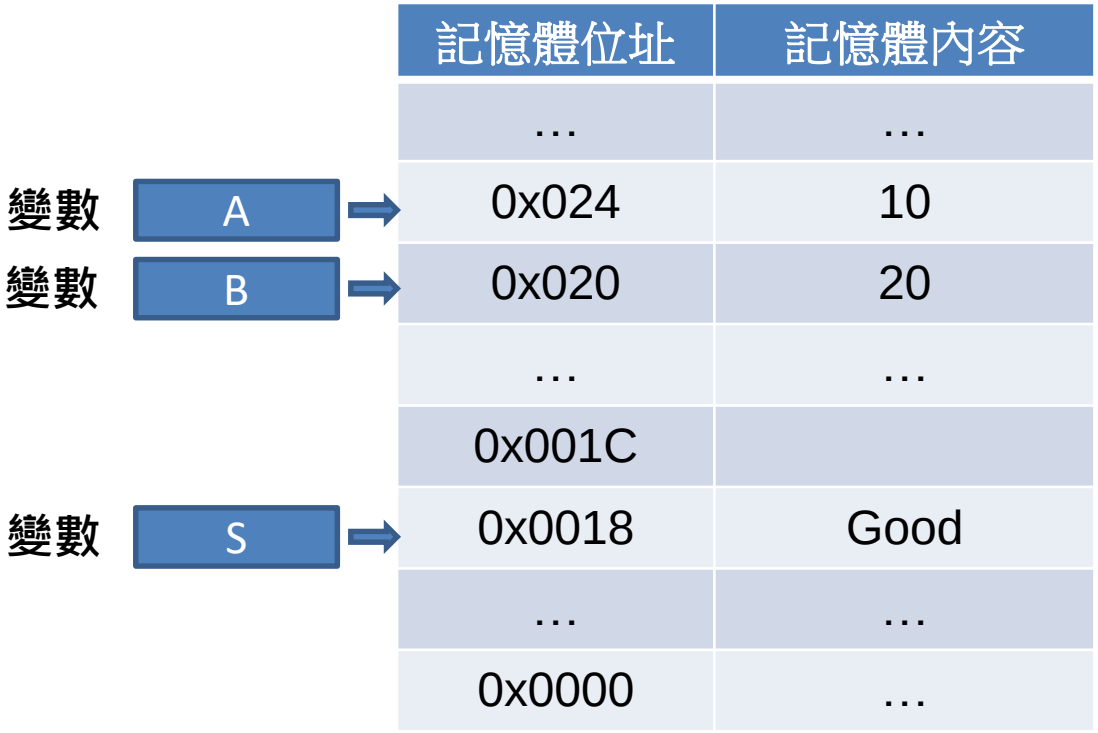
Python_04

變數、型別、運算子

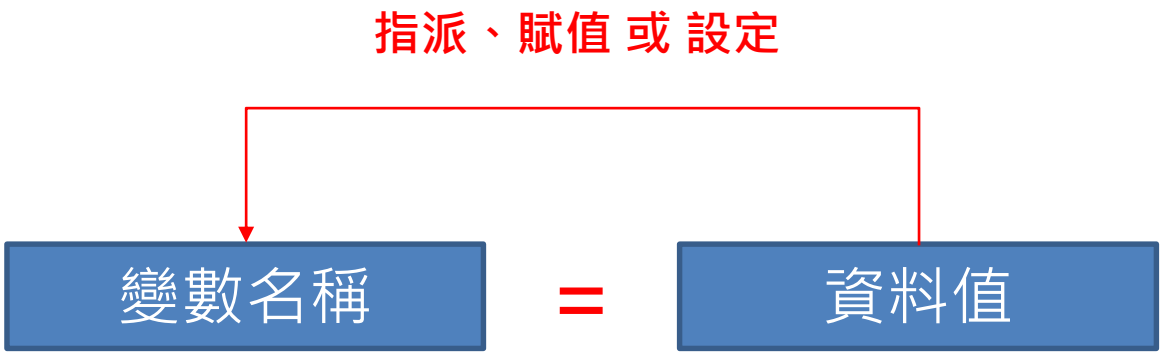
變數 (Variable)

- 變數是識別字，讓程式透過自訂名稱，提取記憶體中的資料(例如：數值、文字)來進行運算。
- 簡單說，變數代表記憶體中的某一筆資料。



變數賦值(Assignment)

- 等號 (=) 是 賦值運算 ，把資料值指派給某一個變數



A = 10
B = 0.12
S = ‘Good’

表達式 (Expression)

加法

2 + 5

2 * 5

結合

'Hello' + 'Python'

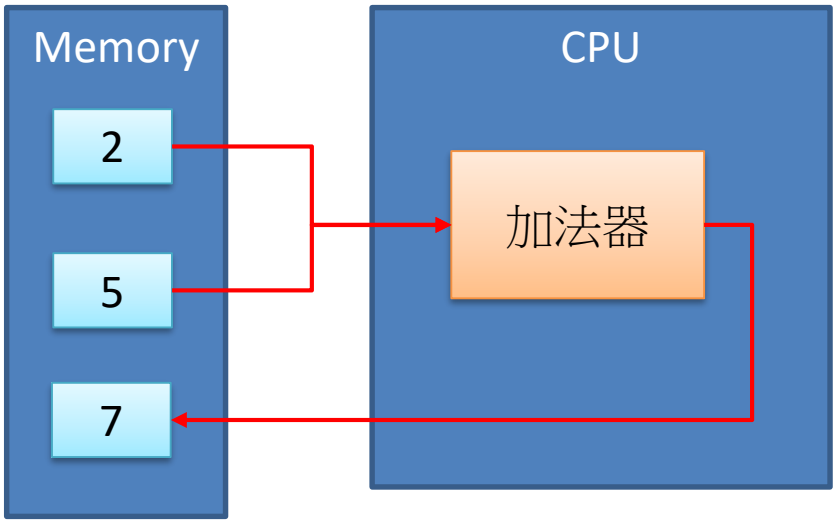
Expression

2 + 5

運算元
Operand

運算子
Operator

運算元
Operand



表達式結合變數賦值

A = 2 + 5

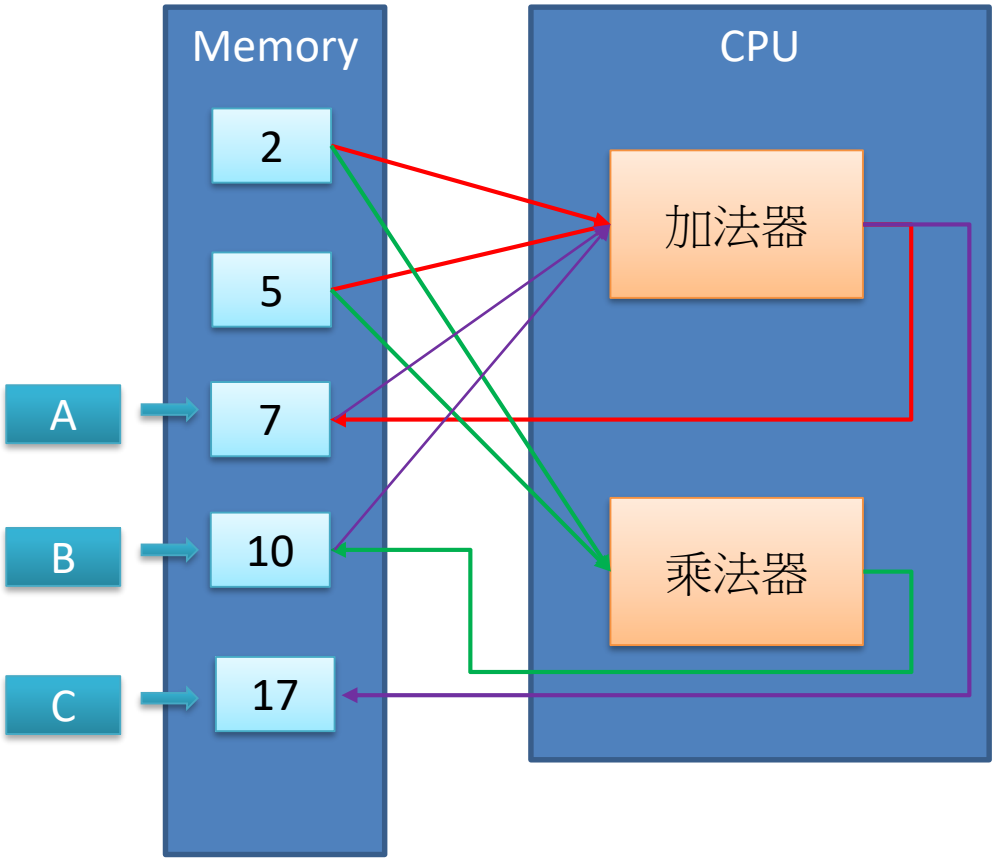
B = 2 * 5

C = A + B

A = ‘Hello’ + ‘Python’

A = B = C = 7

A , B , C = 1 , 2 , “Hello”



動態型別 (Dynamic Type)

- 動態型別語言，建立變數不用宣告型別
  - Python是動態型別，變數在賦值之後，型別才被建立。
- 靜態型別語言，建立變數要先宣告型別，
  - C 語言建立變數要宣告變數型別 (例如：int a ;)

要符合識別字命名規則　　由值決定變數的資料型態(Type)

變數名稱 = 值

A	=	10	整數型別
A	=	0.01	浮點數型別
A	=	“Hello”	字串型別

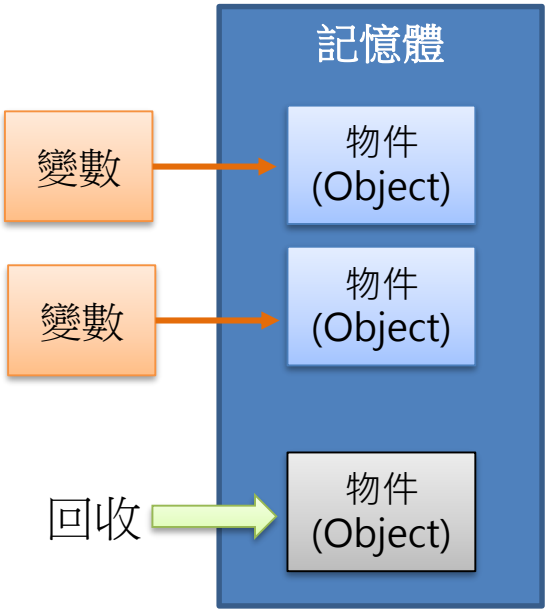
變數與物件參照

- Python的變數，如同一張便利貼(標籤紙)登記著變數名稱，這張便利貼(標籤紙)會指著一個對象，也就是變數代表某個物件。
- Python變數只是識別字，本身沒有型別，型別是所指物件的型別。

A = 10 為例

Python一切皆是物件(Object)

- **Python一切皆是物件(Object)**，每個object有專屬的idendity，type和value。
  - identity：簡稱 id，物件的記憶體位址，物件被創建後，其id將不可改變。
  - type：物件的型別(資料型態)，物件被創建後，其類型不可改變。
  - value：物件的值(資料內容)，和identity與type不同，物件的值有些是可變(mutable)，有些或不可變 (immutable)。
- Python物件不再使用時(沒有變數參照)， Python直譯器會自動垃圾收集(Garbage Collection,GC)，釋放記憶體空間。



變數刪除

- 使用 del 刪除變數

```
A = 1
B = 10
C = 100
del A
del B, C
```

使用type()、id()、print()觀察物件

- Type

  建立變數 A = 10

  命令列互動模式輸入type(A) 或 print(type(A))

  會輸出A的型態為 <class 'int'>

- Id
  - 命令列互動模式輸入id(A) 或 print(id(A))
- value
  - 命令列互動模式輸入 A 或 print(A)

Python3 標準型別

- Python3 標準型別：
  - 數值型別 (Numeric Type)
    - int (整數)：1 , 12, 100
    - bool (布林)：True 、False，True為1，False為0
    - float (浮點數)：1.23、1.8e-2 (即 1.8 x 10^{-2}) 。
    - complex (複數)：1 + 2j
  - 序列型別(Sequence Type)
    - String（字串）
    - Tuple（值組）
    - List（串列）
  - 雜湊型別(Hashable Type)
    - Set（集合）
    - Dictionary（字典）
  - 不可變 (immutable) ： Numeric （數值）、String（字串）、Tuple（值組）
  - 可變(mutable) ： List（串列）、Dictionary（字典）、Set（集合）

不可變資料形別

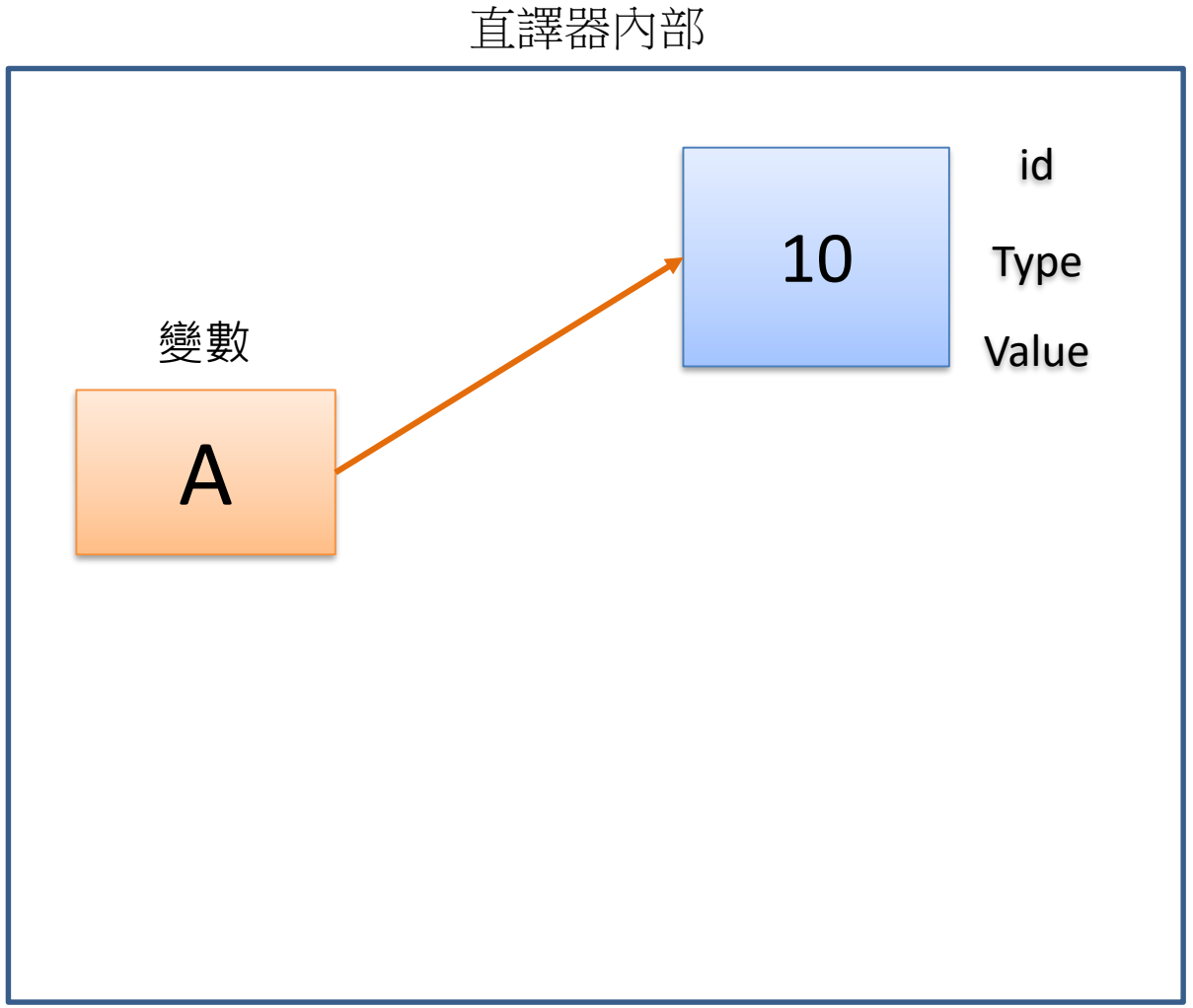
1

A = 10

A = 0.01

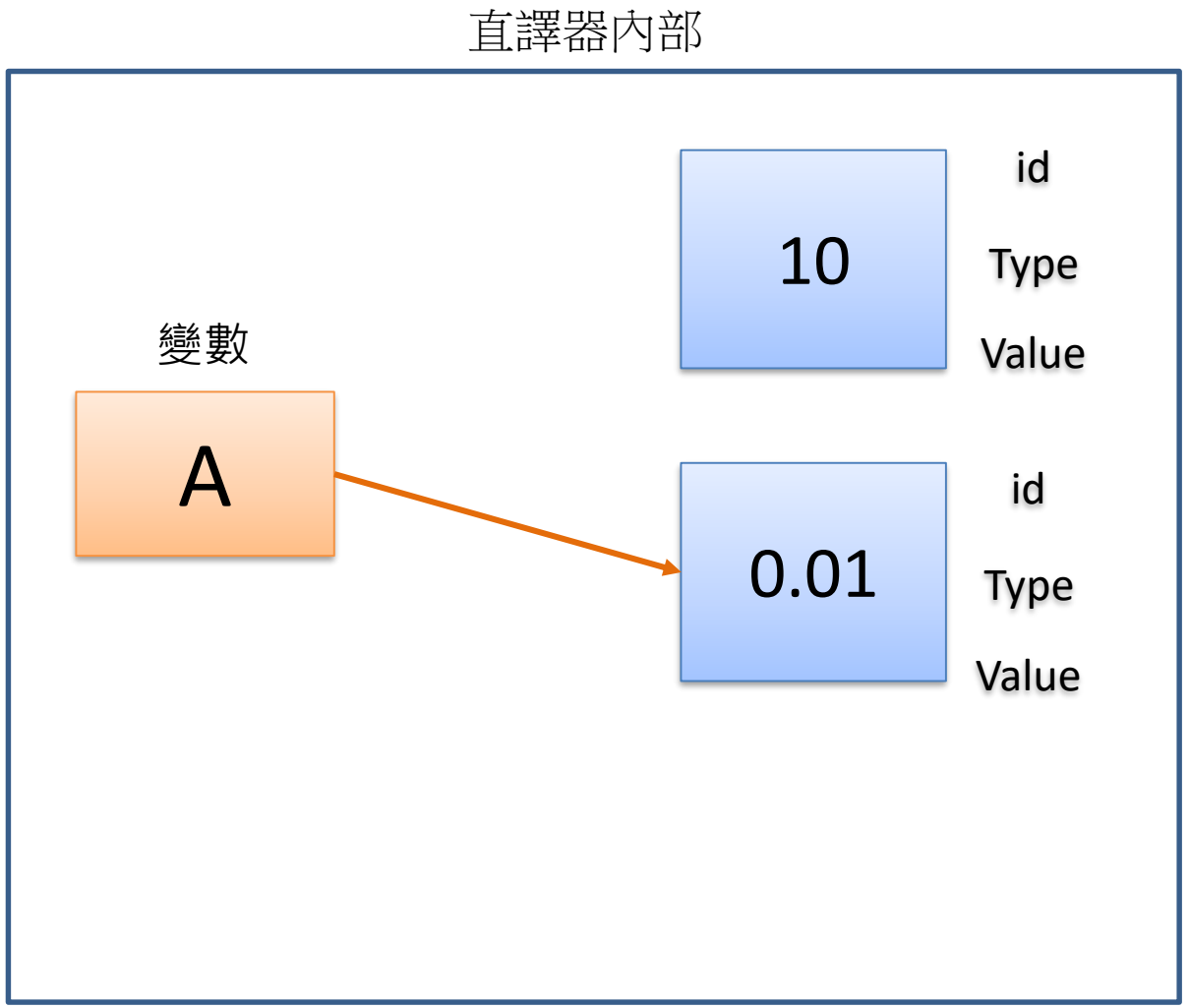
B = 0.01

A = “Hello”



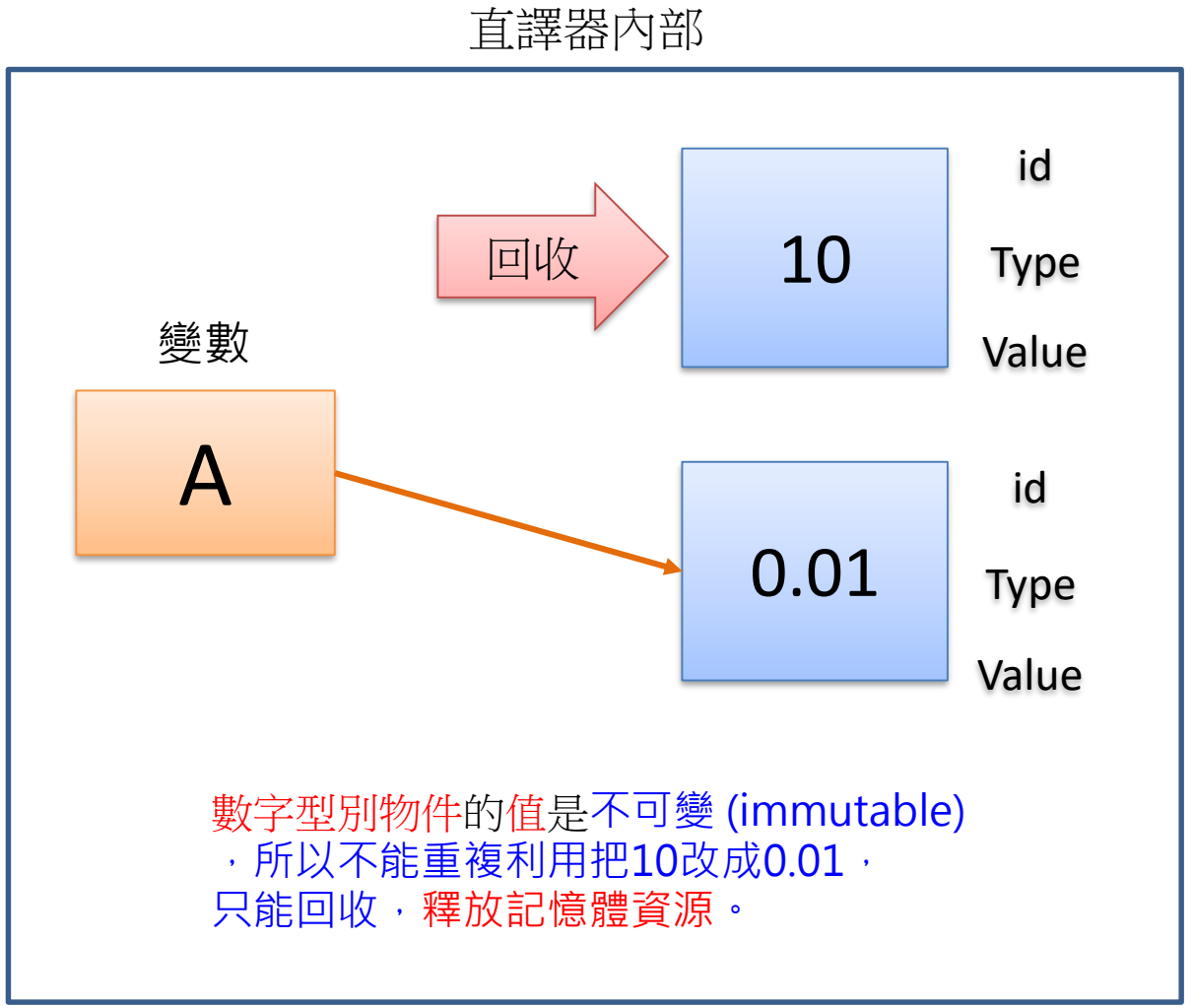
不可變資料形別

1. A = 10
2. A = 0.01
   B = 0.01
   A = “Hello”



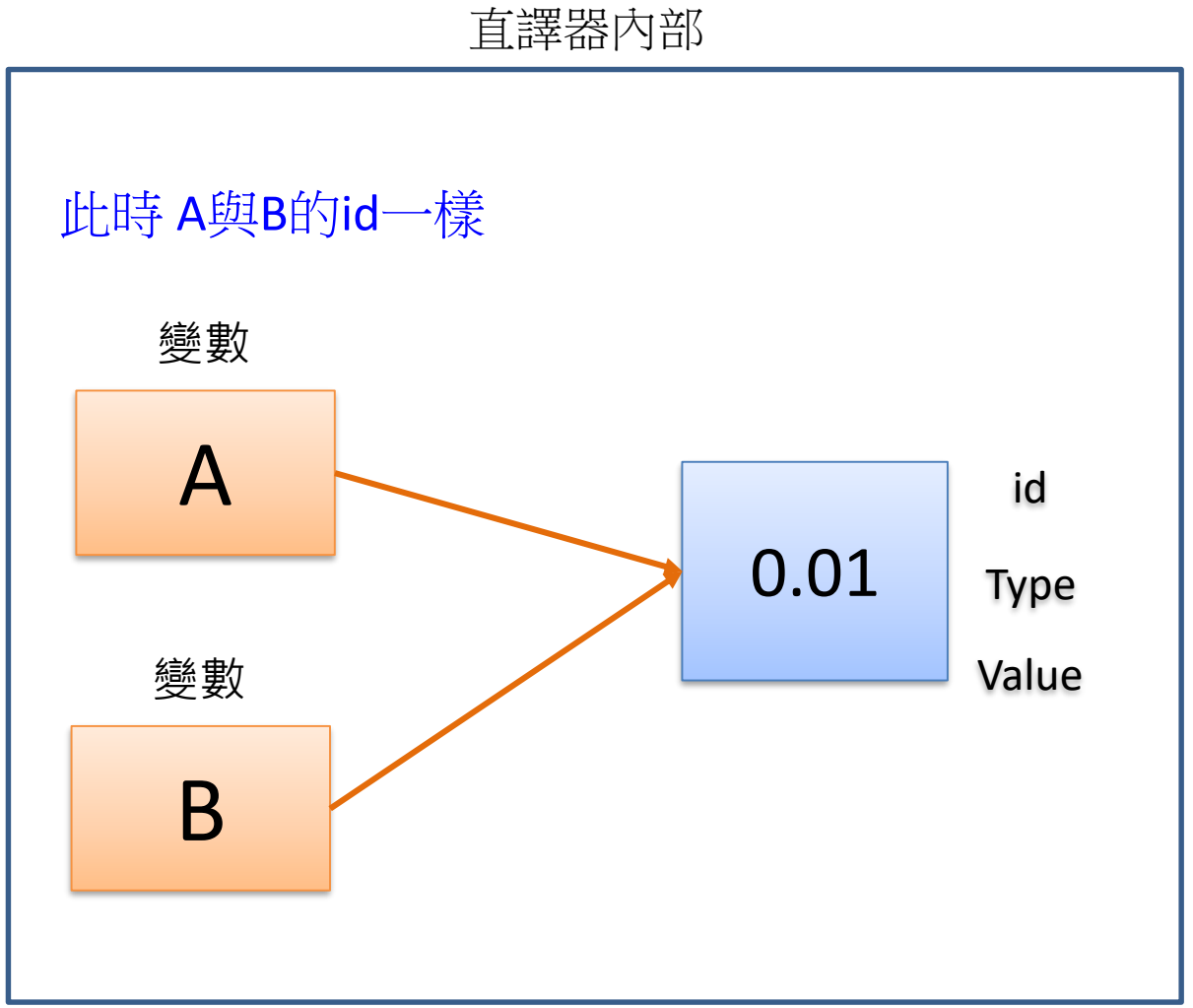
不可變資料形別

1. A = 10
2. A = 0.01
   B = 0.01
   A = “Hello”



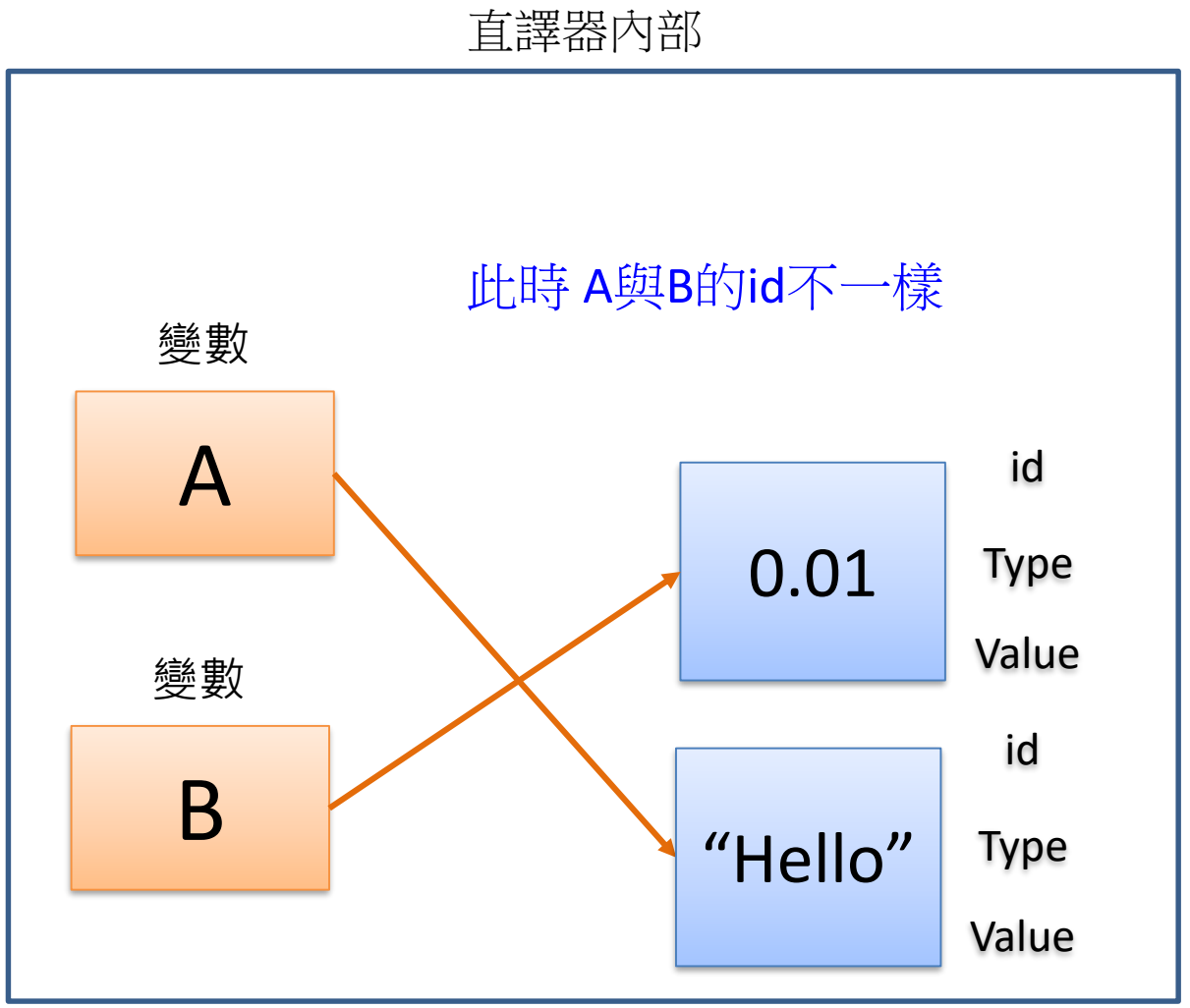
不可變資料形別

1. A = 10
2. A = 0.01
3. B = 0.01
4. A = “Hello”



不可變資料形別

1. A = 10
2. A = 0.01
3. B = 0.01
4. A = “Hello”



隋棠演練

觀察 A 所指的物件 id 變化

```
A =  10
print ( id ( A ) )
A = A + 3
print ( id ( A ) )
```

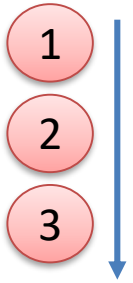
可變資料形別

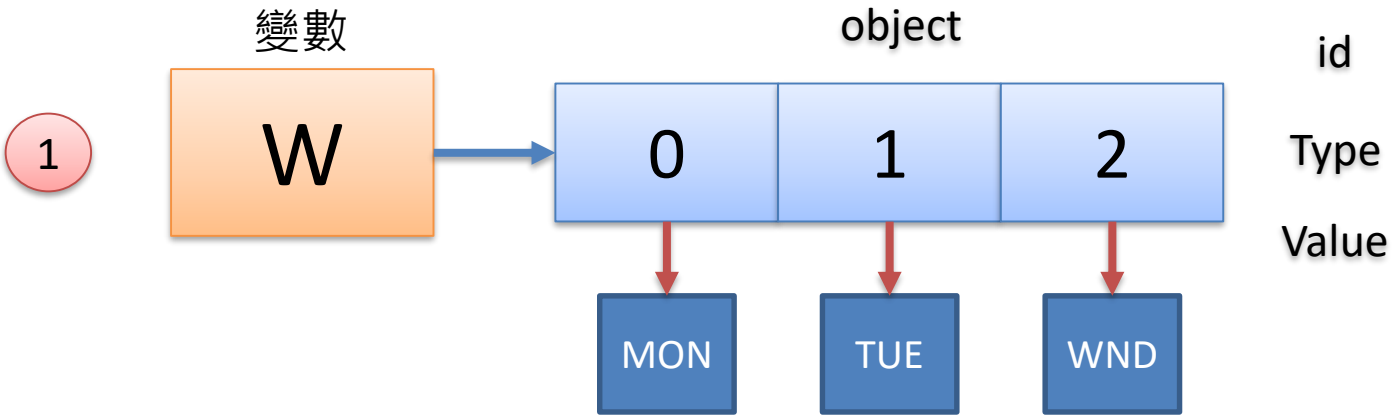
- 以List(串列) W為例：

  W = [‘MON’ , ’TUE’ , ’WND’]

  W[2] = [‘WED’]

  W+= [‘THU’]





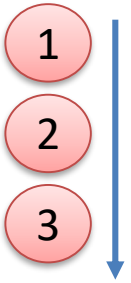
可變資料形別

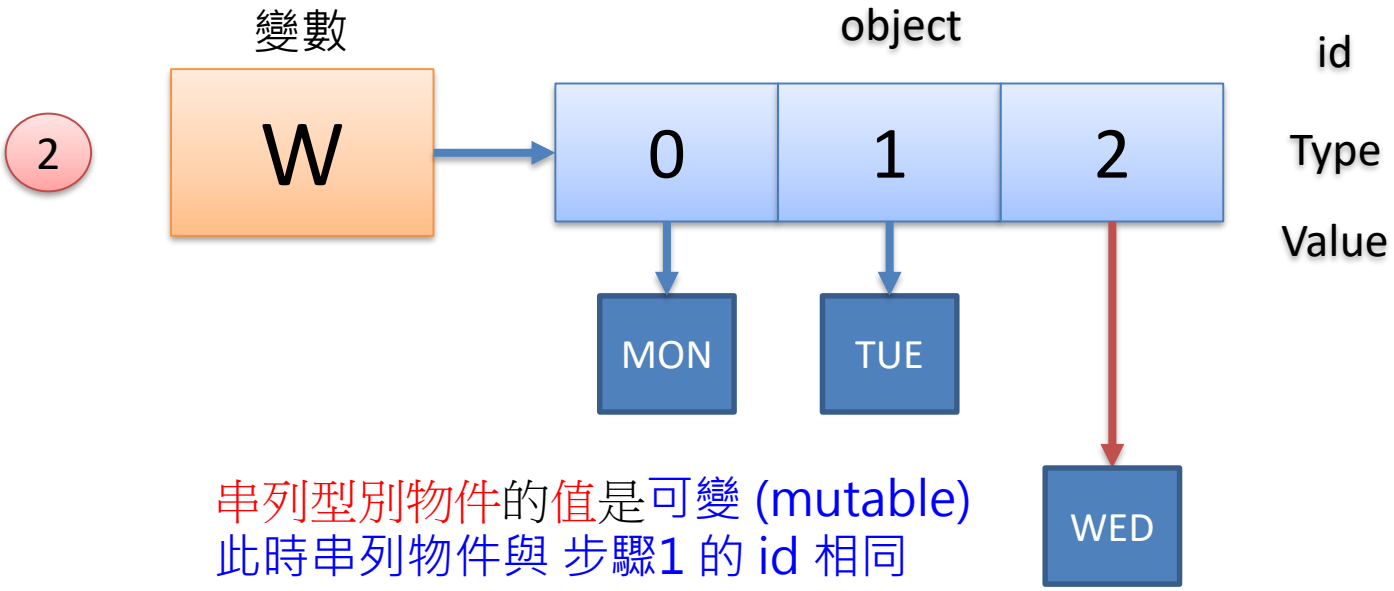
- 以串列 W為例：

  W = [‘MON’ , ’TUE’ , ’WND’]

  W[2] = [‘WED’]

  W+= [‘THU’]





串列型別物件的值是可變 (mutable)
此時串列物件與 步驟1 的 id 相同

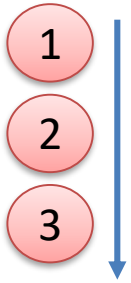
可變資料形別

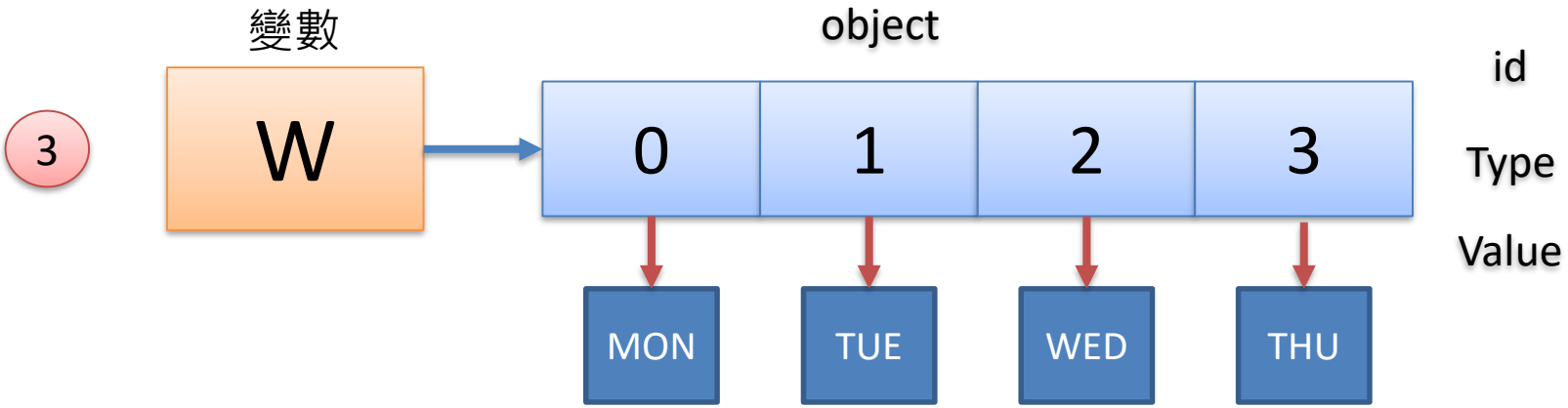
- 以串列 W為例：

  W = [‘MON’ , ’TUE’ , ’WND’]

  W [2] = [‘WED’]

  W += [‘THU’]





串列型別物件的值是可變 (mutable)
此時串列物件與 步驟1 的 id 相同

隨堂演練

- 觀察 L 與 M 所指向的物件 id 與值的變化

```
L = [10, 20, 30 ]
print ( id ( L ) )
L[0] = 11
print ( L )
M = L
print ( id ( M ) )
M[1] = 21
print ( M)
print ( L )
print( id ( L[0] ) )
print( id ( L[1] ) )
print( id ( M[1]) )
print( type(L) )
print( type(L[0]) )
```

Bool 布林型別

- 一個布林值只有True 、False 二種值，要麼是True，要麼是False,在Python中，可以直接用True，False表示布林值（請注意大小寫）
- 3 > 2為 True
- 3 > 5為 False
- type(3 > 5) 可得到 <class 'bool'>

數值型別

- Python3支持 bool (布林) int、float、complex (複數)。
  - bool (布林)，如：True 、False，True為 1，False為 0 。
  - int (整數)，如：100 只有一種整數類型int，表示為長整數，Python3沒有Python2中的Long。
  - float (浮點數)，如：1.23、1.8e-2 (即 1.8×10^{-2}) 。
  - complex (複數)，如：1 + 2j
    - 變數.real可取得實部(real)，變數.imag可取得虛部 (imaginary)。

- 混合計算時，Python會把整數轉換成為浮點數。

Python3 數值表示法

int	float	complex
10	0.0	3.14j
100	15.20	45.j
-786	-21.9	9.322e-36j
0b0101	32.3e+18	.876j
-0490 (錯誤)	-90.	-.6545+0J
-0x260	-32.54e100	3e+26J
0x69	70.2E-12	4.53e-7j

練習

- 觀察 a = 1+2j ; b=3+4j 的四則運算
- 觀察 a.real a.imag
- 觀察 a= 1 ; b=2. ; a+b 的型別

- 複數的加減乘除：設 a,b,c,d 為實數

  加法：(a+bj)+(c+dj)=(a+c)+(b+d)j

  減法：(a+bj)−(c+dj)=(a−c)+(b−d)j

  乘法：(a+bj)*(c+dj)=(ac−bd)+(bc+ad)j

  除法： (a+bj)/(c+dj) =

  $(ac+bd)/(c^2+d^2) + ((bc-ad)/(c^2+d^2))j$

序列型別物件的通用方法

運算	結果
x in s	如果s中的某項等於x則結果為True，否則為False
x not in s	如果s中的某項等於x則結果為False，否則為True
s + t	s與t相拼接
s * n 或 n * s	相當於s與自身進行n次拼接
s[i]	s 的第i項，起始為0
s[i:j]	s 從 i 到 j 的切片
s[i:j:k]	s 從 i 到 j 步長為 k 的切片
len(s)	s 的長度
min(s)	s 的最小項
max(s)	s 的最大項
s.index(x)	x在s中首次出現項的索引號
s.count(x)	x 在s中出現的總次數

字串(String)

- Python 字串型別(String)是一種序列型別 ，由有序的字元組成的，可用索引位置可以存取字串中的字元。
  - 短字串： (用一對單引號或一對雙引號括起來)
  - 長字串：(用一對三個單引號或三個雙引號括起來)
  - 反斜線(\)： 用來對特殊含義的字元進行轉義 (Escape Sequence)
  - 加號(+)：用來串接字串
  - 星號(*)：用來重複字串
  - [:] 用來切片

s = ‘abcdef’

字串s	a	b	c	d	e	f
由前索引	0	1	2	3	4	5
由後索引	-6	-5	-4	-3	-2	-1

練習

- 練習字串的通用操作

```
s = 'abcdabcdef‘
s[0:2]
s[:]
s[:4]
s[:-1]
s[-1]
s.index(‘a’)
s.count(‘b’)
len(s)
 ‘a’ in s
```

串列 (list)

- 串列(list)，使用中括號[ ]，是一種序列型別
- 串列中的元素可以是不同資料型別
- 串列中的元素可以是串列
- 串列可以是巢狀(多維)的串列

L = [“Alex”, “Kevin”, “Martin”, ”Rudy”, ”Zoe”]

串列L	“Alex”	“Kevin”	“Martin”	”Rudy”	”Zoe”
由前索引	0	1	2	3	4
由後索引	-5	-4	-3	-2	-1

練習

- 任意物件的串列
  a = [2, 3, 4]
  b =[6 , 7 , 8.5 , “XYZ”]
  c = [ ]
  d = [2, [a, b]]
  e = a + b

- 串列的基本操作
  取出元素：x = a[1]
  取出子串列：y = b[1:3]
  多維度取出元素：z = d[1][0][2]
  設定元素：b[0] = 4

值組(tuple)

- 值組tuple使用是小括號(),使用索引位置可以存取元素
- 可以收集不同資料型態的元素
- tuple中的元素允許是tuple
- tuple的元素是有序的(誰前誰後有關係)
- 與list不同的，tuple是一種唯讀且不可 變更的資料結構，不可取代tuple中的任意一個元素，因為它是唯讀不可變更。

T = (10, 20, 30, 40, 50, 60)

值組T	10	20	30	40	50	60
由前索引	0	1	2	3	4	5
由後索引	-6	-5	-4	-3	-2	-1

隨堂演練

- tuple

  f = (2,3,4,5)

  g = ()

  h = (2, [3,4], (10,11,12))

- tuple的操作

  x = f[1]

  y = f[1:3]

  z = h[1][1]

雜湊型別 (Hashable Type)

- Set（集合）
- Dictionary（字典）

集合 (set)

- 集合(set)的常見的用途包括成員檢測、從序列中去除重複項以及數學中的集合類計算，例如交集、並集、差集等。
- 創建一個空集合必須用**set()**而不是{ }，因為**{ }**是用來創建一個空字典。
- set 的元素是無序的(誰前誰後沒關係)

```
student = {'Tom', 'Jim', 'Mary', 'Tom', 'Jack', 'Rose'}
```

隨堂演練

- 集合的操作

```
S1 = {1,2,3,1,1,2,3,4,5,1}
S2 = {5,3,2,1,2,1,4}

print(S1)
print(S2)
```

字典 (dict)

- 建立字典變數可利用大括弧{ }，裡頭以 key : value 為配對的資料項目(item)或稱為元素(element)，若有多筆資料再 以逗號區隔開。
- 與list, tuple不同，不以索引位置存取元素，以key當作索引存取元素
- 使用字典須注意， key必須是不可變的 (immutable) 資料型態，如數字、字串 (string) 等，value則沒有限制
- dict 的元素是無序的

D = { "apple" : 200 , "Orange" : 300, "Banana" : 250 }

隨堂演練

- 字典的操作

```
D = {}
D[‘G’] = “google”
D[‘Y’] = “youtube”
D[‘F’] = “facebook”
print(D[‘G’])
```

型別轉換(Type Casting)

- 處理資料的時候，有些資料型別不符合程式需求的型別，可進行型別轉換，Python為強制資料型別轉換(Explicit Type Conversion)
  - 將 x 轉換為整數：int(x)
  - 將 x 轉換為浮點數：float(x)
  - 將 x 轉換為字串：str(x)
  - 將 x 轉換為布林型別：bool(x)
- 浮點數強制轉換成整數，不會四捨五入，會捨棄小數部份
- 型別轉換會在記憶體空間配置新的物件，內容為轉換後的結果。

Python型別轉換的內建函數

函式	描述
int(x [,base])	將x轉換為一個整數
float(x)	將x轉換到一個浮點數
complex(real [,imag])	創建一個複數
str(x)	將對象x 轉換為字串
repr(x)	將對象x 轉換為表達式字串
eval(str)	用來檢查在字串中的有效Python表達式,並返回一個對象
tuple(s)	將序列s 轉換為一個元組
list(s)	將序列s 轉換為一個列表
set(s)	轉換為可變集合
dict(d)	創建一個字典。d 必須是一個(key, value)元組序列。
frozenset(s)	轉換為不可變集合
chr(x)	將一個整數轉換為一個字元
ord(x)	將一個字元轉換為它的整數值
hex(x)	將一個整數轉換為一個十六進製字串
oct(x)	將一個整數轉換為一個八進製字串

隨堂演練

- X = 100，將 X 轉成字串

```
X =  100
A = str(X)
print(A)
```

Python 運算子(Operators)

- 運算子分類
  - 算術運算子
  - 比較運算子
  - 賦值運算子
  - 邏輯運算子
  - 位元運算子
  - 成員運算子
  - 身份運算子

算術運算子

假設 a=10 b=21

運算子	描述	實例
+	加，兩個變數相加	a + b 輸出結果31
-	減， 得到負數或是一個變數減去另一個變數	a - b 輸出結果-11
*	乘，兩個變數相乘或是回傳一個被重複若干次的字串	a * b 輸出結果210
/	除，x 除以 y	b / a 輸出結果2.1
%	餘數，回傳除法的餘數	b % a 輸出結果 1
**	次方，回傳x的y次方	a**b 為10的21次方
//	取整除，向下取接近除數的整數	9 // 2 輸出結果 4 - 9 //2 輸出結果- 5

比較運算子

假設 a=10 b=20

運算子	描述	實例
==	等於- 比較對像是否相等	(a == b) 回傳 False。
!=	不等於- 比較兩個對像是否不相等	(a != b)回傳True。
>	大於- 回傳x是否大於y	(a > b) 回傳False。
<	小於- 回傳x是否小於y。所有比較運算子回傳1表示真，回傳0表示假。這等同特殊的變數True和False(注意大寫)。	(a < b) 回傳 True。
>=	大於等於- 回傳x是否大於等於y。	(a >= b) 回傳False。
<=	小於等於- 回傳x是否小於等於y。	(a <= b) 回傳 True。

賦值運算子

運算子	描述	實例
=	簡單的賦值運算子	c = a + b 將a + b 的運算結果賦值為 c
+=	加法賦值運算子	c += a 等同於 c = c + a
-=	減法賦值運算子	c -= a 等同於 c = c - a
*=	乘法賦值運算子	c *= a 等同於 c = c * a
/=	除法賦值運算子	c /= a 等同於 c = c / a
%=	餘數賦值運算子	c %= a 等效於 c = c % a
**=	次方賦值運算子	c **= a 等同於 c = c ** a
//=	取整除賦值運算子	c //= a 等同於 c = c // a

邏輯運算子

假設 a=10 b=20

運算子	邏輯表達式	描述	實例
and	x and y	“與” 如果x 為False，x and y 回傳False，否則它回傳y 的計算值。	(a and b) 回傳20。
or	x or y	“或” 如果x 是True，它回傳x 的值，否則它回傳y 的計算值。	(a or b) 回傳10。
not	not x	“非” 如果x 為True，回傳False 。如果x 為False，它回傳True。	not(a and b) 回傳False

位元運算子

假設 a = 60，b =13

運算子	描述	實例
&	位元AND運算子：參與運算的兩個值,如果兩個相應位都為1,則該位的結果為1,否則為0	(a & b) 輸出結果12 ，二進制解釋： 0000 1100
|	位元OR運算子：只要對應的二個二進位有一個為1時，結果位就為1。	(a | b) 輸出結果61 ，二進制解釋： 0011 1101
^	位元XOR運算子：當兩對應的二進位相異時，結果為1	(a ^ b) 輸出結果49 ，二進制解釋： 0011 0001
~	位元NOT運算子：對數據的每個二進制位取反,即把1變為0,把0變為1。~x類似於-x-1	(~a) 輸出結果-61 ，二進制解釋： 1100 0011， 在一個有符號二進制數的補碼形式。
<<	左移動運算子：運算數的各二進位全部左移若干位，由"<<"右邊的數指定移動的位數，高位丟棄，低位補0。	a << 2 輸出結果240 ，二進制解釋： 1111 0000
>>	右移動運算子：把">>"左邊的運算數的各二進位全部右移若干位，">>"右邊的數指定移動的位數	a >> 2 輸出結果15 ，二進制解釋： 0000 1111

成員運算子

運算子	描述	實例
in	如果在指定的序列中找到值回傳True，否則回傳False。	x 在y 序列中, 如果x 在y 序列中回傳True。
not in	如果在指定的序列中沒有找到值回傳True，否則回傳False。	x 不在y 序列中, 如果x 不在y 序列中回傳True。

身份運算子

運算子	描述	實例
is	is 是判斷兩個識別字是不是引用自一個對象	x is y ,類似id(x) == id(y) ,如果引用的是同一個對象則返回True，否則返回False
is not	is not 是判斷兩個識別字是不是引用自不同對象	x is not y，類似id(a) != id(b)。如果引用的不是同一個對象則返回結果True，否則返回False。

運算子優先順序

高

運算子	描述
**	指數
~ + -	位元反轉, 正負號
* / % //	乘，除，餘數和取整除
+ -	加法 減法
>> <<	右移，左移運算子
&	位元'AND'
^ |	位元運算子
<= < > >=	比較運算子
<> == !=	等於運算子
= %= /= //= -= += *= **=	賦值運算子
is is not	身份運算子
in not in	成員運算子
not and or	邏輯運算子

低

單元練習

1. B = 4.56，請問變數B的type、id、value?
2. 「010, 華僑銀行, 011, 上海銀行, 012, 台北富邦, 013, 國泰世華, 016, 高雄銀行 」，請將前述資料使 Python字典型態表示。
3. 請將 ’ ABC ’ 重複5次輸出在同一列?
4. S=‘abcdefghijklm’ 請印出字串切片 ‘ijklm’
5. A = 5，請問 A+=5, A*=5 分別為多少?
6. 請問 Python 不相等的比較運算子?
7. 請問 and 與 & 有什麼不同?