Lecture note page 10: 做完後和朋友交換答案

Python中做備註的語法：

<https://chatgpt.com/share/a7c063d0-77b5-49e5-b0e6-ddede4efcb9f>

多行備註或是文檔字串 (docstring) -> “”” OR ‘’’

一行備註 -> #

注意：超級無敵重要的觀念: <https://chatgpt.com/share/229244f3-bfff-4d27-bfbd-cabbb4dce047> ; <https://chatgpt.com/share/229244f3-bfff-4d27-bfbd-cabbb4dce047>

在Python中，變數名(variable)存儲的是引用，而不是直接存儲數值。

當我們在Python中定義一個變數並賦予它某個值時，實際上變數名指向的是這個值所在的內存位置，換句話說，變數名存儲的是對這個值的引用。

Python中的對象分為可變對象和不可變對象，這兩者在變數引用上的行為有所不同。

不可變對象：如整數（int）、浮點數（float）、字符串（str）、元組（tuple）等。當這些類型的變數改變值時，實際上是創建了一個新對象，然後變數引用新的對象，而不會修改原來的對象。

可變對象：如列表（list）、字典（dict）、集合（set）等。當對這些對象進行修改時（如添加元素），變數仍然引用原來的對象，只是對象的內容發生了變化。-- > 代表可以修改對象內容！！！

疑問：那既然變數名存儲的是引用，而不是直接存儲數值，那為何作用在變數名上的操作可以直接改變存儲在該地址上的數值呢？

Ans: Python中變數名存儲的是引用，而不是直接存儲數值，然而，當我們對變數名進行操作時，會發現可以「直接改變」變數指向的內容。這涉及到Python的變數操作機制，特別是可變對象和不可變對象的區別。

不可變對象（如整數、字符串、元組）：一旦創建，它們的內容無法被改變。如果試圖改變它的值，Python會創建一個新的對象，並讓變數引用新對象。

可變對象（如列表、字典、集合）：可以直接修改它們的內容，而不用創建新的對象。這是因為可變對象支持「就地修改」，所以變數引用的仍然是同一個對象，只是其內容發生了變化。

總結：

變數名存儲的是對象的引用，當你對變數進行操作時，Python根據該對象的類型決定如何處理。

對可變對象，操作直接改變內存中對象的內容，變數名仍然指向原來的內存地址。

對不可變對象，操作會創建一個新的對象，變數名指向新的內存地址，而原來的對象保持不變。

理解這一點對掌握Python的變數行為至關重要，有助於避免無意間的數據修改或內存混淆。

承上，上述觀念也會和shallow copy 和deep copy 相關

<https://chatgpt.com/share/ef18976d-c70e-478a-bee4-1ec7dc7b2341>

2.1.2.1 Naming variables

規則： 記得不可以用數字起首

語法：function "format".

Printing statements that include both a string and another variable

# A string in Python is always denoted using "".

x = 3

y = 5

sum\_xy = x+y

print("The sum of x and y is {}".format(sum\_xy)) 注意是雙引號後面才接.format()

Printing statements that include both a string and several variables （一次性打印多個數值）

x = 3

y = 5

diff\_xy = x-y

prod\_xy = x\*y

ratio\_xy = x/y

print("The difference of x and y is {}. The product of x and y is {}. Finally, the ratio

of x and y is {}".format(diff\_xy, prod\_xy, ratio\_xy))

2.1.2.4 Comparing variables

Using Python, you can compare two variables of the same type to each other.

2.1.3.1 Strings

String in python always denoted using “” 注意是雙引號