

Kotlin 讀書會 S4E9

補充資料

請關注第四梯次讀書會活動：<https://tw.kotlin.tips/study-jams/4>

共讀書本及討論群

- 導讀投影片：<https://reurl.cc/GoEnd3>
- 活動錄影：<https://youtu.be/NXX4kWOO4Z0>
- 共讀書本：<https://www.tenlong.com.tw/products/9789865021870>
- 讀書會討論群
 - Line 群 https://line.me/R/ti/g/jAG_c1DMaO (請先加聖佑 line id: shengyoufan 後傳訊息給聖佑, 聖佑再邀請入群)
 - Telegram 群 https://t.me/joinchat/DAUjOBfu7_05sa9YYb5FQ

導讀筆記

- Array
 - 特性: 陣列的內容可變, 無法改變陣列大小, 項目不可以移除

```
val arr1 = arrayOfNulls<Any>(2) //兩個陣列的空間
arr1[1] = 3
arr1[0] = "array"
println(arr1) //arr1變數的記憶體位置
println("arr1 = ${arr1.toList()}") //arr1變數的內容

val arr2 = arrayOf(1, 3, "a")
println("arr2 = ${arr2.toList()}")
arr2[2] = 5
println("size = ${arr2.size}")

//arr2.average() //無法算平均

val arr3 = arrayOf(1,6,6,7,9,30)
//相當於 val arr3:Array<Int> = arrayOf(1,6,6,7,9,30)
println(arr3.average())
println(arr3.sum())
println(arr3.minOrNull()) //課本的min()跟max()目前已經棄用了
println(arr3.maxOrNull())
```

```

//取用arr3裏頭的值
//forEach一個一個照順序讀出
arr3.forEach {
    println(it)
}
//reverse反向一個一個讀取
arr3.reverse()

```

- List
 - 特性: List有順序性, 內容可重複, 具有索引, 內容不可變

```

val list1 = listOf(1, 2, "a", "c")
println(list1)

//val a:Int = list1[1] 直接取值會出錯, 因為型態是Any, 要變成Int必須要先轉型
val a: Int = list1[1] as Int

//因此建議List先指定型態, 用Any的話取用通常都要轉型才能使用
val list2: List<String> = listOf("Tea", "Eggs", "Milk")
val i = list2[2]

val list3: List<Int> = listOf(1, 5, 7, 19) //跟Array很像
println(arr3.sum())
println(arr3.average())
println(arr3.minOrNull())

val list4: List<String>
list4 = listOf("a", "b", "c", "d")
if (list4.contains("c")) { //找到list裡有c字串的index
    println(list4.indexOf("c"))
}

data class Student(val id:Int, val name:String, val age:Int)
val students:List<Student>
val maggie = Student(1,"Maggie", 18)
val johnson = Student(2,"Johnson",19)
students = listOf(
    maggie,
    johnson,
    Student(3, "Kevin", 15),

```

```

        Student(5, "Teng", 25)
    )
    println(students) //印出學生名單

    //過濾學生小於20歲, 並印出學生名字
    students.filter{
        it.age < 20
    }.forEach{
        println(it.name)
    }
    //指定只取前面兩個學生
    println(students.take(2))
    //移掉前面兩個學生
    println(students.drop(2))
    //就算做了take跟drop, 並不會更動到原本的學生名單
    println(students)

    //對學生的年紀由小到大進行排序
    students.sortedBy {
        it.age
    }.forEach{
        println(it.name)
    }
    //對學生的年紀由大到小進行排序
    students.sortedBy {
        it.age
    }.reversed().forEach{
        println(it.name)
    }
}

```

- Set
 - 特性: 無序性, Set的內容值是不重複的, 遍歷取得

```

val lottery:Set<Int> = setOf(1,5,7,19,45,23,5) //故意多放一個5
println(lottery) //重複的5會被濾掉只剩一個[1,5,7,19,45,23]

val set2 = setOf<String>("a", "z","x", "d", "a")
val set3 = setOf<String>("a", "c", "r", "d")
//兩個Set加起來
println(set2.plus(set3)) //結果會變成 [a,z,x,d,c,r] 重複的值也會過濾掉
//兩個Set組合起來

```

```
println(set2.zip(set3))//結果會變成[(a,a),(z,c),(x,r),(d,d)]

//Set裡也可以放物件
data class Student(val id:Int, val name:String, val
age:Int)
val maggie = Student(1,"Maggie", 18)
val johnson = Student(2,"Johnson",19)
val set = setOf<Student>(
    maggie,
    johnson,
    maggie,
    Student(1,"Maggie", 18)
)
println(maggie == Student(1,"Maggie", 18))//結果是true,判定
是重複的內容
println(set)//只會有一個maggie跟johnson的內容
```

- Map
 - 特性: 不可變

```
data class Student(val id:Int, val name:String, val
age:Int)
val students = mapOf(
    "Kevin" to Student(3, "Kevin", 15),
    "Teng" to Student(5, "Teng", 25)
)
println(students["Kevin"])//用key取出map的值
//檢查是否map內有對應的key
if(students.containsKey("Maggie")){
    println(students["Maggie"])
}else{
    println("found")
}
//檢查是否map內有對應的值
val a = Student(5, "Teng", 25)
if(students.containsValue(a)){
    println("exists")
}
//讀出map的key跟value
for((key, value ) in students){
    println("key = ${key}: value:${value}")
}
```

- List, Set, Map都有內容不可變的特性, 若要可以變更, 就使用mutable
 - MutableList

```
val list2: MutableList<String> = mutableListOf("Tea",
"eggs", "Milk")
list2[2] = "Hot Tea"//Milk會更改成Hot Tea
val list2_1 = listOf("Tea", "Cold Tea", "Coffee")
list2.addAll(list2_1)//加上list2_1
println(list2)
//用Set把list2過濾重複的內容
println(list2.toSet())
//remove某一個值
list2.remove("Tea")
println(list2)
```

- MutableSet
- MutableMap

```
val students:MutableMap<String, Student> =
mutableMapOf(
    "Kevin" to Student(3, "Kevin", 15),
    "Teng" to Student(5, "Teng", 25)
)
val maggie = Student(1,"Maggie", 18)
val johnson = Student(2,"Johnson",19)
//增加新的學生, 用put
students.put("Maggie", maggie)
students.put("Johnson",johnson)
for((key, value ) in students){
    println("key = ${key}: value:${value}")
}
```

- 靜態的記憶體空間是固定, 佔較少資源, 可變動的會佔較多, 所以如果內容是固定不變可盡量使用List, Set, Map

參考資料

-