

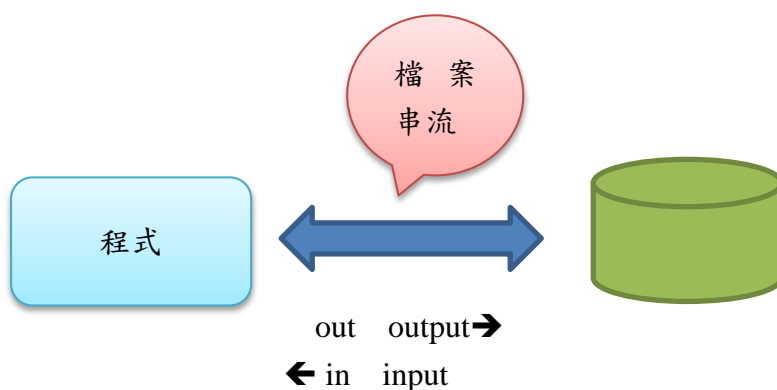
“電腦不可以關機，否則存在記憶體中的資料就不見了。安啦!我的資料都存到硬碟啦!!”

檔案與串流(Files and Streams)

簡介

目前為止的程式資料都是儲存在變數或是陣列當中，然而當程式結束或是區域變數離開其生存空間時，資料就會消失。

本章介紹檔案串流(stream)輸入、輸出的功能，建立檔案長久保存資料。



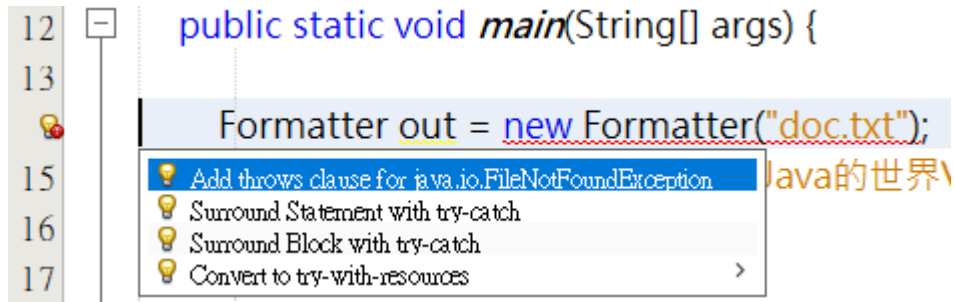
存檔格式可以是一般文字格式，也可以是二進位格式，這裡的範例只使用到文字格式。更詳細的內容材料請參考課本。

存檔案(Write)：一次寫入 非常簡單

```

public static void main(String[] args) throws FileNotFoundException {
    Formatter out = new Formatter("doc.txt");
    String doc1 = "大家好，歡迎來到 Java 的世界\n這裡有無盡的寶藏等你來挖角!\n";
    out.format("%s", doc1); //準備好 文字內容
    out.flush(); //馬桶沖水! 將緩衝區的資料串流 stream 沖出去
    out.close(); //蓋上馬桶 將資料串流 stream 關閉
}

```



另一種方式 try-catch

Surround Block with try-catch 把多行程式碼一起用 try 包圍(surround)起來
Surround statement with try-catch 只會包圍一行

```
public static void main(String[] args) {
    try {
        Formatter out = new Formatter("doc.txt");
        String doc1 = "大家好，歡迎來到 Java 的世界\n這裡有無盡的寶藏等你來挖角!\n";
        out.format("%s", doc1);
        out.flush();
        out.close();
    } catch (FileNotFoundException ex) {
        System.out.println("檔案存取錯誤!");
    }
}
```

讀取檔案(Read)：一行一行讀出 也非常簡單

```
1 Scanner in = new Scanner(new File("doc.txt"));
2 String msg = "";
3 while (in.hasNextLine()) {
4     msg = in.nextLine();
5     System.out.println(msg);
6 }
7 in.close(); //關閉串流
8 //再讀一次
```

9	<code>in = new Scanner(new File("doc.txt")); //水已經流完了 需重新接一次水管</code>
10	<code>while (in.hasNextLine()) {</code>
11	<code> msg = in.nextLine();</code>
12	<code> System.out.println(msg);</code>
13	<code>}</code>
14	<code>in.close(); //關閉串流</code>

實例:學生成績檔案讀寫

一行一行寫入成績檔案

1	<code>try {</code>
2	<code> Formatter out = new Formatter("score.txt");</code>
3	<code> out.format("%s,%d,%d\n", "Bill", 20, 80);</code>
4	<code> out.format("%s,%d,%d\n", "Tom", 90, 70);</code>
5	<code> out.close();</code>
6	<code> System.out.println("寫檔案完成!");</code>
7	<code>} catch (FileNotFoundException e) {</code>
8	<code> System.out.println("開檔案錯誤!");</code>
9	<code>}</code>

一行一行讀出成績檔案

1	<code>//一行一行讀入</code>
2	<code>Scanner input;</code>
3	<code>input = new Scanner(new File("score.txt"));</code>
4	<code>while (input.hasNextLine()) {</code>
5	<code> String data =input.nextLine();</code>
6	<code> System.out.printf("%s\n", data);</code>
7	<code>}</code>
8	<code>System.out.println();</code>
9	
10	<code>//處理學生成績</code>

11

```
String[] names;
int[][] grades;

int student_size = 0; //紀錄有幾位學生
int numgrades = 0; //紀錄有幾次成績

input = new Scanner(new File("score.txt"));
String row = "";
while (input.hasNextLine()) {
    student_size++;
    row = input.nextLine();
}

numgrades = row.split(",").length - 1; //扣掉前面一個姓名才是成績個數

System.out.printf("學生人數:%d \n", student_size);
System.out.printf("成績個數:%d \n", numgrades);
```

● 再讀一次 放入成績陣列與姓名陣列

```
grades = new int[student_size][numgrades];
names = new String[student_size];

input = new Scanner(new File("score.txt"));
int stu = 0; //學生索引
while (input.hasNextLine()) {
    row = input.nextLine();

    String[] s = row.split(","); //這裡改成各種分隔符號都可以!!

    names[stu] = s[0];
    for (int j = 0; j < numgrades; j++) {
        grades[stu][j] = Integer.parseInt( s[j + 1] );
    }
    stu++;
}

printGrades(grades, names);
```

● 列印學生姓名 成績

```
public static void printGrades(int[][] grades, String[] names) {
    for (int i = 0; i < grades.length; i++) {
        System.out.printf("%s ", names[i]);
        for (int j = 0; j < grades[i].length; j++) {
```

	<pre> System.out.printf("%d ", grades[i][j]); } System.out.println(); } }</pre>
--	---

讀檔案

The [java.util.Scanner](#) class is a simple text scanner which can parse primitive types and strings using regular expressions.

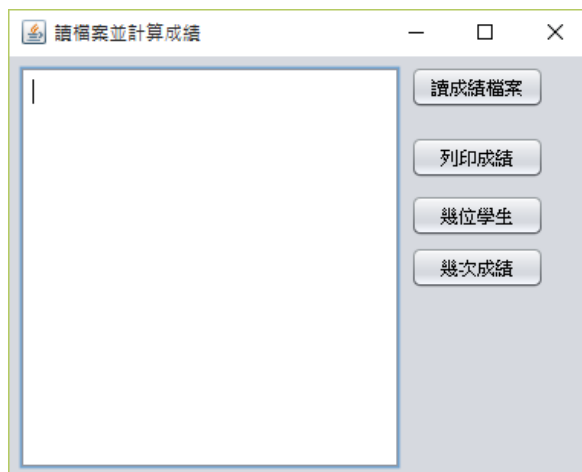
寫檔案

The [java.util.Formatter](#) class provides support for layout justification and alignment, common formats for numeric, string, and date/time data, and locale-specific output.

還有很多種讀寫檔案方式。

習題

請將上述 Console 的功能，改寫為 GUI 功能，參考畫面如下：
檔案名稱固定為“score.txt”



開檔/讀檔 GUI-存檔位置選單

使用者可以自行決定開檔以及讀檔的位置

當使用者按下「Write」鍵後，會出現一個儲存視窗如上圖，使用者可以自行決定檔案名稱以及欲存放的位置，若該路徑已存在同樣的檔名，則內容會被覆蓋；反之若該路徑沒有存在相同的檔名，則會自動建立一個文字檔。

```
File filename;
```

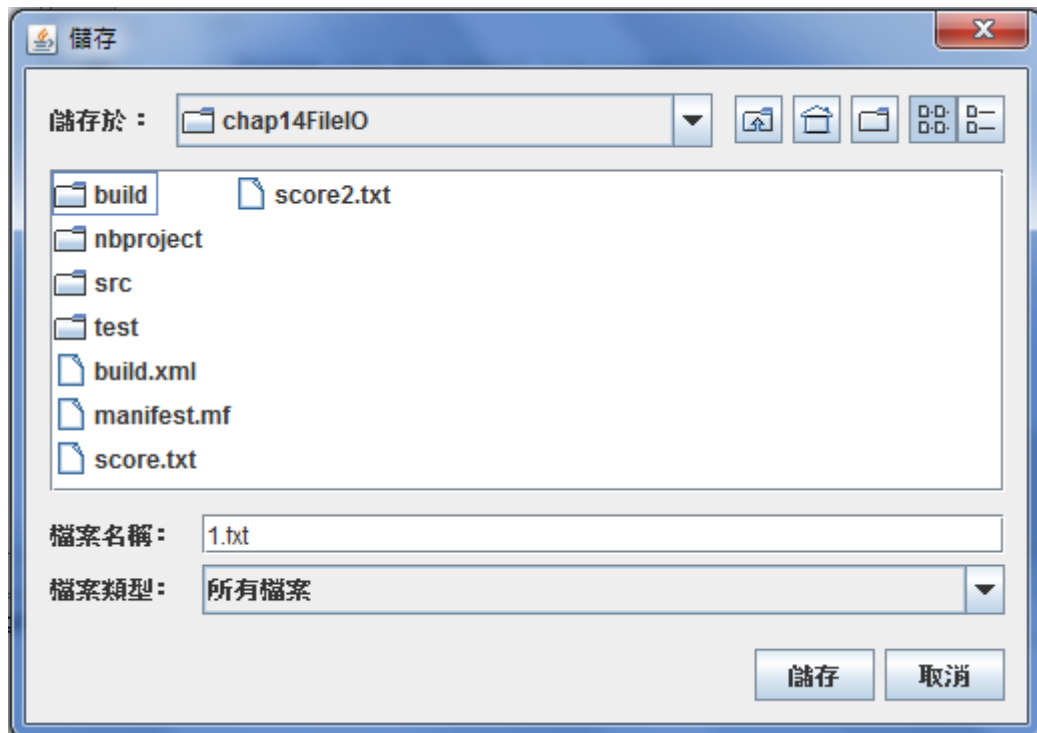
```
// 新增一個 Dialog 名為 ch 讓使用者選擇要儲存的檔案
```

```
JFileChooser ch = new JFileChooser();
```

```
ch.showSaveDialog(this); // 顯示儲存視窗 ch
```

```
filename = ch.getSelectedFile(); // 用 getSelectedFile()把檔案抓起來
```

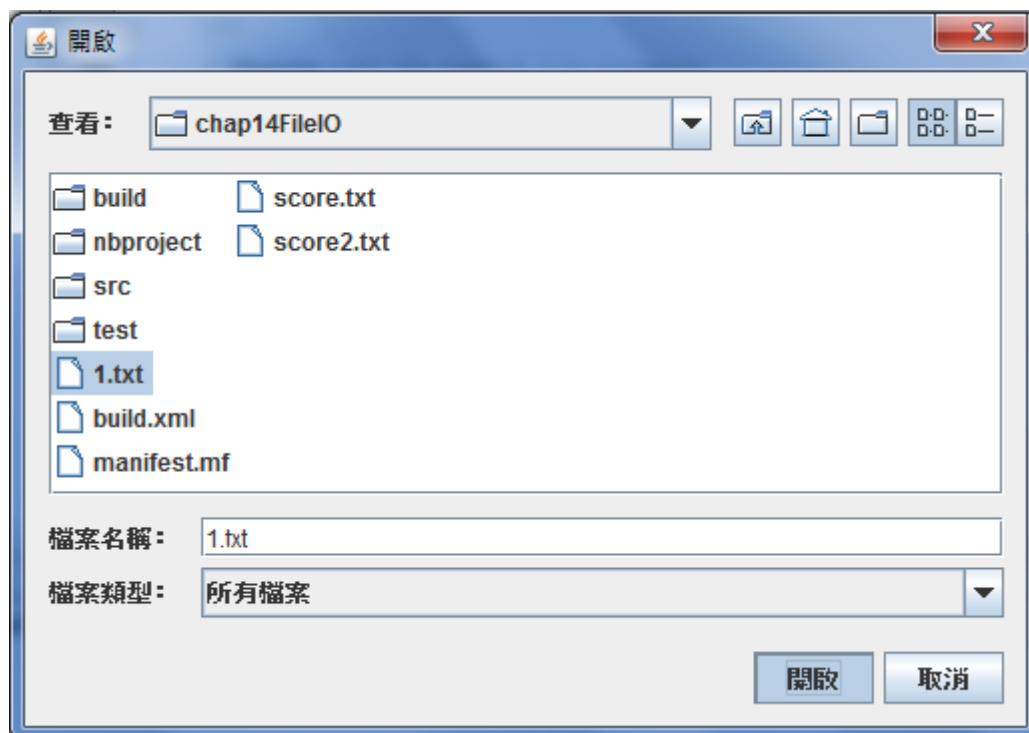
```
output = new Formatter(filename); // 將檔案儲存成 Formatter 名為 filename
```



按下「Read」鍵，同樣會出現一個對話視窗，用以開啟文字檔。

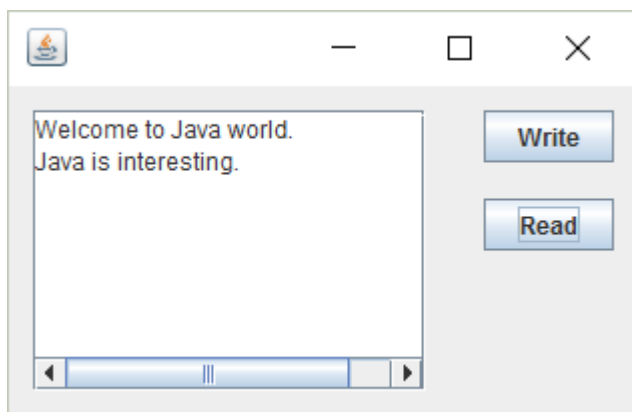
本範例選擇剛剛建立的「1.txt」，選擇後按「開啟」即完成，並顯示於TextArea。

```
JFileChooser ch = new JFileChooser();  
ch.showOpenDialog(this);  
filename = ch.getSelectedFile();  
Scanner input = new Scanner(filename);
```



習題

寫一個非常簡單又沒有甚麼功能，只能存檔讀檔的文字編輯器



按下「Write」後，讓使用者，存入多行文字資料

按下「Read」鍵，讀取文字檔案內容，顯示在 TextArea 多行文字資料

請搭配 JFileChooser 功能

```

1  try {
2      Formatter out = new Formatter("doc1.txt");
3      out.format("%s", JTextArea1.getText());
4      out.close();
5      System.out.println("寫檔案完成!");
6
7  } catch (FileNotFoundException e) {
8      System.out.println("開檔案錯誤!");
9  }

```

read

```

1  try { // 嘗試抓錯誤
2      // 直接用 new File 新增一個檔案 接到 Scanner
3      Scanner input = new Scanner( new File("doc1.txt") );
4      String msg = ""; // 先將 msg 定義成空字串(這樣接下來用+=的才能不重疊!!)
5      while ( input.hasNextLine() ) { // 只要還沒遇到 EOF(null)就繼續
6          String line = input.nextLine()+"\n";
7          JTextArea1.append( line );
8      }
9      } catch (FileNotFoundException ex) { // 抓例外(FileNotFoundException) 放到 ex
10         System.err.println("Error file!"); // 若有例外 顯示出現錯誤訊息
11         System.exit(1); // 跳出訊息
12     }

```