程式設計與應用(一)補充講義

蔡欣男

資料型別–可變與不可變

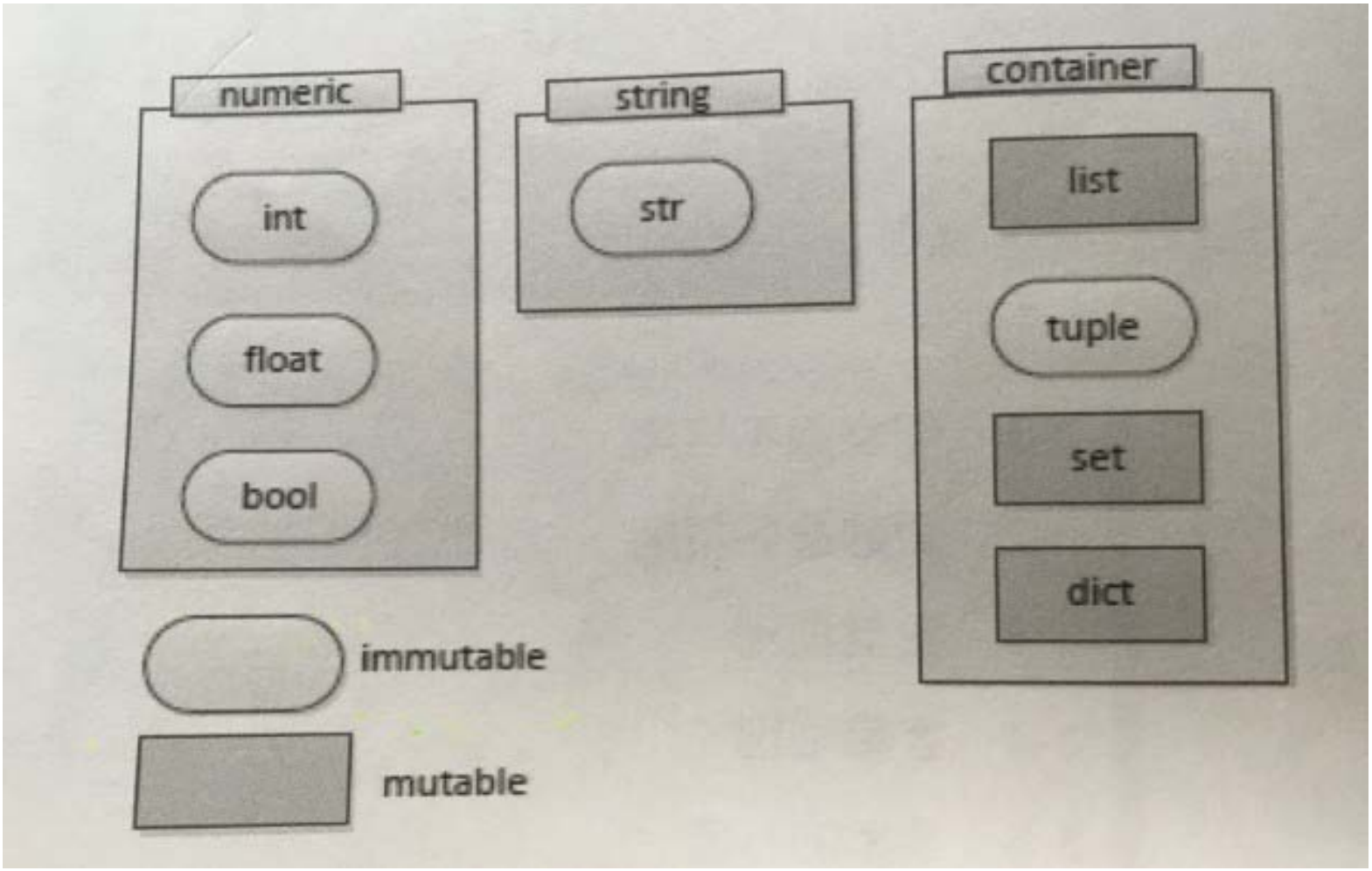
- 數值包含整數(int)、浮點數(float)、布林(bool)；容器包含串列(list)、元組(tuple)、集合(set)、字典(dict)等。字串與元組是屬於不可變的(immutable)；串列、集合與字典是屬於可變的(mutable)。

- 不可變的：意指物件一旦被建立，它的內容值就是固定不變。如果內容值被改變，則原有的物件與變數的連結關係即重新建立，舊物件由系統自動收回。int、float、bool、str、tuple是屬於不可變的資料型別。

- 可變的：意指元素內容可以改變，且不影響物件與變數的連結關係。list、set、dict是屬於可變的資料型別。

資料型別–可變與不可變

- 資料型別分可變(mutable)與不可變(immutable)



函式的呼叫–傳址與傳值

- 呼叫程式與被呼叫的函數之間，其參數的傳遞方式分為 **傳址(call by reference)** 與 **傳值(call by value)**。

- **傳址**：傳入參數為參考的記憶體位址，函數內如對此參數進行修改，會影響原參數的內容值。因為，傳入者與接收者同一個記憶體位址。
- **傳值**：複製一份傳入參數的內容值，給函數內負責接收的參數，兩者是互相獨立的參數，不同記憶體位址，若函數內如對此參數進行修改，不會影響原參數的內容值。

提示：

- 不可變的資料型別屬傳值的參數傳遞方式。
- 可變的資料型別屬傳址的參數傳遞方式。

函式的呼叫–傳址與傳值

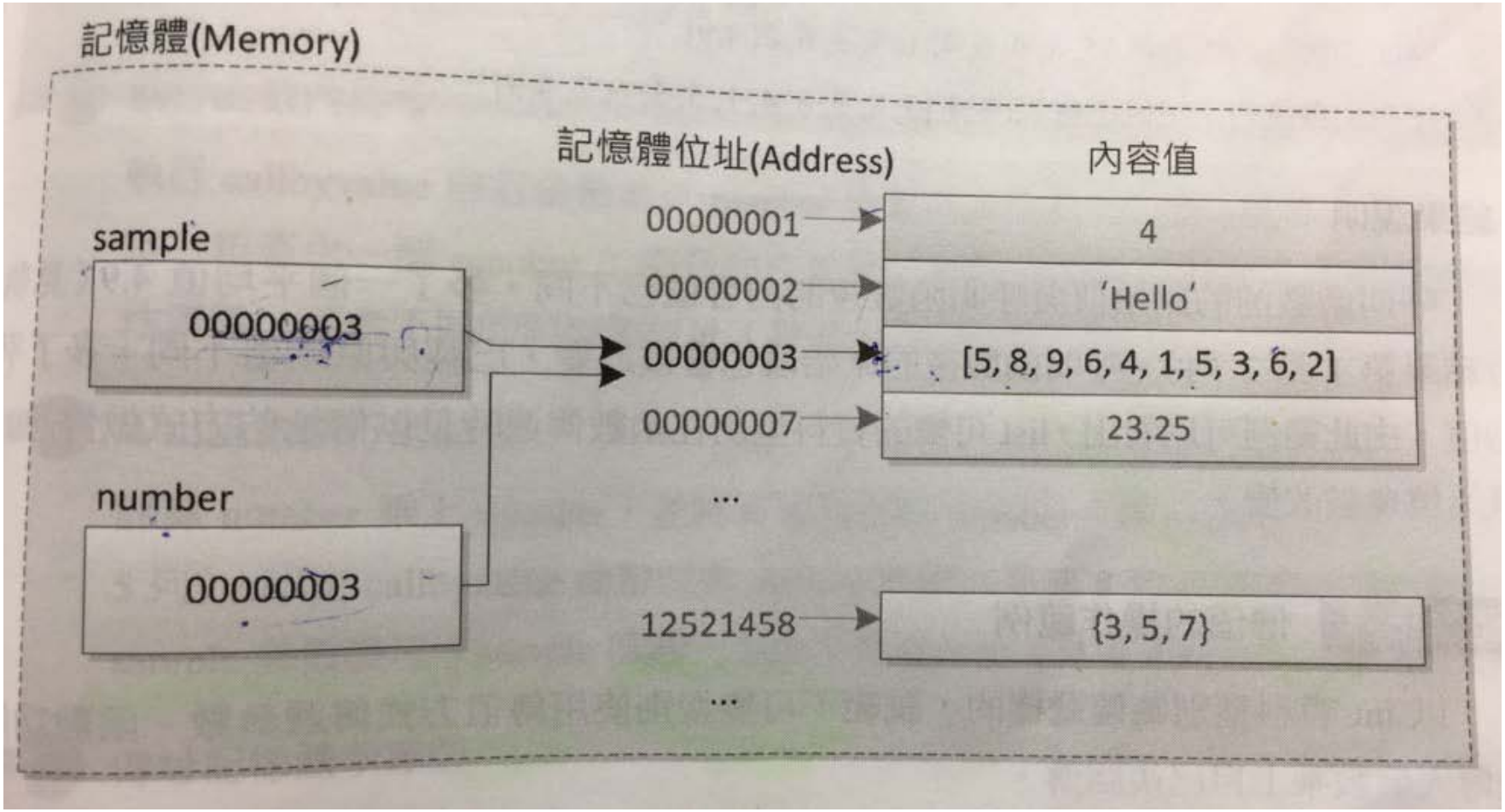
- 傳址的操作範例：

  以list資料型別為傳遞標的，示範可變型別使用傳址方式傳遞參數。

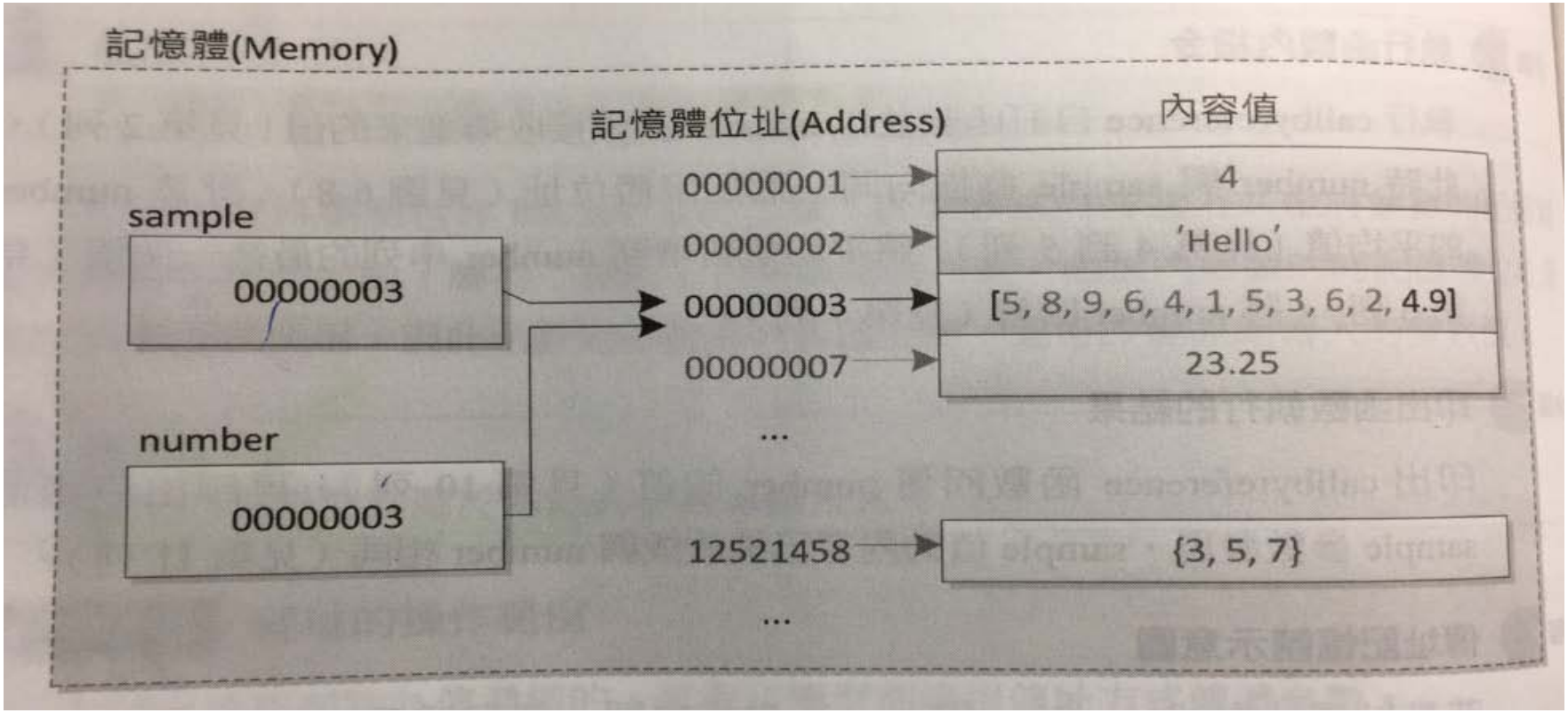
```
def callbyreference(number):
    # callbyreference 函數功能: 計算平均新增到傳入參數之後
    n = len(number)
    meanv = sum(number) / n
    number.append(meanv)
    return number

sample = [5, 8, 9, 6, 4, 1, 5, 3, 6, 2]

print("呼叫函數前的原始值", sample)

print("呼叫函數後的內容值", callbyreference(sample))

print("呼叫函數後的原始值也會被改變", sample)
```

函式的呼叫–傳址與傳值

- 傳址方式的記憶體位址參考示意圖



- 內容改變後的傳址參數示意圖



- 執行結果：

```
呼叫函數前的原始值 [5, 8, 9, 6, 4, 1, 5, 3, 6, 2]
呼叫函數後的內容值 [5, 8, 9, 6, 4, 1, 5, 3, 6, 2, 4.9]
呼叫函數後的原始值也會被改變 [5, 8, 9, 6, 4, 1, 5, 3, 6, 2, 4.9]
```

函式的呼叫–傳址與傳值

- 傳值的操作範例：

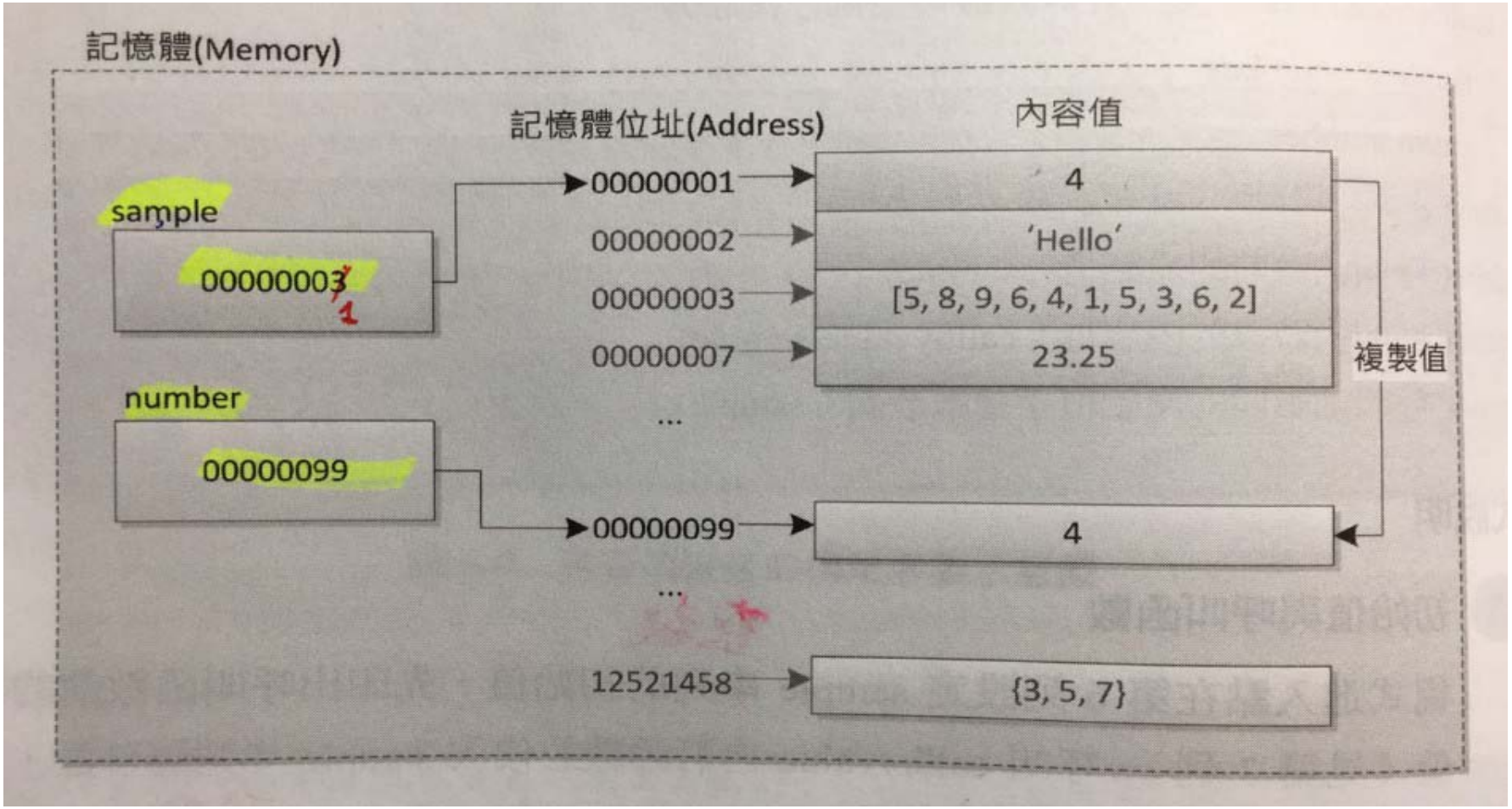
  以int資料型別為傳遞標的，示範不可變型別使用傳值方式傳遞參數。

  函數功能是傳入參數乘上自己後回傳。

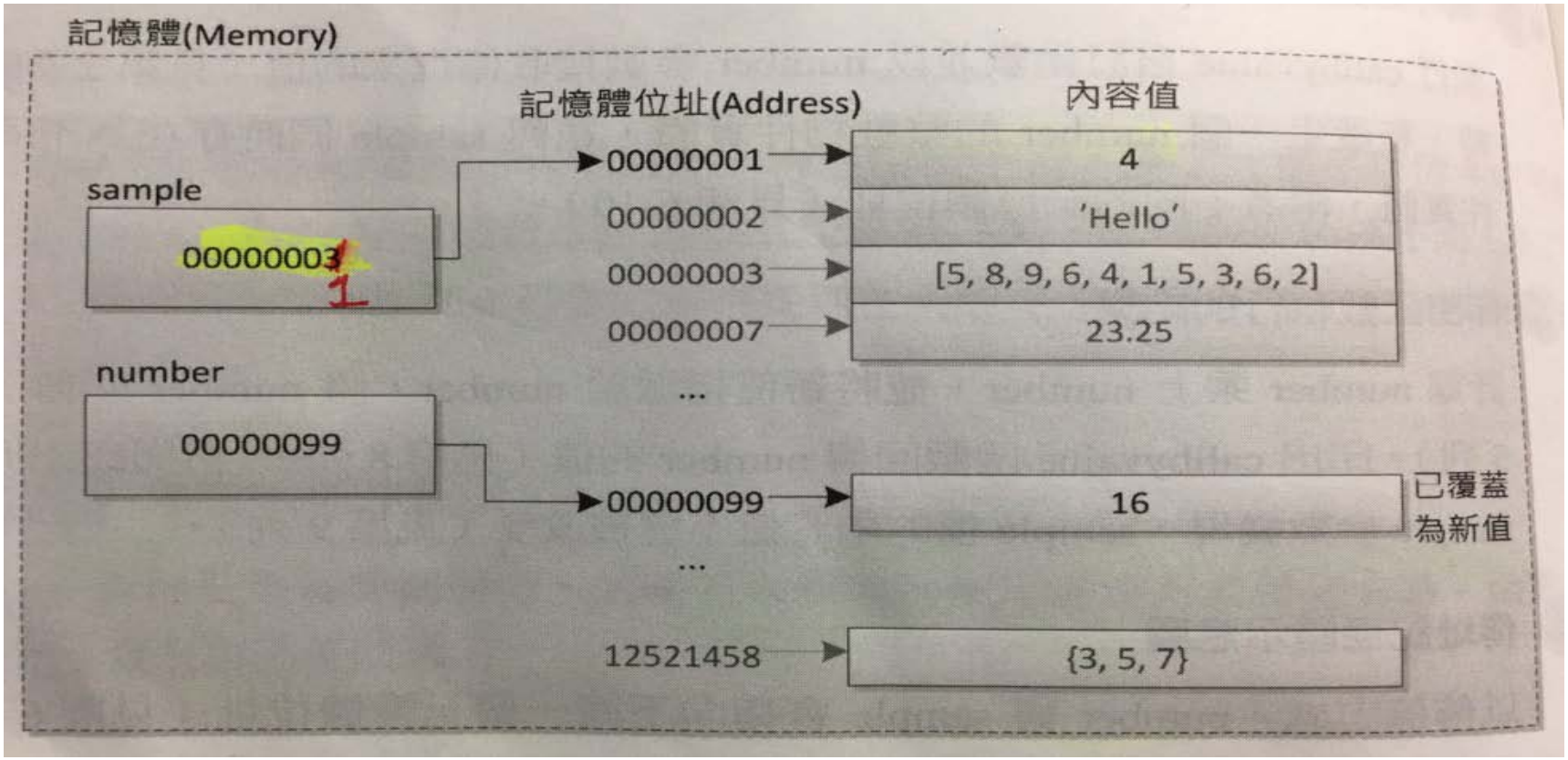
```
def callbyvalue(number):
    # callbyvalue 函數功能: 計算參數乘上自己後回傳
    number *= number
    return number

sample = 4

print("呼叫函數前的原始值", sample)

print("呼叫函數後的內容值", callbyvalue(sample))

print("呼叫函數後的原始值不會被改變", sample)
```

函式的呼叫–傳址與傳值

- 傳值方式之複製值



- 傳值方式之覆蓋新值



- 執行結果：

呼叫函數前的原始值 4
呼叫函數後的內容值 16
呼叫函數後的原始值不會被改變 4