CH4-5
陣列(Array)、字典(Dictionary)

概要 陣列(Array)

有大量的資料要使用時，我們就會用到陣列，先想像現在有一個一格一格的保管箱，陣列就像是一個由多個保管箱組合而成的組合櫃，當想要取出資料時，就必須透過有著該保險箱號碼牌來指定要打開哪個櫃子。

使用方法 陣列(Array)的建立

建立陣列

建立陣列時，使用中括弧[　]來表示；值與值之間，利用逗號來區分。

```
var 陣列名 = [值1, 值2, 值3, … ]
```

例

```
var IntArray = [5,2,0]
var StrArray = ["S","w","i","f","t"]
```

陣列的型別指定

建立陣列時，我們可以對其指定型別，但若指定的型別錯誤，Xcode

也會提出警告。

```
var 陣列名:[型] = [值1, 值2, 值3, … ]
```

例

```
var IntArray:[Int] = [5,2,0]
var StrArray:[String] = ["S","w","i","f","t"]

var IntSrray:[Int] = ["A","P","P"] //無法將字元放入int的陣列中
```

陣列的初始值設定

建立陣列時，可以為陣列宣告個數，也可以先放入初始值在內。

使用下列語法可將陣列內的值都設為相同的一個值。

```
var 陣列名 = Array(count : 個數 , repeatedValue : 值)
```

例

```
var repeatedIntArray = Array(count: 4, repeatedValue:7)          [7, 7, 7, 7]
var repeatedStringArray = Array(count: 3, repeatedValue:"SS")    ["SS", "SS", "SS"]
```

空陣列的製作

當需要建立一個空的陣列時，我們會使用中括弧[ ]來建置。

```
var 陣列名 =[  ]
```

例

```
var empArray = []
```

空陣列的型別指定

建立空陣列時，我們也可以對其指定型別。

```
var 陣列名: [型] = [  ]
var 陣列名 = [型] ( )
```

例

```
var empArray = [Int]()
```

使用方法 陣列(Array)的取得

陣列的資料個數

當要得知陣列中的資料個數時，可以使用count屬性來取得。

陣列名.count

例

```
var StrArr = ["S","W","I","F","T"]                    ["S", "W", "I", "F", "T"]
var count = StrArr.count                              5
```

將 count 屬性加至後方後，便可以取得 StrArr 內值的個數，這裡要注意的地方是，一定要有一個值用來紀錄 count，否則無法使用。

指定序號取出陣列內容

當要從陣列取出資料時，可以指定其擺放的序號，來取出櫃子裡的值，

由於序號是從0開始，因此若有五筆資料，則只會有序號0,1,2,3,4。

陣列名[序號]

例

```
var StrArr = ["S","W","I","F","T"]                    ["S", "W", "I", "F", "T"]

var temp = StrArr[0]                                  "S"
var temp2 = StrArr[4]                                 "T"
var temp3 = StrArr[5]                                 error
Execution was interrupted, reason: EXC_BAD_INSTRUCTION (code=EXC_I386_INVOP, subco...
```

切記序號是由 0 開始，若超出範圍，則會顯示錯誤，如圖中欲取出序號 5，即顯示錯誤，因為序號只到 4 而已。（這是新手常犯的錯誤，需要特別注意。）

指定序號範圍取出陣列內容

想要列出範圍內的值時，必須要先知道開始的序號及結束的序號，才能進行範圍指定。

```
陣列名[ 開始序號...結束序號 ]
```

例

```
var StrArr = ["S","W","I","F","T"]                    ["S", "W", "I", "F", "T"]
var temp = StrArr[0...3]                              ["S", "W", "I", "F"]
```

查看陣列中的所有資料

我們可以使用 for in 迴圈來依序顯示放在陣列中的資料。

```
for 欲裝入變數的位置 in 陣列名 {                    }
```

例

```
var StrArr = ["S","W","I","F","T"]                    ["S", "W", "I", "F", "T"]

for temp in StrArr {
    print("值=" + temp)                               (5 times)
}
```

將 StrArr 內的值逐一放入 temp 中並顯示出來。

顯示結果

```
值=S
值=W
值=I
值=F
值=T
```

我們也可以使用 for in 迴圈搭配上 enumerate，使序號也一併顯示。
但因 Swift 改版之關係，因此這裡語法上會有點小小不同，請留意。

Swift 1.0 使用以下方法

for(序號，欲裝入變數的位置) in enumerate(陣列名) {　　}

Swift 2.0 請改為使用以下方法

for(序號，欲裝入變數的位置) in 陣列名.enumerate( ) {　　}

例

(範例為使用 Swift2.0 之方法)

```
var StrArr = ["S","W","I","F","T"]

for (number,temp) in StrArr.enumerate() {
    print("陣列[\(number)] = \(temp)")
}
```

```
["S", "W", "I", "F", "T"]

(5 times)
```

程式的最後一行為列印的部分。在這裡因為我們有多個參數要一起列印出來，因此，我們將參數加上括號，並使用” \ ”來告訴程式，接下來是一個參數。

顯示結果

```
陣列[0] = S
陣列[1] = W
陣列[2] = I
陣列[3] = F
陣列[4] = T
```

使用方法 陣列(Array)的使用

增加陣列的資料

若要將一筆資料加入陣列中，我們可以使用以下方法。

陣列名.append(欲加入的資料)

例

```
var StrArr = ["S","W","I"]                                  ["S", "W", "I"]

StrArr.append("F")                                          ["S", "W", "I", "F"]
StrArr.append("T")                                          ["S", "W", "I", "F", "T"]
```

若要將另一個陣列加入現有的陣列中，我們可以使用以下方式。

原陣列名+=欲加入的陣列

例

```
var StrArr = ["S","W","I"]                                  ["S", "W", "I"]

var AnotherArr = ["F","T"]                                  ["F", "T"]

StrArr += AnotherArr //將後者陣列的值接在前者陣列的值後面      ["S", "W", "I", "F", "T"]
```

若要將一筆資料指定加入陣列的某個位置，可以使用以下方法。

陣列名.insert(資料, atIndex:指定位置)

例

```
var StrArr = ["S","W","I","T"]                              ["S", "W", "I", "T"]

StrArr.insert("F", atIndex:3 ) //指定接在序號3的位置，因此T被擠到序號4  ["S", "W", "I", "F", "T"]
```

刪除陣列的資料

若要將最後一筆資料從陣列中刪除，我們可以使用以下方法。

```
陣列名.removeLast( )
```

例

```
var StrArr = ["S","W","I","F","T"]                    ["S", "W", "I", "F", "T"]
StrArr.removeLast();                                  "T"
StrArr                                                ["S", "W", "I", "F"]
```

這裏我們將 T 移除，則最後顯示的陣列已沒有 T

若是要將陣列中指定的某一筆資料從陣列中刪除，可使用以下方法。

```
陣列名.removeAtIndex( )
```

例

```
var StrArr = ["S","W","I","F","T"]                    ["S", "W", "I", "F", "T"]
StrArr.removeAtIndex(2);                              "I"
StrArr                                                ["S", "W", "F", "T"]
```

在這裡我們指定將序號 2 的 I 移除，切記序號是由 0 開始編號。

若是要將整個陣列全部刪除，可使用以下方法。

```
陣列名.removeAll ( )
```

例

```
var StrArr = ["S","W","I","F","T"]                    ["S", "W", "I", "F", "T"]
StrArr.removeAll();                                   []
StrArr                                                []
```

陣列內的值都已被刪除，因此顯示出來的陣列則為空陣列。

概要 字典(Dictionary)

試想，字典像陣列一樣是一個大保險櫃，保險櫃上有一格一格的保險箱。
不同的是，陣列的每一小格是由號碼牌（即為序號）來對應並打開。
但字典則是每個保險箱都有自己的名字（我們稱它為索引鍵，Key），當我們需要某個保險箱內的資料時，就必須要給它Key值，才能拿到資料。
由於這樣的方式就像我們在查字典的過程，這就是Dictionary稱為字典的由來。
每個值(value)都有屬於自己的一個獨一無二的索引鍵(key)

使用方法 字典(Dictionary)的建立

建立陣列

建立陣列時，使用中括弧[　]來表示；值與值之間，利用逗號來區分。

```
var 字典名 = [值1:資料1, 值2:資料2, 值3:資料3, ... ]
```

例

```
var StrDic = ["A":"hello","B":"good","C":"Morning"]//key值為文字
var intDic = ["A":1, "B":2, "C":3]//key值為文字，資料為整數
var numberDic = [1:"hello",2:"good",3:"Night"] //key值為整數
```

建立空陣列並指定型別

建立陣列時，使用中括弧[　]來表示；值與值之間，利用逗號來區分。

```
var 字典名 = Dictionary <型別,型別> ( )
```

例

```
var empDic = Dictionary<String,String>()
```

使用方法 字典(Dictionary)的取得

字典的資料個數

當要得知字典中的資料個數時，可以使用count屬性來取得。

```
字典名.count
```

例

```
var strDic = ["HK":"Honkong", "TPE":"TIPEI", "JAN":"JAPAN"]
var count = strDic.count
```

```
["JAN": "JAPAN
3
```

取出字典的資料

當要得知字典中的資料個數時，可以使用count屬性來取得。

```
字典名[ 值 ]
```

例

```
var strDic = ["HK":"HONKONG", "TPE":"TIPEI", "JAN":"JAPAN"]
var temp = strDic["TPE"]
```

```
["JAN": "JAPAN", "TI
"TIPEI"
```

在這裡我們給了字典 Key 值”TPE”，而他找到了我們要的資料”TIPEI”

例

```
var strDic = ["HK":"HONKONG", "TPE":"TIPEI", "JAN":"JAPAN"]

if let temp = strDic["TPE"] {
    print("找到= \(temp) ")
} else {

print("沒找到")

}
```

```
["JAN": "JAPAN", "TPE": "TIPEI",

"找到= TIPEI \n"
```

但是為了防止因為沒找到而造成程式錯誤，因此我們建議在這樣的情況時，使用 if else 判斷是否有找到，再顯示出相對應的結果。

查看字典中的所有資料

我們可以使用 for in 迴圈來依序顯示放在字典中的資料。

for (key值, 資料) in 字典名 {　　}

例

```
var strDic = ["HK":"Honkong", "TPE":"TIPEI", "JAN":"JAPAN"]
for(key,temp) in strDic{
    print("strDic[\(key)]=\(temp)")
}
```

```
["JAN": "JAPAN", "TPE": "TIPEI",

(3 times)
```

顯示結果

```
strDic[JAN]=JAPAN
strDic[TPE]=TIPEI
strDic[HK]=Honkong
```

增加字典的資料

若要將一筆資料加入字典中，我們可以使用以下方法。

字典名[欲加入的資料之key值] = 資料名

例

```
var strDic = ["TPE":"TIPEI", "JAN":"JAPAN"]
strDic["KOR"] = "KOREA"
strDic
```

```
["JAN": "JAPAN", "TPE": "TIPEI"]
"KOREA"
["JAN": "JAPAN", "TPE": "TIPEI", "KOR": "KOREA"]
```

刪除字典的資料

若要將最後一筆資料從字典中刪除，我們可以使用以下方法。

字典名.removeValueForKey(欲刪除的資料之key值)

例

```
var strDic = ["TPE":"TIPEI",
    "JAN":"JAPAN","KOR":"KOREA"]
strDic.removeValueForKey("KOR")
strDic
```

```
["JAN": "JAPAN", "TPE": "TIPEI", "KOR": "KOREA"]

"KOREA"

["JAN": "JAPAN", "TPE": "TIPEI"]
```