

# JAVA 基本觀念-陣列與字串

修平科技大學 資訊網路技術系  
陳文敬老師 jameschen@hust.edu.tw

# Agenda

1. 處理大量資料的必要性?
2. 處理大量資料有哪些選項?
3. 文字(字串)資料的處理
4. 利用迴圈(重複)結構來處理資料
5. 練習、練習、再練習 ...

# 紀錄與處理資料

- 宣告變數 → 儲存資料 → 處理資料

*int a, b, c ;*

- 現在全班有**50**個人，每個人有**10**次小考成績，你打算如何記錄、處理？

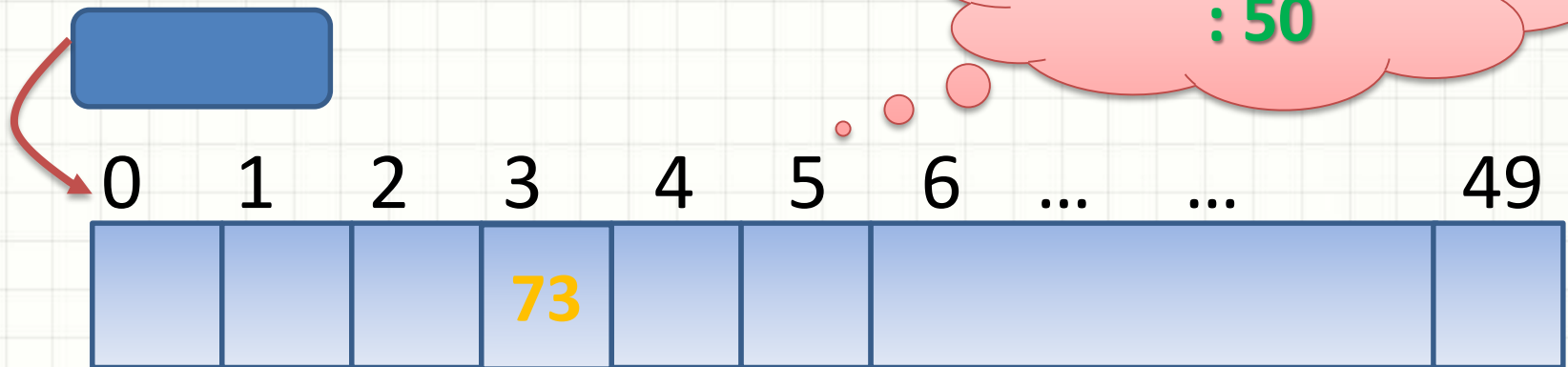
# 處理大量資料的可能選項(1)

- 陣列 (Array) – 固定大小、配置後不可變動

```
int[] scores;
```

```
scores = new int[50]; // 儲存 50 筆整數
```

```
scores[3] = 73;
```



# 處理大量資料的可能選項(2) \*

- 容器 (**Collection**) – 系統動態調控空間大小

// 清單：有順序、可重複

```
List<String> fruits = new ArrayList<String>();
```

```
fruits.add("Apple");
```

// 集合：無順序、不可重覆

```
Set<Integer> numbers = new TreeSet<Integer>();
```

```
numbers.add( 32 );
```

// 對應表：Key(鍵) → Value(值)

```
Map<String, String> enroll = new HashMap<String, String>();
```

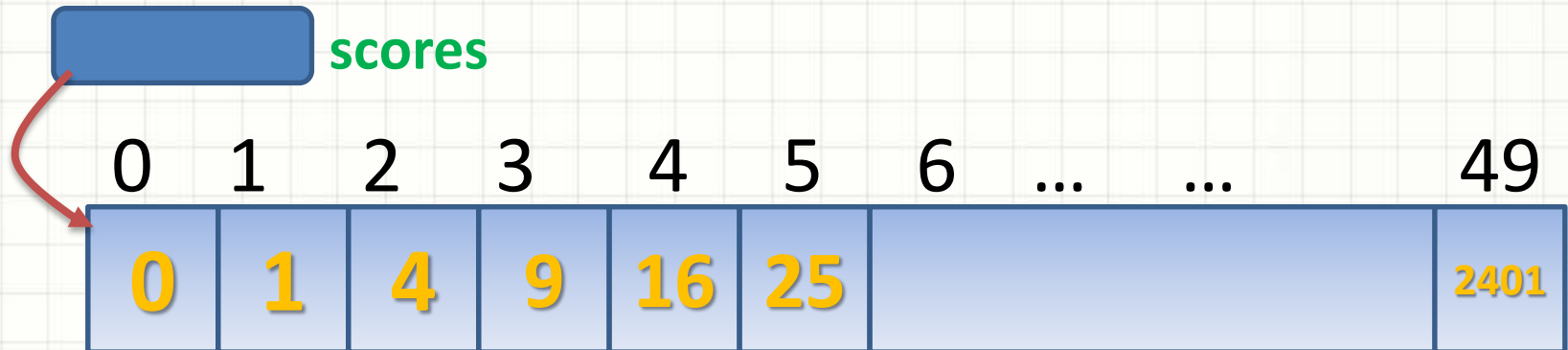
```
enroll.put("BN101999", "James Chen");
```

# 利用迴圈設定陣列元素內容值



```
int[] scores ;  
scores = new int[50];    // 儲存 50 筆整數  
for (int i=0; i<50 ; i++) {  
    scores[ i ] = i * i;  
}
```

# 利用迴圈讀取陣列元素內容值



```
for (int item : scores ) {  
    System.out.printf("%d,", item);  
}
```

# 一維陣列的記憶體配置與初始化

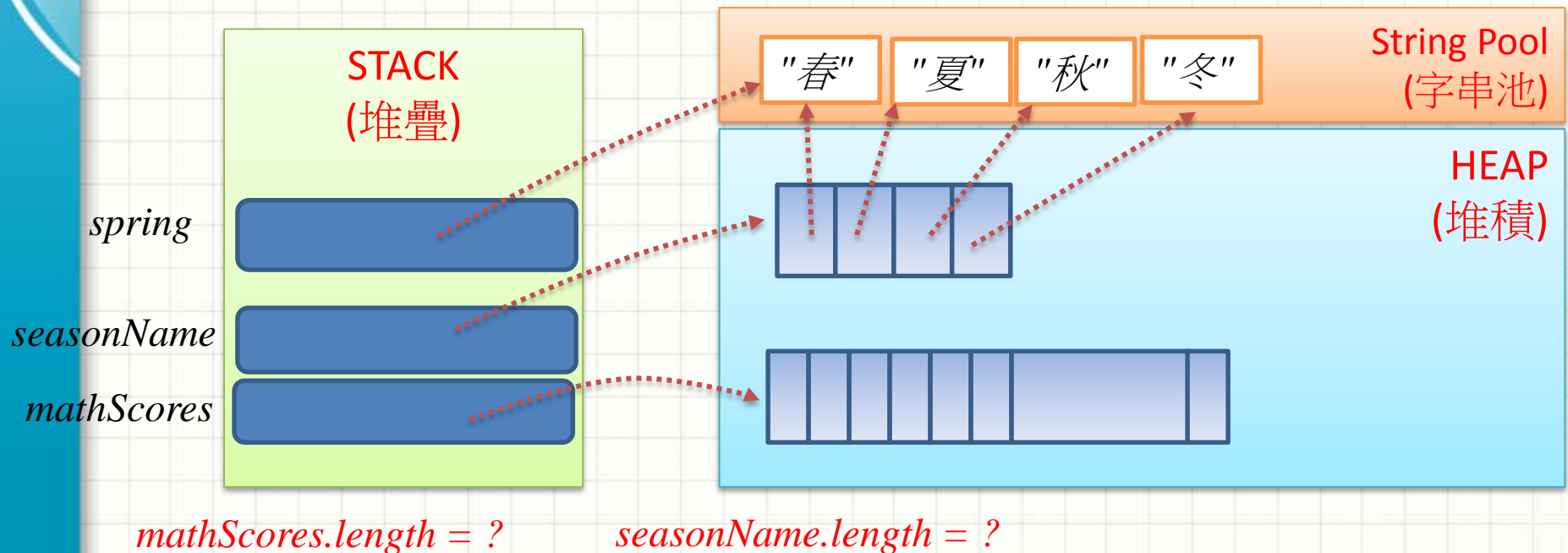
// 一維整數陣列:配置空間

```
int[] mathScores = new int[50]; // 單一科目
```

// 一維字串陣列: 給初始值

```
String[] seasonName = {"春", "夏", "秋", "冬"};
```

```
String spring = "春";
```



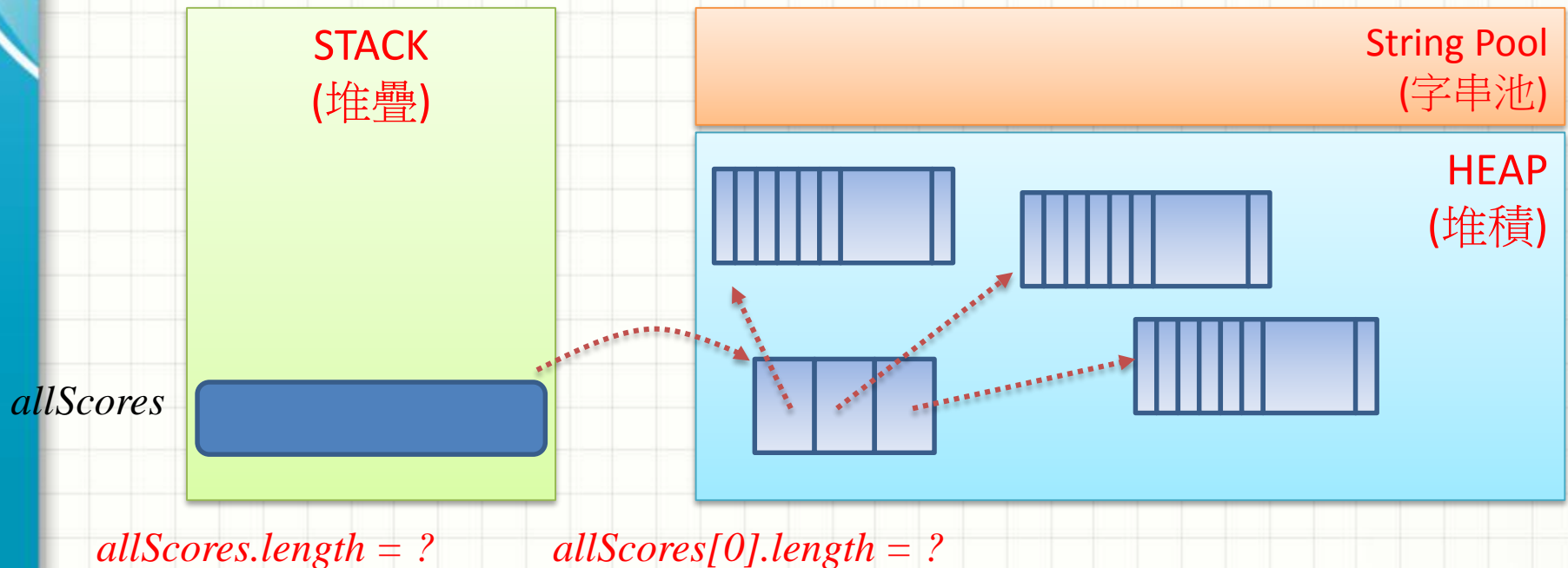


# 多維陣列的宣告與記憶體配置

// 二維整數陣列：配置空間

```
int[][] allScores = new int[3][50]; // 三種科目
```

→ 利用一維陣列模擬!!



# 多維陣列的宣告與初始化

// 二維整數陣列：給初始值

```
int[][] data = {  
    { 1, 2, 3},  
    { 4, 5, 6, 7},  
    { 8, 9, 10}  
};
```

// 三維整數陣列

```
int[][][] d3 = new int[5][10][50];
```

# 練習時間：ITSA (C\_AR01:一維陣列反轉 I)

- 題目敘述: 將一維陣列內的資料反轉後輸出。
- 輸入: 輸入一個一維陣列，元素最多不超過100個
- 輸出: 輸出反轉後的陣列, 最後需有換行。

輸入範例

```
4 6 3 69 234
56 89 23 3 1
176 5 890 643 2
0 500 6 634 55 123
87 77 32 22 111 4
```

輸出範例

```
234 69 3 6 4
1 3 23 89 56
2 643 890 5 176
123 55 634 6 500 0
4 111 22 32 77 87
```

# C\_AR01 : 一維陣列反轉 I

- 觀察：

- 每列資料數量是否一致？

- 思考：

- 如何反轉順序？

- 如何動態讀取不同數目的資料？

# \* 陣列相關的內建功能(運算)

- 複製陣列：全部、部分元素

*array2 = array1.clone( ); // 完全複製*

*System.arraycopy( 來源, from, 目的, to, length);*

- 內建 **Arrays** 類別

*Arrays.equals( array1, array2); // 比對*

*Arrays.fill( array, value); // 填充預設值(value)*

*Arrays.sort( array ); // 排序: 由小到大*

*Arrays.binarySearch( array, value); // 搜尋資料*

# 文字資料處理：*String* (字串)

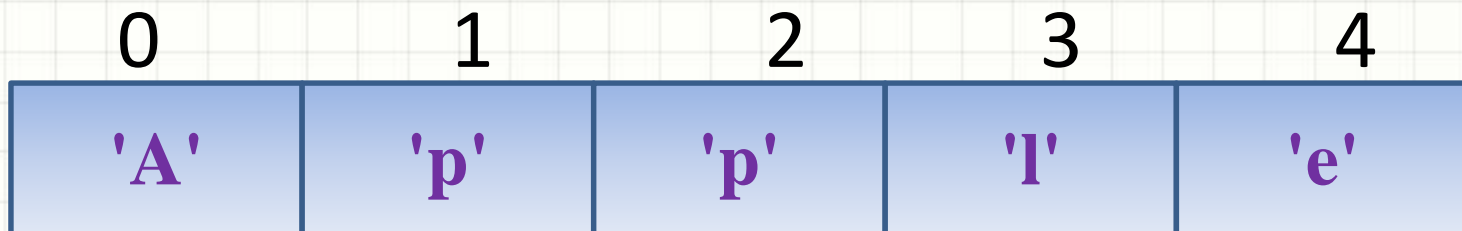
- *String* = a array of characters (=字元陣列)
- Java 定義了三個**模組(類別)**來處理文字資料
  - *String* :字串生成後，內容不可更動 (效能)
  - *StringBuffer* :可以動態增修字串內容(多工)
  - *StringBuilder* :可以動態增修字串內容(單工)
- 三個類別各有其對應的處理功能
  - **查詢 Java Docs APIs**  
<https://docs.oracle.com/javase/7/docs/api/>

# *String ? char[ ] ?*

- 為何不用 **char[ ]** ? 而要另外定義 **String** ?

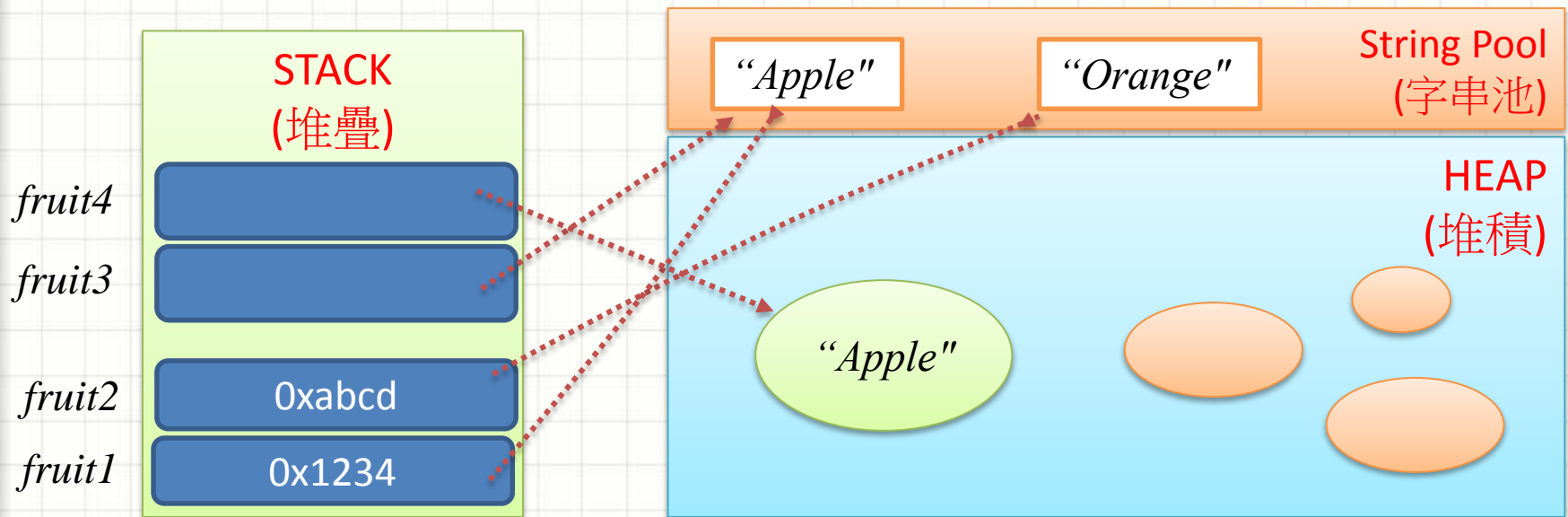
```
char[] data1 = { 'A', 'p', 'p', 'l', 'e' };
```

```
String str1 = "Apple";
```



# 字串物件的記憶體配置

- *String fruit1 = "Apple";*
- *String fruit2 = "Orange";*
- *String fruit3 = fruit1;*
- *String fruit4 = new String("Apple");*





# 數值與字串的比較?

- `int a=1, b=2, c=1;`

比對條件	評估結果
<code>a == b</code>	
<code>a == c</code>	
<code>s1 == s2</code>	
<code>s1 == s3</code>	
<code>s1 == s4</code>	

- `String s1="Apple";`

- `String s2="Apple";`

- `String s3="apple";`

- `String s4=new String("Apple");`

■ 如果要比較的是【內容】呢?

# String : 字串物件的常用方法

回傳值	功能(方法)名稱	功能說明	
char	<i>charAt(int index)</i>	取得指定位置的字元	從 0 開始
int	<i>length()</i>	字串總長度(字元個數)	
int	<i>indexOf(int ch)</i>	出現 <u>ch 字元</u> 的位置	從 0 開始
int	<i>indexOf(int ch, int from)</i>		
int	<i>indexOf(String str)</i>	出現 <u>str 字串</u> 的位置	
int	<i>indexOf(String str, int from)</i>		
boolean	<i>equals(String str)</i> <i>equalsIgnoreCase(String str)</i>	比對字串內容是否相同	大小寫 有無區別?
String[]	<i>split(String regex)</i>	根據 <u>regex</u> 切割字串	正規表示式
String	<i>substring( int pos)</i>	擷取部分字串內容	新字串物件
String	<i>substring( int begin, int end)</i>	(begin ~ end-1)	
String	<i>toUpperCase()</i>	全部轉為大寫字母	新字串物件
String	<i>toLowerCase()</i>	全部轉為小寫字母	
String	<i>trim()</i>	去除前後的"白字元"	新字串物件

<https://docs.oracle.com/javase/7/docs/api/java/lang/String.html>

# *StringBuffer* : 字串內容的處理

回傳值	功能(方法)名稱	功能說明	
char	<i>charAt( index)</i>	取得指定位置的字元	
StringBuffer	<i>substring( begin, end)</i>	擷取部分子字串 (begin~end-1)	不包含 end
void	<i>setCharAt(index, ch)</i>	設定指定位置的字元(=ch)	
StringBuffer	<i>deleteCharAt( index)</i>	刪除指定位置的字元	
StringBuffer	<i>append(data)</i>	添加新資料(data)到尾端	
StringBuffer	<i>insert(offset, data)</i>	插入資料(data)到指定位置	
StringBuffer	<i>replace(begin, end, data)</i>	將指定範圍 begin~(end-1) 更換為新內容data	不包含 end
StringBuffer	<i>delete(str)</i>	刪除指定字串	全部
StringBuffer	<i>reverse()</i>	將字串倒置	
String	<i>toString()</i>	將 StringBuffer 轉為 String	效率

# 練習時間：ITSA (C\_AR22:字母出現的頻率)

- **題目敘述**：通訊上面常常會用霍夫曼碼(Huffman Code)的方式來減少編碼的長度，但是要使用霍夫曼碼前，必須先知道字母出現的頻率，**請你寫一個程式，用來統計字母(不分大小寫)出現的次數。**
- **輸入**：  
輸入一行英文字。
- **輸出**：各字母出現的次數，第1個數字代表A出現的次數，第2個數字代表B出現次出，依此類推，第26個數字代表Z出現的次數。各數字間用一個空白間隔，最後一個數字後面沒有空白。

輸入範例

輸出範例

No pain,no gain!

2 0 0 0 0 1 0 2 0 0 0 0 4 2 1 0 0 0 0 0 0 0 0 0

<http://e-tutor.itsa.org.tw/e-Tutor/mod/programming/view.php?id=2200>

# C\_AR22 : 字母出現的頻率

- **觀察：**
  - 資料長度是否固定?
  - 大寫字母、小寫字母同時出現?
  - 26 個字母之外的其他字元?
- **思考：**
  - 如何讀取所有資料?
  - 如何切割出個別字元?
  - 大寫字母、小寫字母比對?
  - 如何記錄 26 個字母的出現次數?

# 建議練習題目

- **C\_AR21\_成績計算I**

- <http://e-tutor.itsa.org.tw/e-Tutor/mod/programming/view.php?id=2196>

- **C\_AR34\_身分證號驗證**

- <http://e-tutor.itsa.org.tw/e-Tutor/mod/programming/view.php?id=2200>

- **C\_AR76\_查詢ATM餘額**

- <http://e-tutor.itsa.org.tw/e-Tutor/mod/programming/view.php?id=3702>