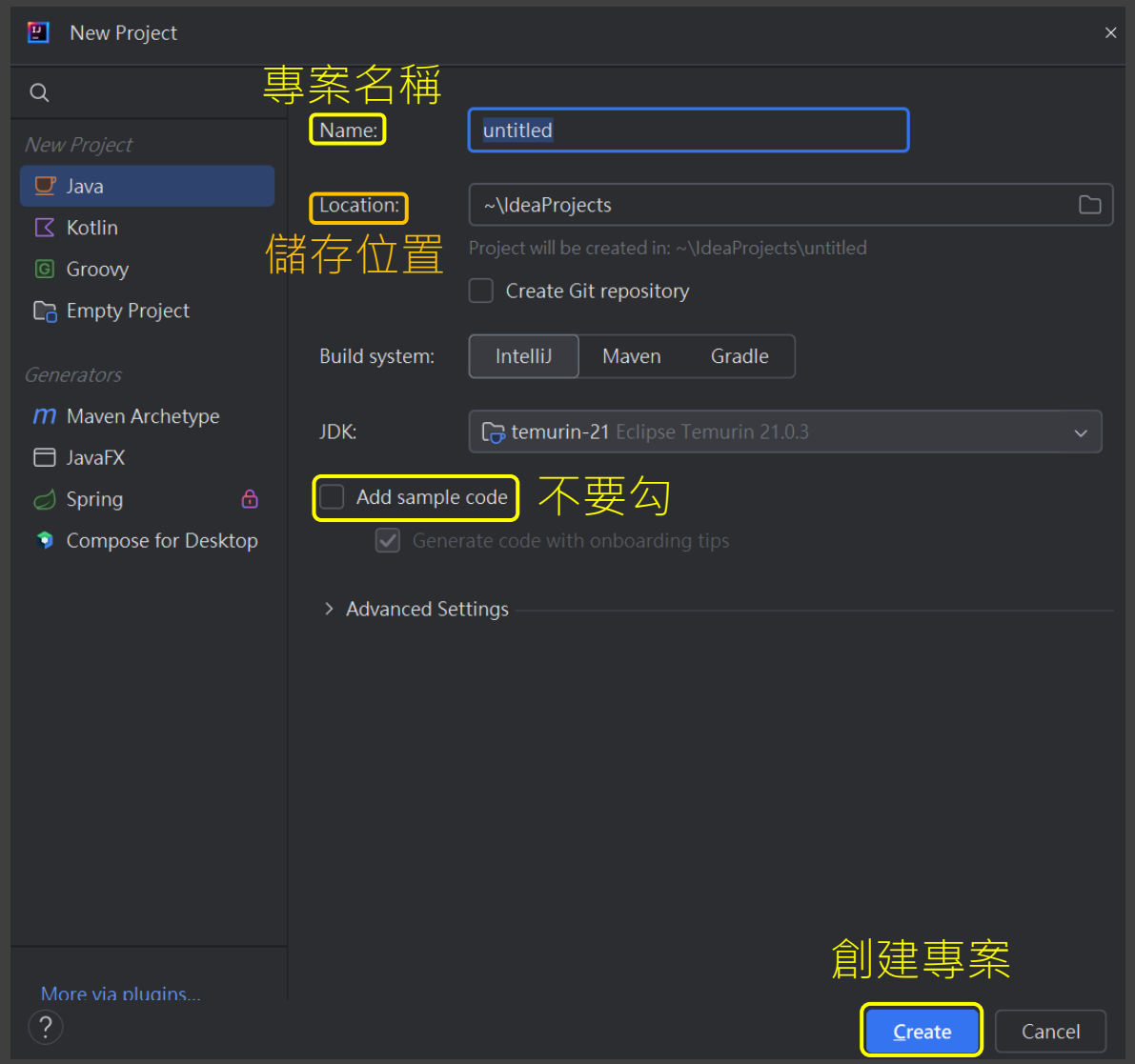
開啟後的介面


Welcome to IntelliJ IDEA
IntelliJ IDEA
2024.1.4
Projects
Customize
Plugins
Learn
Welcome to IntelliJ IDEA
Create a new project to start from scratch.
Open existing project from disk or version control.
新增專案
New Project
Open
Get from VCS
從版本控制系統獲取
開啟專案資料夾

新增專案

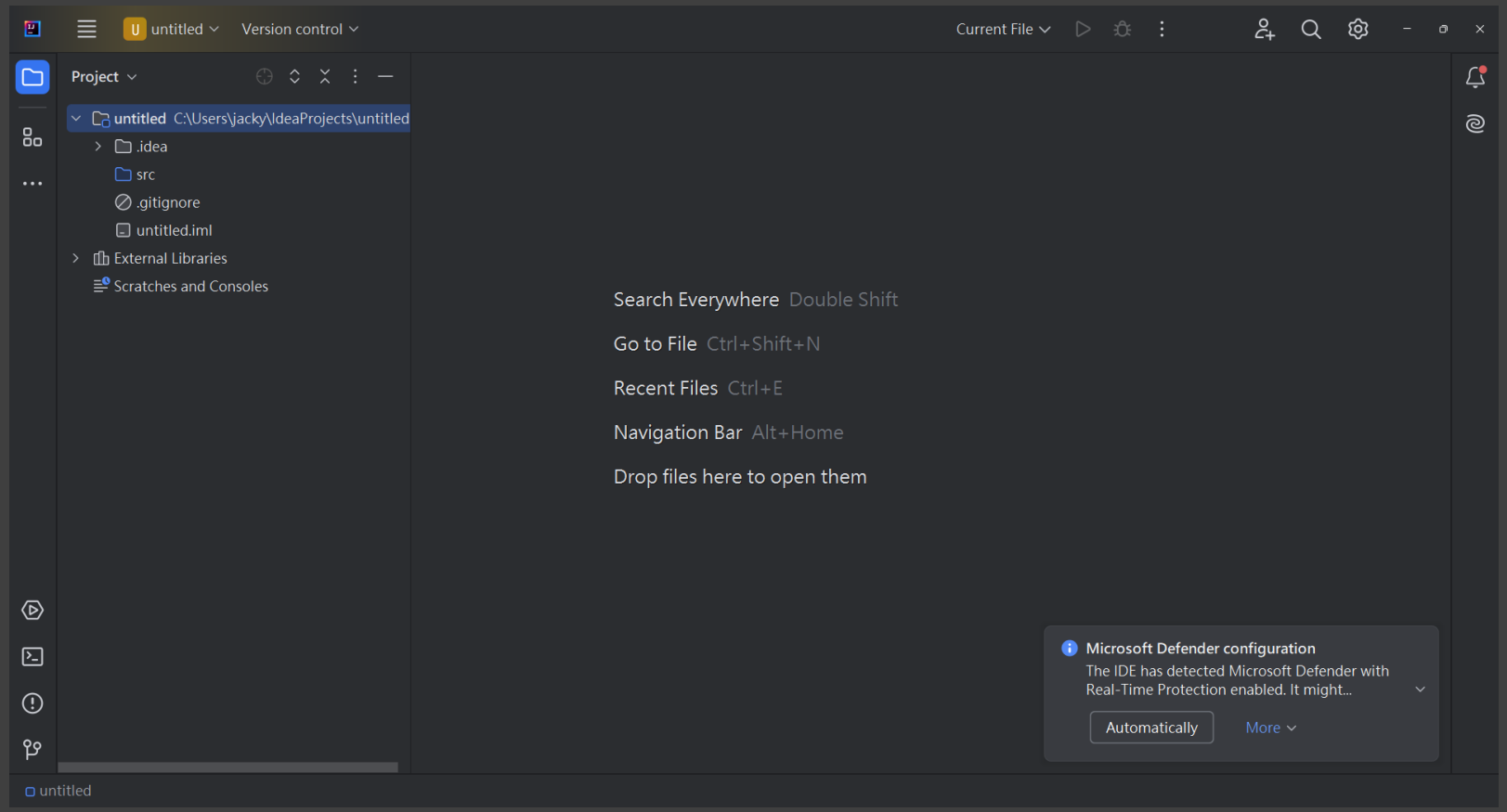
點擊 "New Project" 後
會出現右圖視窗
填好名稱後就可以創建專案

注意：不要勾選
"Add Sample Code"



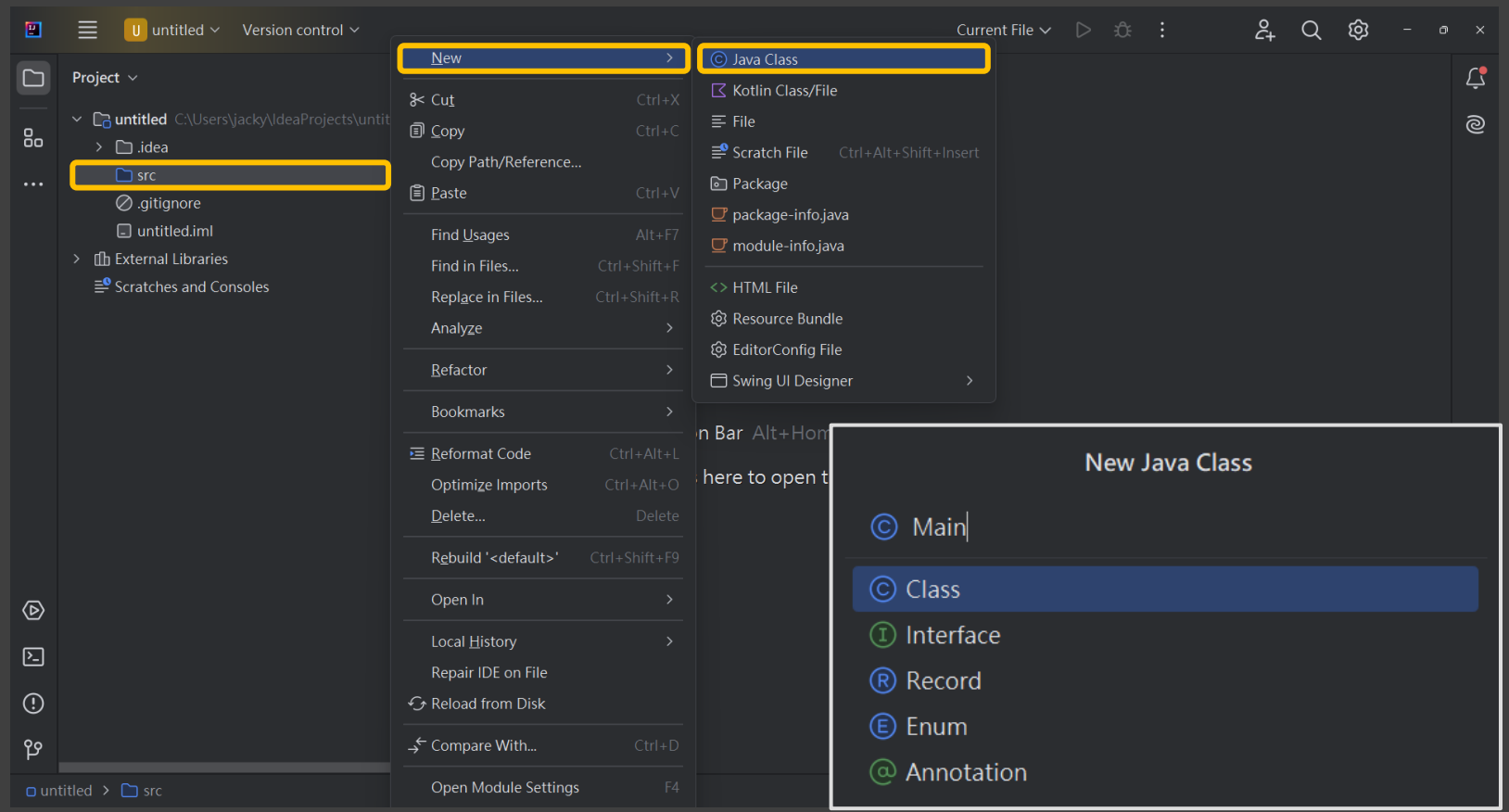
新增專案

創建完後
就會顯示專案



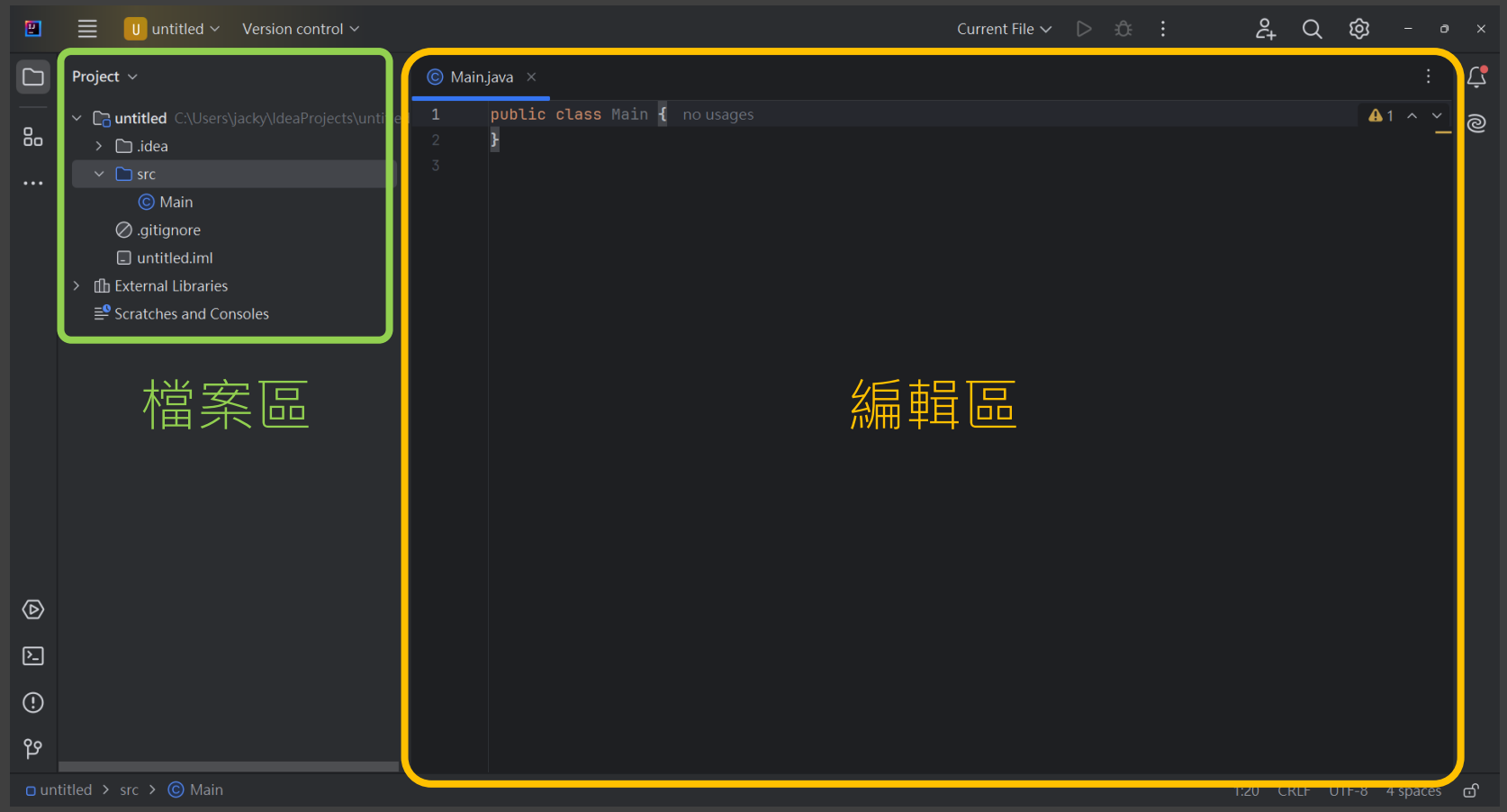
新增檔案

要新增 class
只需要在資料夾上
右鍵 -> New
-> Java Class
填入類別名稱
按下 Enter
即可創建
目前 class
皆放在 "src" 下



新增檔案

創建完後
會自動開啟檔案
即可在編輯區編輯
所有專案檔案
都會出現在檔案區



編輯

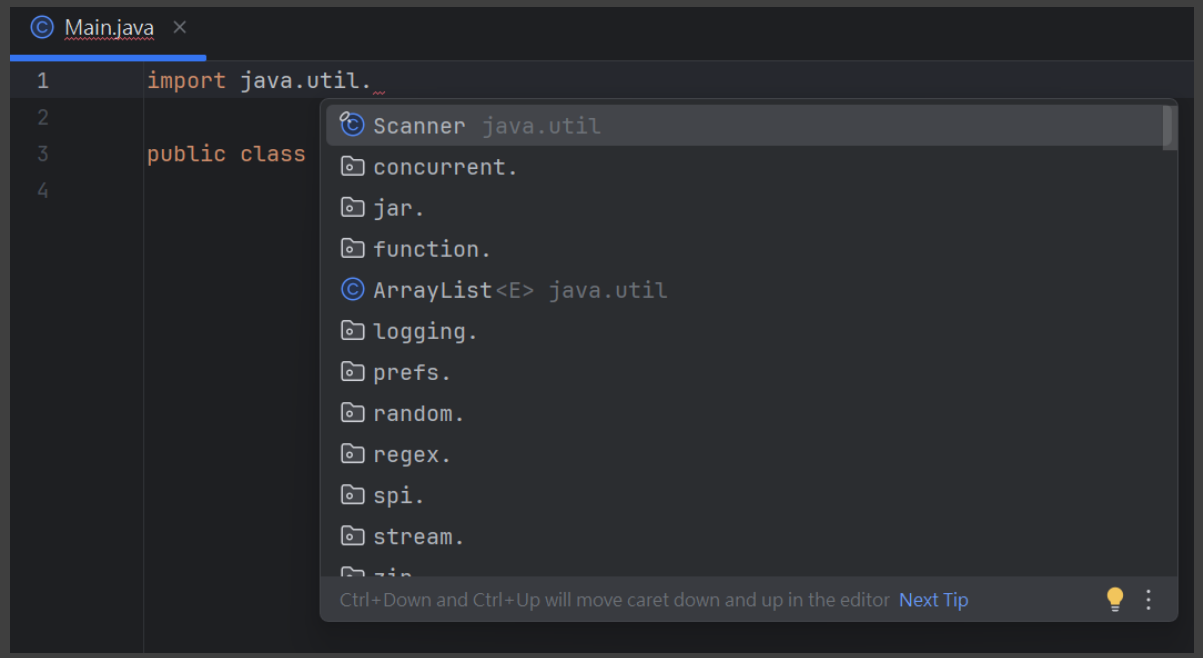
嘗試將上一個程式打上去
並觀察打字過程發生了什麼

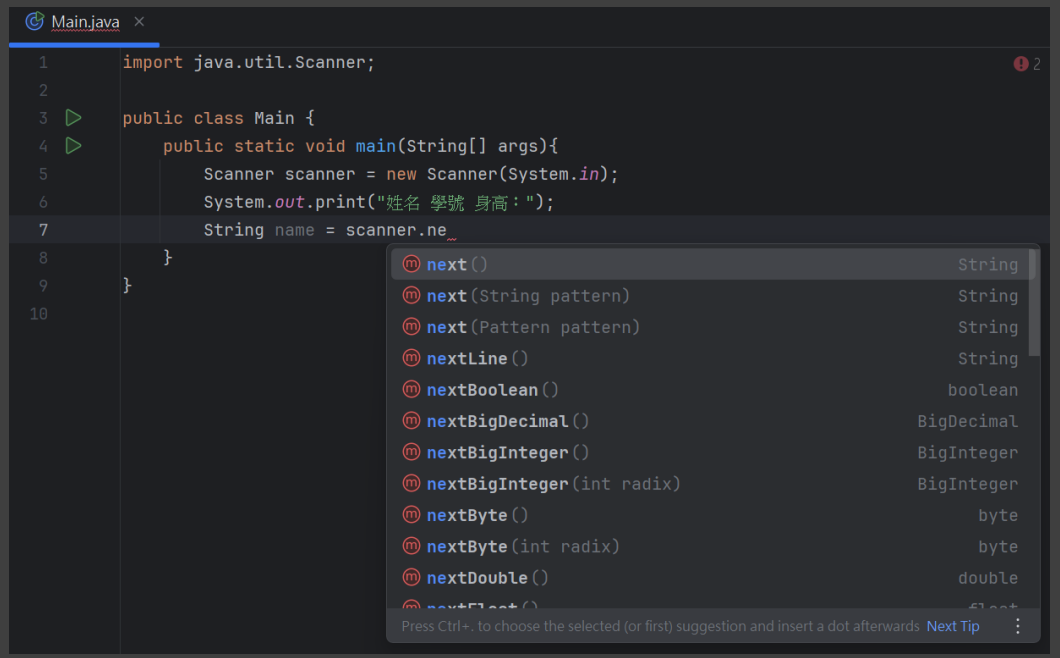
```java
import java.util.Scanner; // 載入套件

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in); // 創建新的 Scanner 實例
        System.out.print("姓名 學號 身高：");
        String name = scanner.next(); // 讀入下一個字串並存入變數 name
        int studentId = scanner.nextInt(); // 讀入下一個 int 並存入變數 studentId
        double height = scanner.nextDouble(); // 讀入下一個 double 並存入變數 height
        System.out.printf("姓名：%s 學號：%d 身高：%.2f%n", name, studentId, height);
    }
}
```

自動補全

打字過程中出現了些許變化：

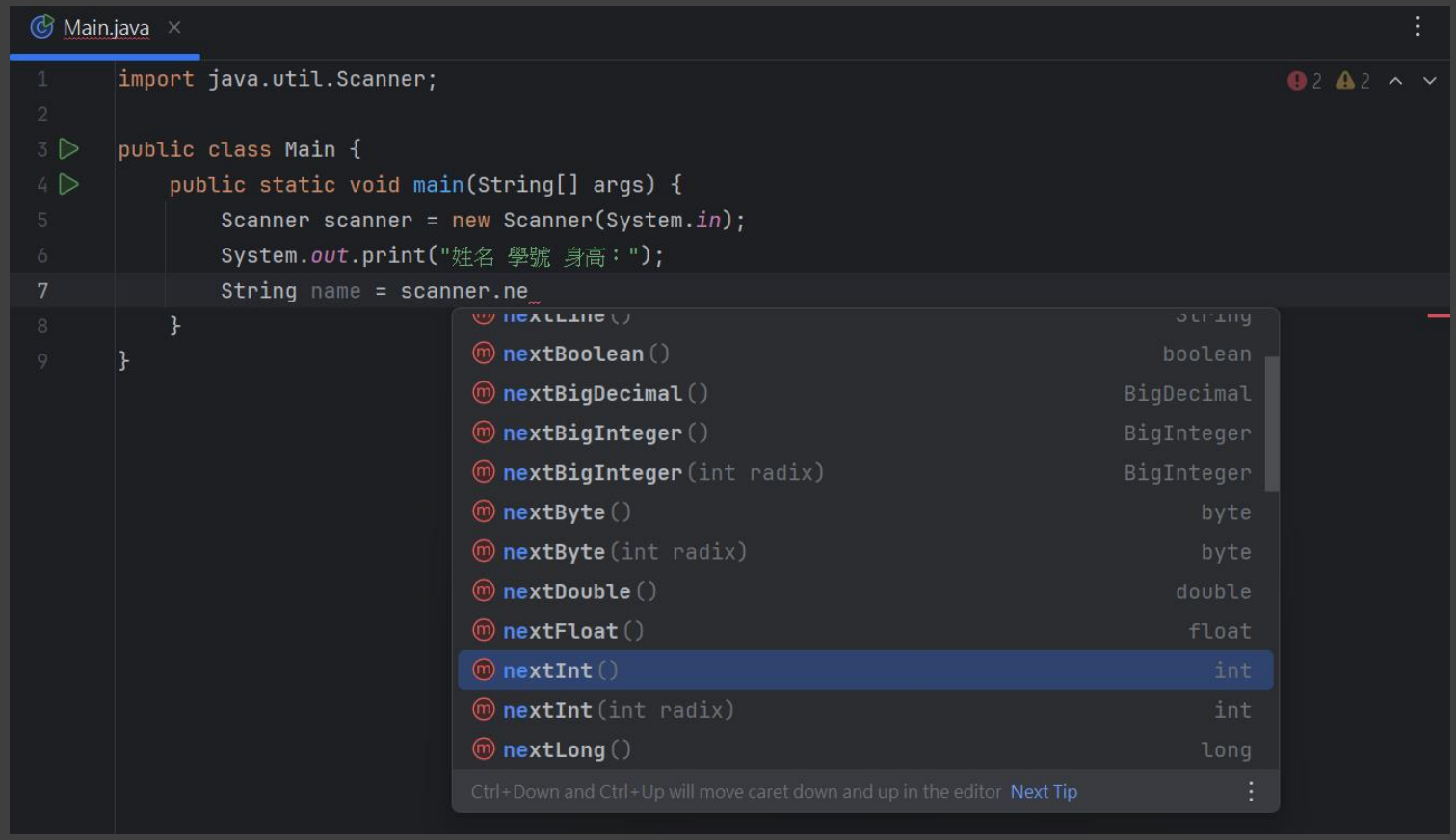
1. 打了左括號，會自動補右括號，雙引號也是
2. 會列出可以接什麼





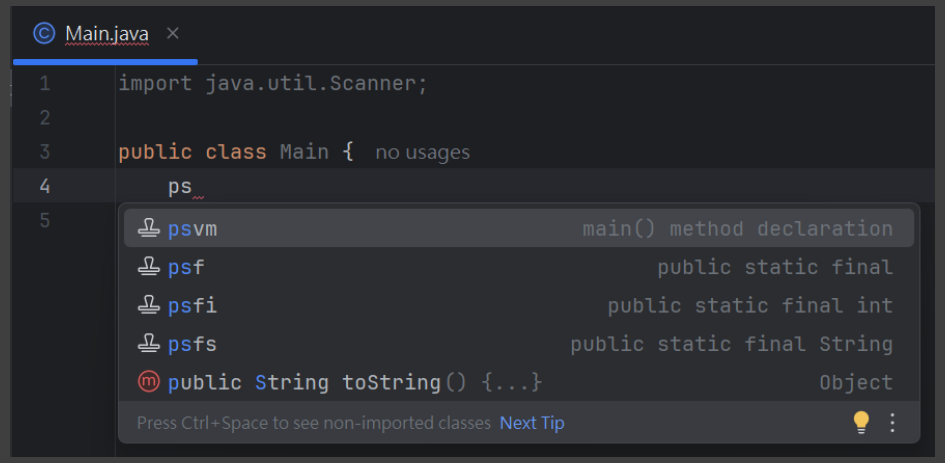
自動補全

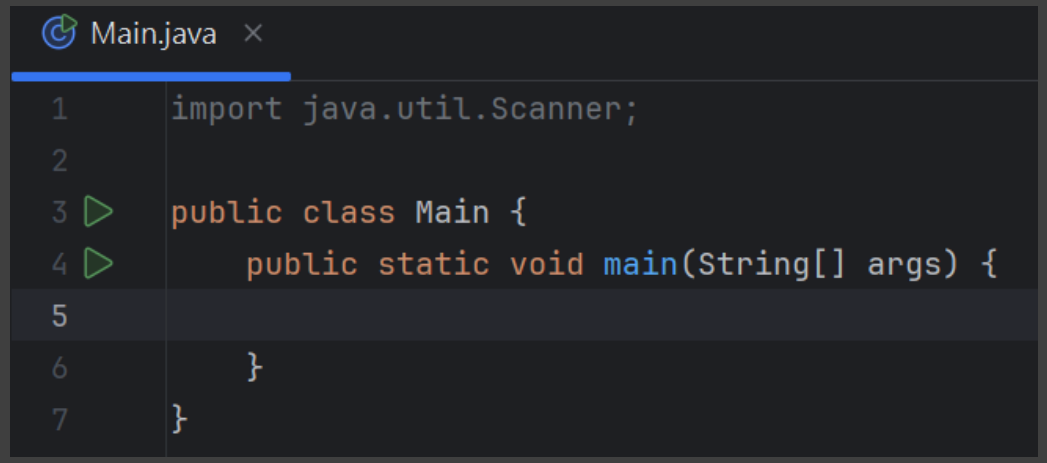
不只列出可以接什麼
還可以通過上下鍵
選擇要輸入的
然後按 Tab 或 Enter
讓 IDE 幫你自動補全

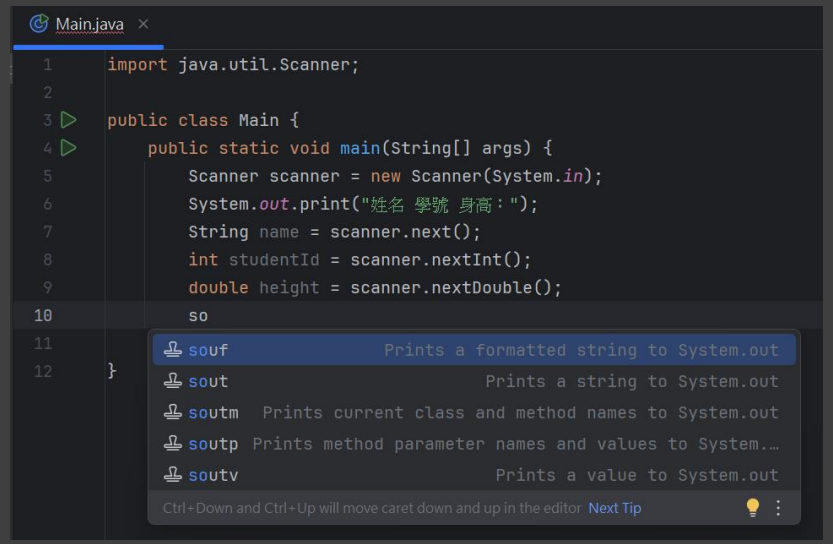


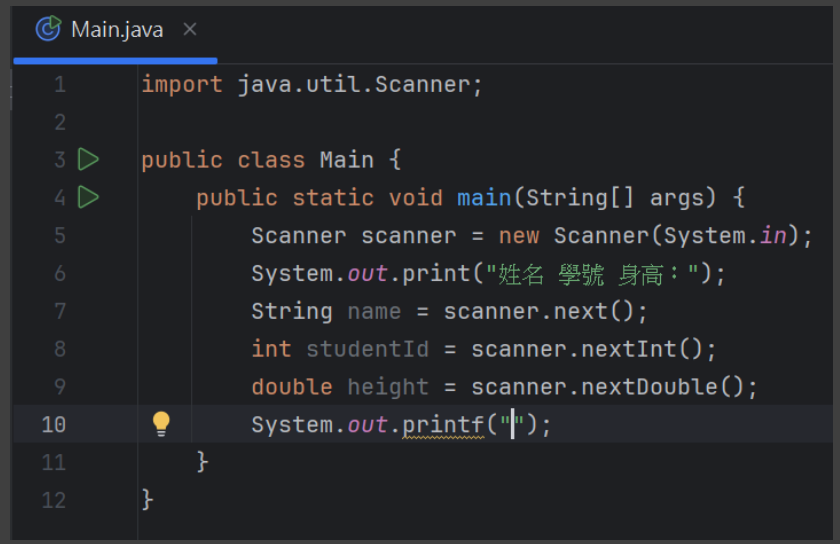
自動補全

甚至，IDEA 還有內建許多快捷縮寫，如 psvm、sout 等



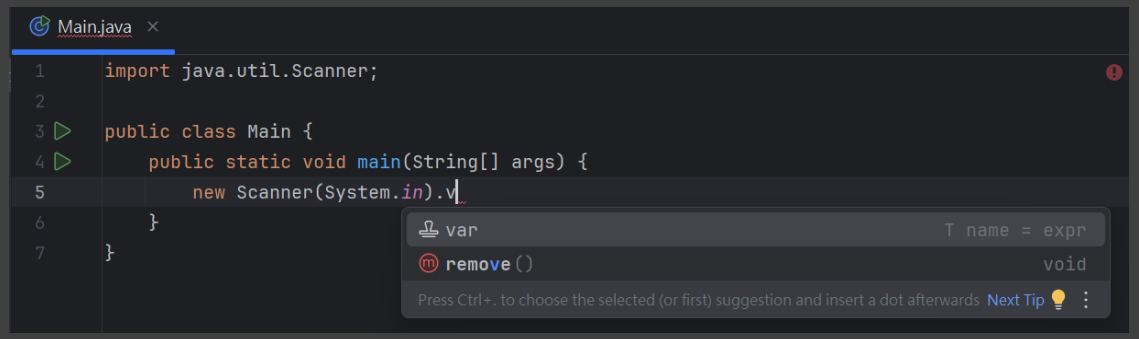


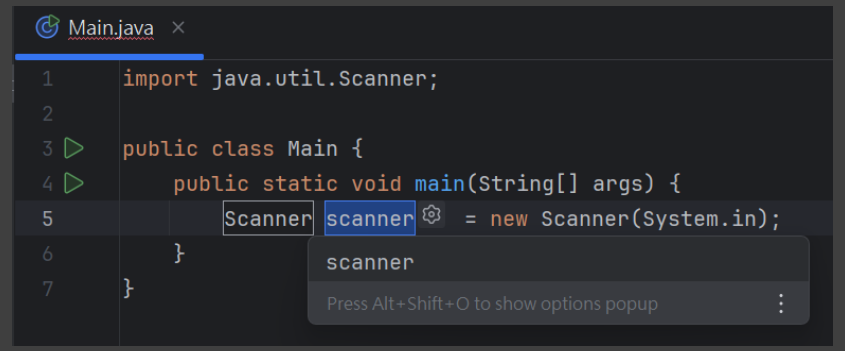


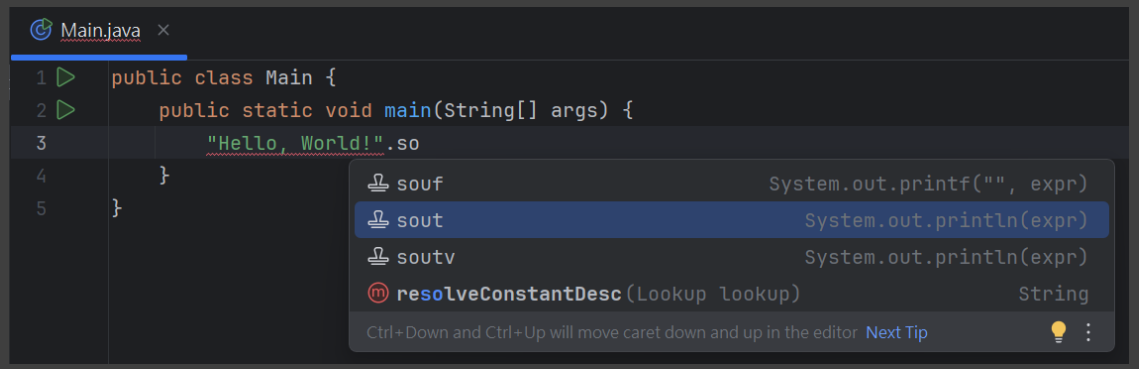


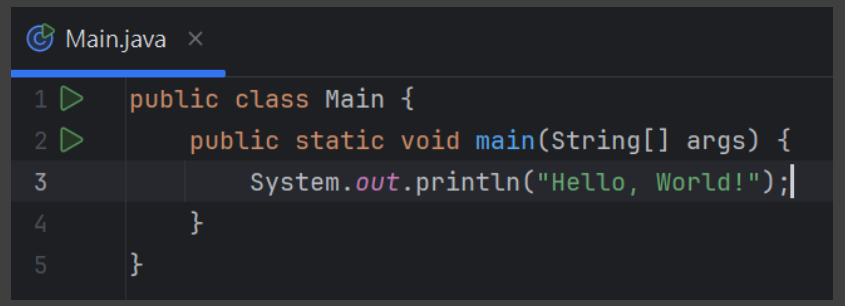
自動補全

還能通過 .var 等縮寫讓 IDE 幫你補更多東西



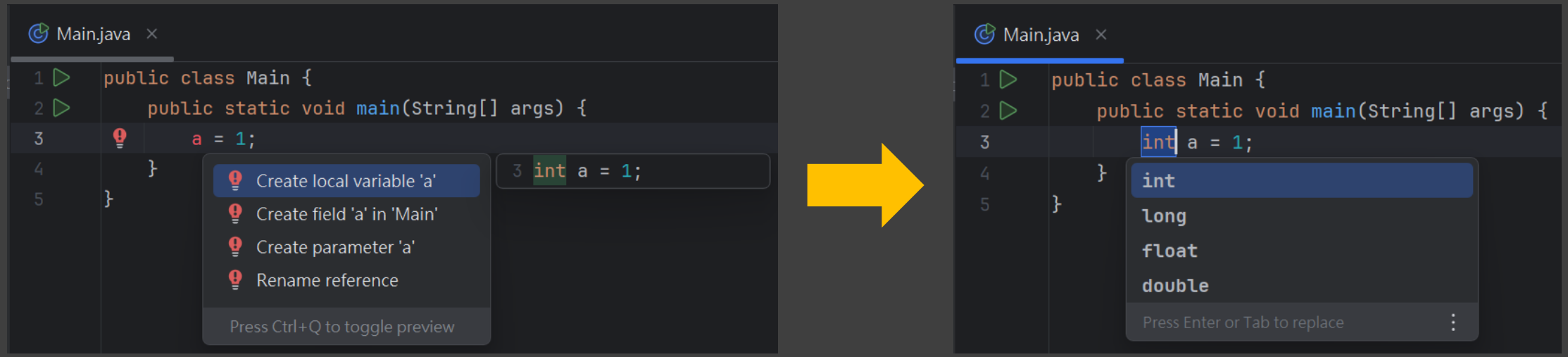






動作

文字游標在某些地方時
還可以按 Alt + Enter 開啟動作選單
可以選擇並執行動作

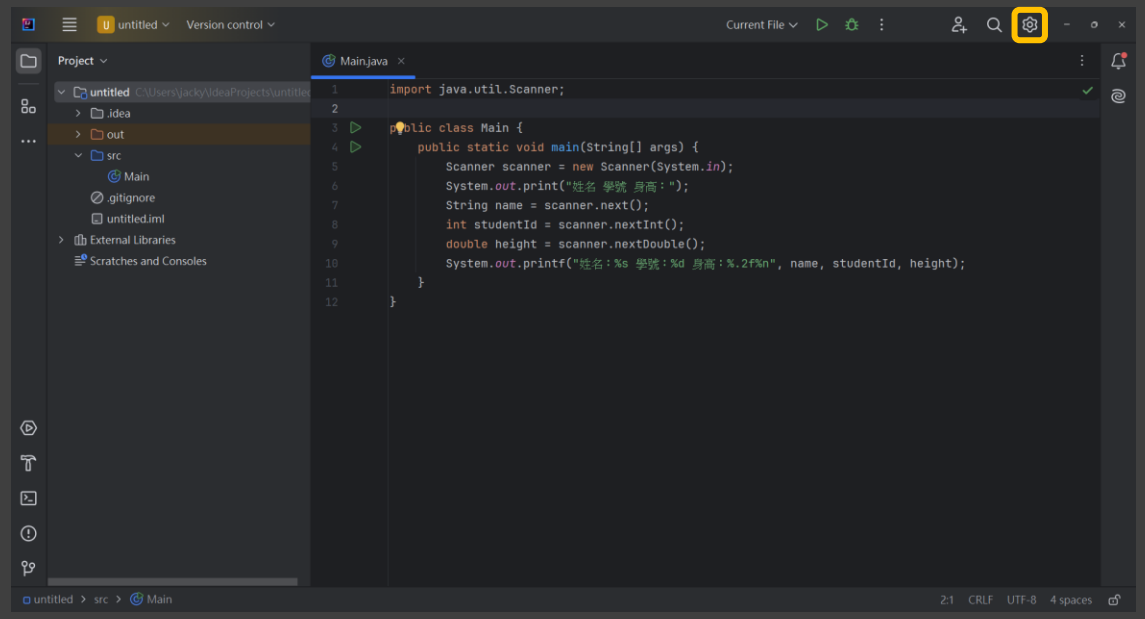


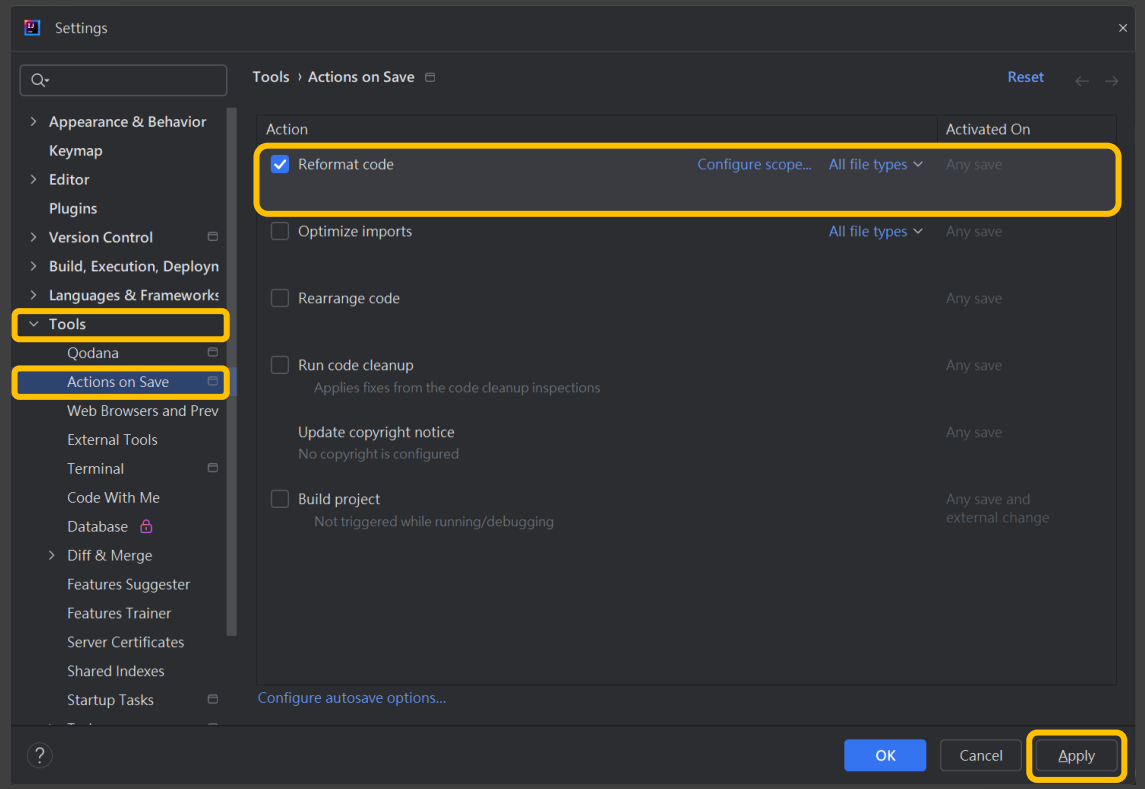
自動格式化

如果想要讓程式碼更容易閱讀
可以開啟自動格式化
讓 IDE 在檔案儲存時幫你格式化

開啟方法：

Setting -> Tools
-> Actions on Save
-> Reformat code
然後按下 Apply 即可





自動格式化

格式化前：

Main.java

```
1  import java.util.Scanner;
2  public class Main{public static void main(String[]args){
3      Scanner scanner=new Scanner(System.in);
4          System.out.print("姓名 學號 身高：");
5          String name=scanner.next();
6  int studentId=scanner.nextInt();double height=scanner.nextDouble();
7  System.out.printf("姓名：%s 學號：%d 身高：%.2f%n",name,studentId,height);
8  }}
```

格式化後：

Main.java

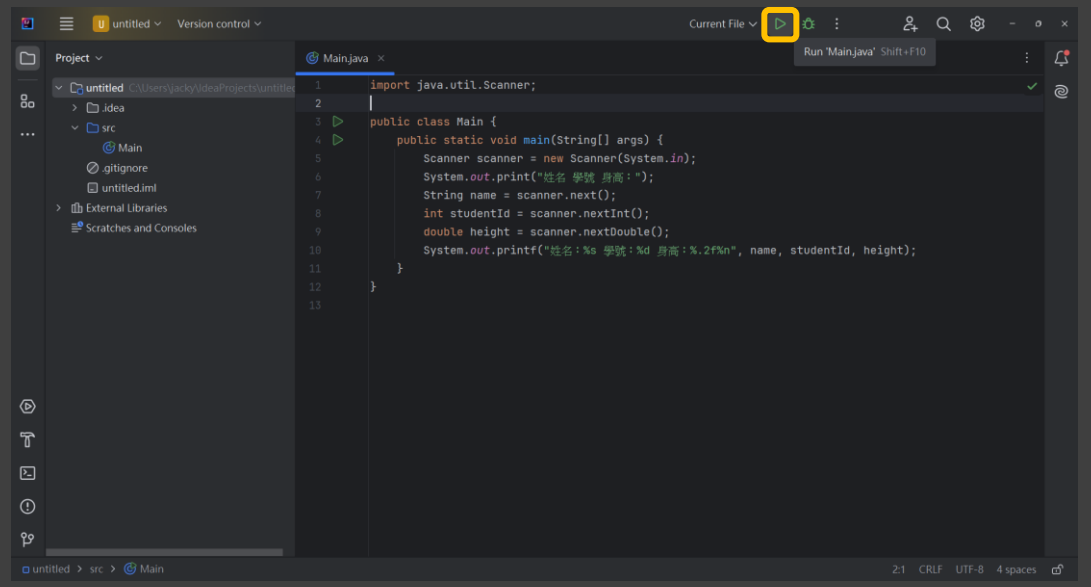
```
1  import java.util.Scanner;
2
3  public class Main {
4      public static void main(String[] args) {
5          Scanner scanner = new Scanner(System.in);
6          System.out.print("姓名 學號 身高：");
7          String name = scanner.next();
8          int studentId = scanner.nextInt();
9          double height = scanner.nextDouble();
10         System.out.printf("姓名：%s 學號：%d 身高：%.2f%n", name, studentId, height);
11     }
12 }
```

快捷鍵

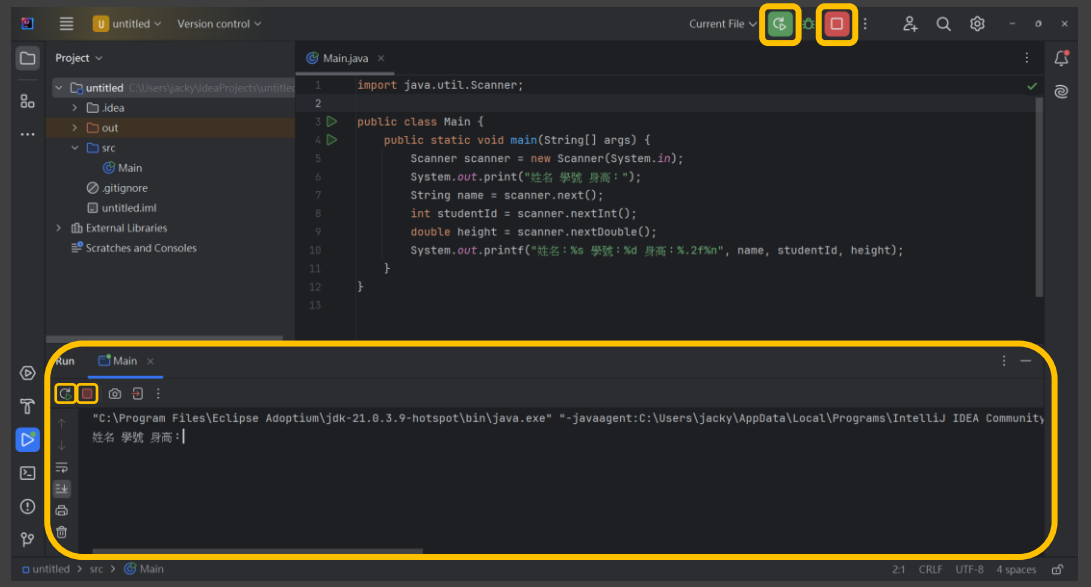
除了支援之前介紹的快捷鍵之外
還有許多好用的快捷鍵
如 `Ctrl + D` 複製貼上該行程式碼
`Ctrl + /` 單行註解
`Ctrl + Shift + /` 多行註解

執行

按下右上角的執行按鈕
即可編譯與執行
無需自行輸入指令

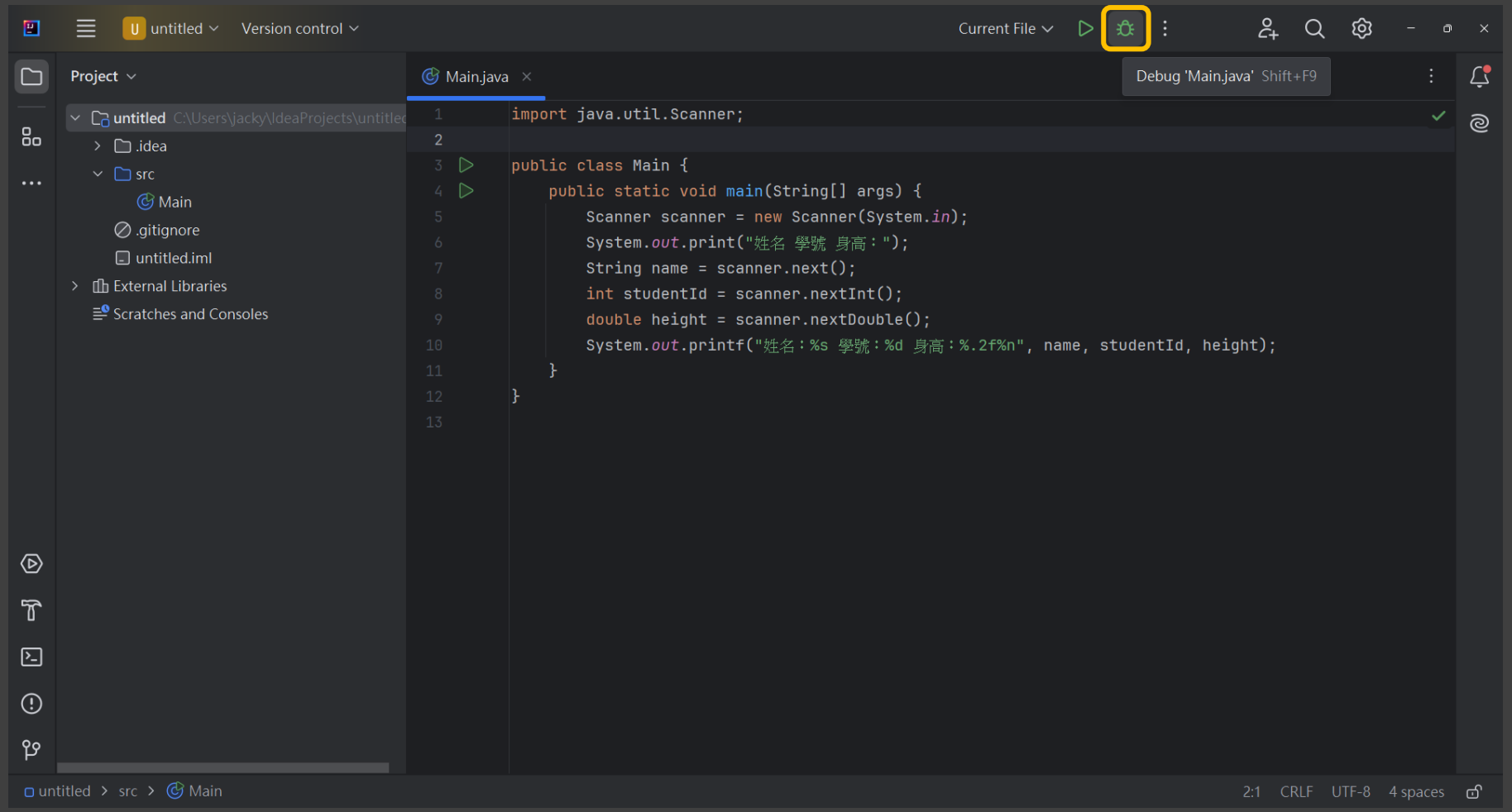


執行的輸入和輸出在下方的主控台
按下停止按鈕可停止執行
按下重啟按鈕可重新執行



除錯

按下右上角的
除錯按鈕
即可開始進行除錯



除錯

除錯(debug)比一般執行多的功能在於可以下斷點(breakpoint)

程式在下斷點後，當執行到下斷點的行前，就會先暫停

只需要在行編號上點擊左鍵即可下斷點，再按一次即可移除斷點

Main.java

```
1   import java.util.Scanner;
2
3   public class Main {
4       public static void main(String[] args) {
5           Scanner scanner = new Scanner(System.in);
6           System.out.print("姓名 學號 身高：");
7           String name = scanner.next();
8           int studentId = scanner.nextInt();
9           double height = scanner.nextDouble();
            System.out.printf("姓名：%s 學號：%d 身高：%.2f%n", name, studentId, height);
11      }
12  }
```

除錯

當程式停下後
即可進行許多操作
如查看、修改變數的值
也可以移除或加新斷點

除錯

若想要繼續執行，可以選擇：

恢復(resume)、步過(step over)、
步入(step in)、步出(step out)等

恢復就是程式繼續執行

步過就是執行該行，然後繼續暫停

步入與步出在更複雜的程式碼才能體現效果，之後會介紹

另外，還可以暫時忽略所有斷點