Python

變數宣告與賦值

變數/variable

- 變數

  數值會改變的數，稱為變數（variable），例如：溫度

  而恆常不會變動的數叫做常數（constant），例如：圓周率π（3.14159265359…）

  「變數」也可以看成是一種包含不同數值的數

- 變數與程式設計

  在程式設計的過程中，變數的值需要能夠隨時存取或重新得到得到新值(賦值)

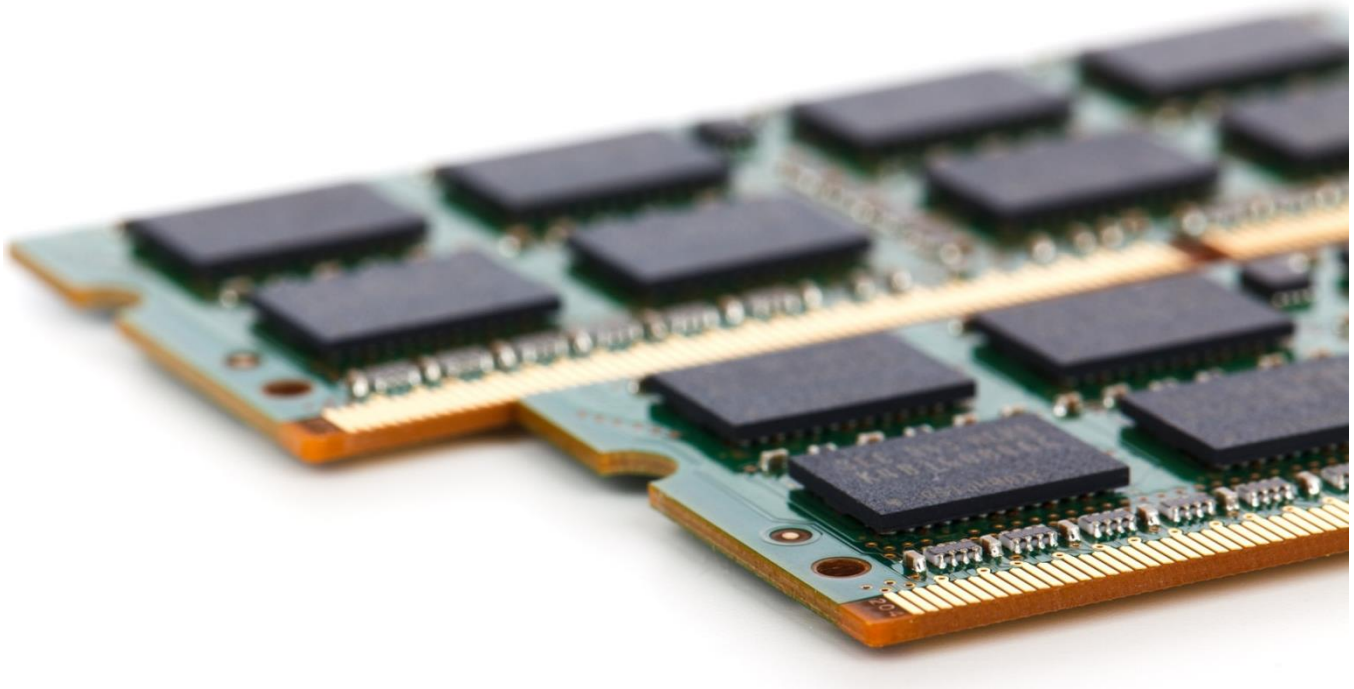
  因此，變數必須存放在電腦的記憶體，以便隨時存取

星期	一	二	三	四	五
溫度	26	27	22	18	10

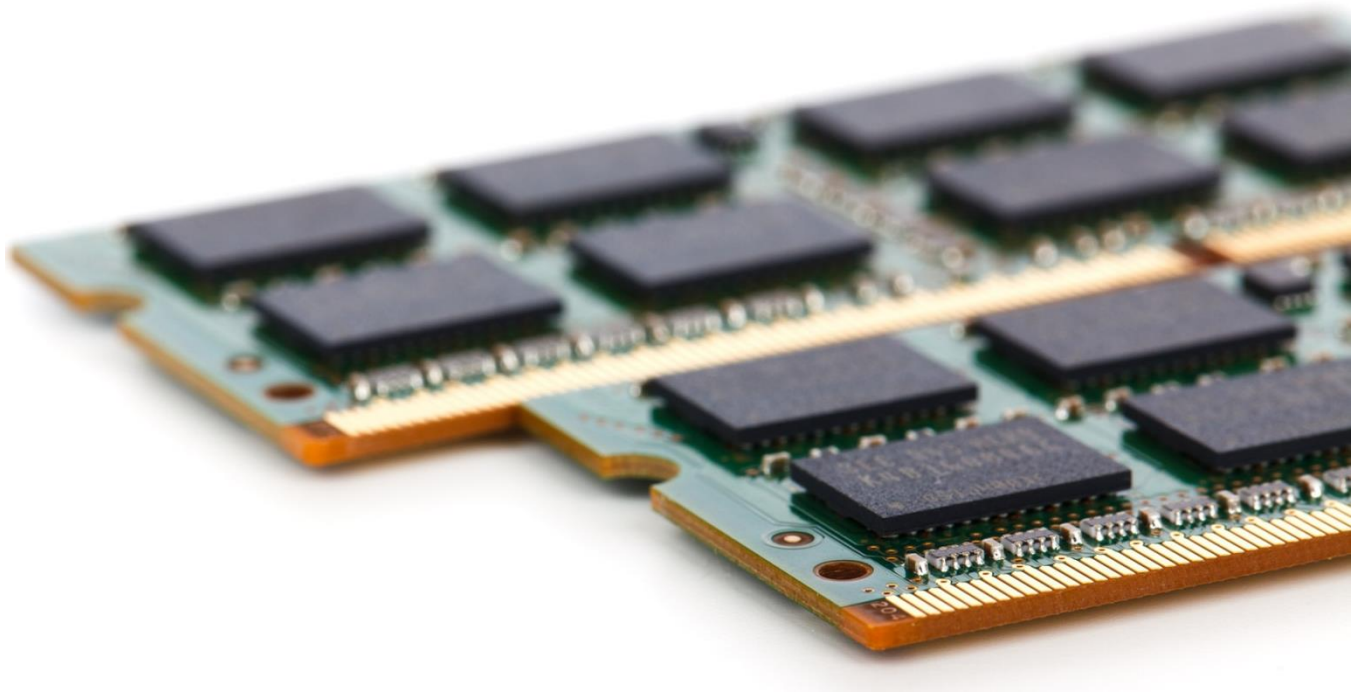
每天的溫度是變數

寫一個程式，
指定一個溫度(變數)為3，
並且印出來。

變數的值(資料)
放在電腦的記憶體，記憶體猶如資料的家



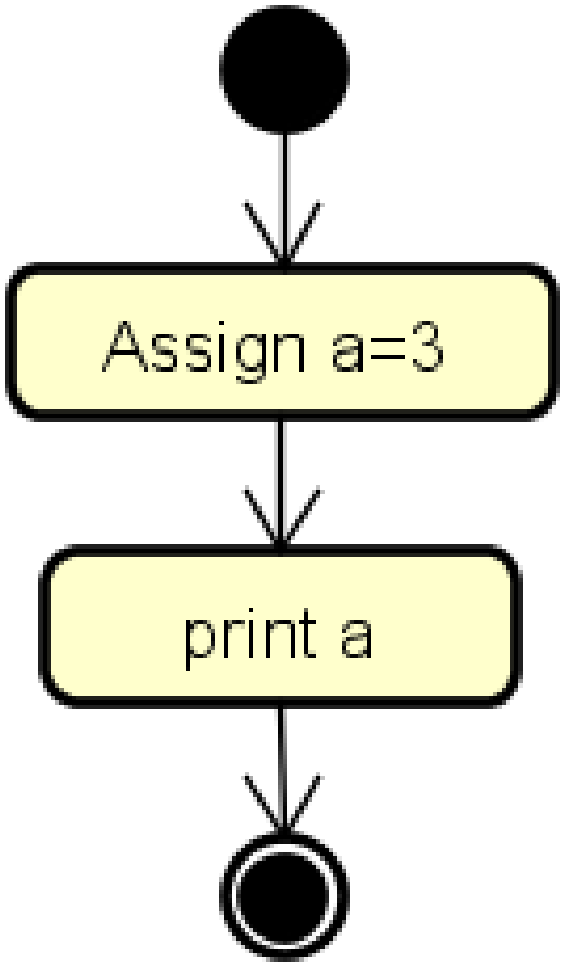
變數進駐記憶體，如何圈地？



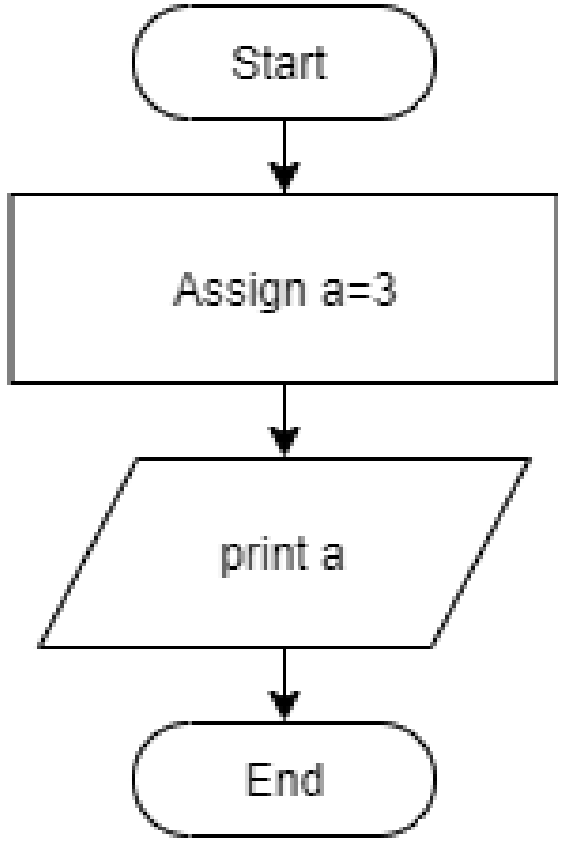
想像為有編號的櫃子

位置	存放的資料
第一個櫃子	3

流程



UML活動圖

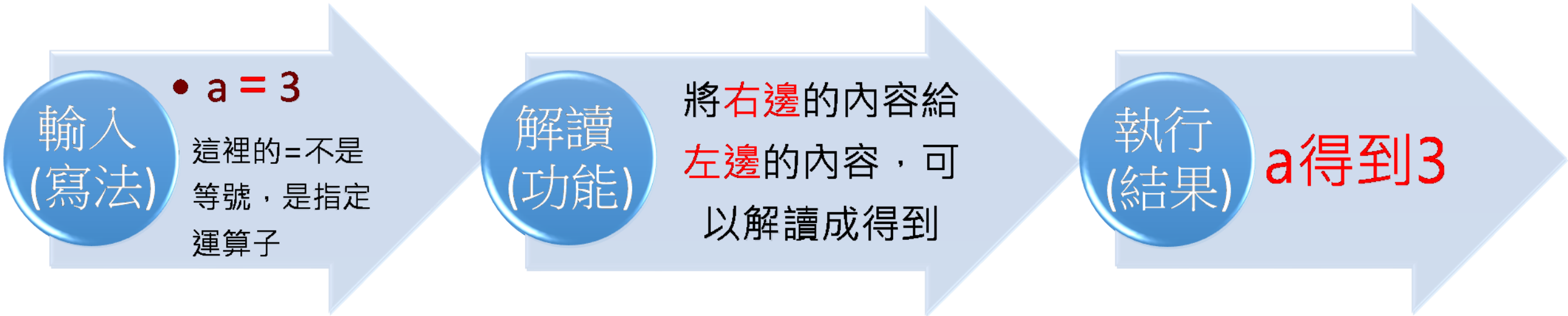


流程圖

程式碼

```
a = 3  # 註解1 ,註解2
print(a) # 註解3
```

註解1：=的用法


輸入
(寫法)
• a = 3
這裡的=不是等號，是指定運算子
解讀
(功能)
將右邊的內容給左邊的內容，可以解讀成得到
執行
(結果)
a得到3

註解2：a=3

- 程式碼意義
  - 代表向電腦申請 一個櫃子 準備用來儲放資料
  - 櫃子的位置需要設定名字

- a 是櫃子位置的名字
  - a 稱為 變數名

- 3 是資料內容
  - 3 是櫃子內存放的資料，也可以說
  - 3 是變數的內容值

櫃子名稱	存放的資料
a	3

如何給變數取名

- 原則上隨便取
  - 可以是單獨的字母：a、b、c
  - 可以是多個字母的結合：aaa、bbb、ccc
  - 也可以是混數字的：number1、number2

- 但下面幾個一定要注意
  - 變數開頭要是字母或底線(_)，不能是數字！
  - 大小寫不一樣就是不同變數喔！
    - aaa ≠ AAA
  - 名字中間不能夾有空白
  - 有預設功能的字不能用，如：if, for, else, …
    - 可參考: https://www.w3schools.com/python/python_ref_keywords.asp

取名不麻煩，

A=3

a1=3

temp=3

temperature =3

怎麼取名字比較好？

- 有意義的比較好
  變數名稱如果命名的好，
  一眼就看出其中的**含意**，以及變數名稱的**種類**，
  整個架構也會比較清楚。

下方的命名，比上方好

A=3

a1=3

temp=3

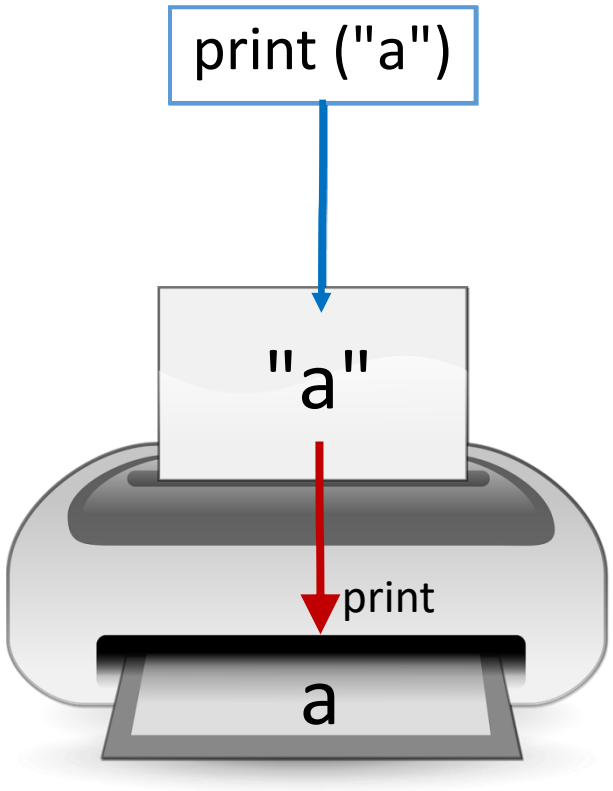
temperature =3

資料內容的類型(資料型態)

- a=3
- 3 是資料內容
- 資料可以分成不同類型，有整數、小數，還有很多種
  - python 的輸入不需要管資料型態，不需指定型態，變數存放的型態也可隨時根據儲存的值進行修改

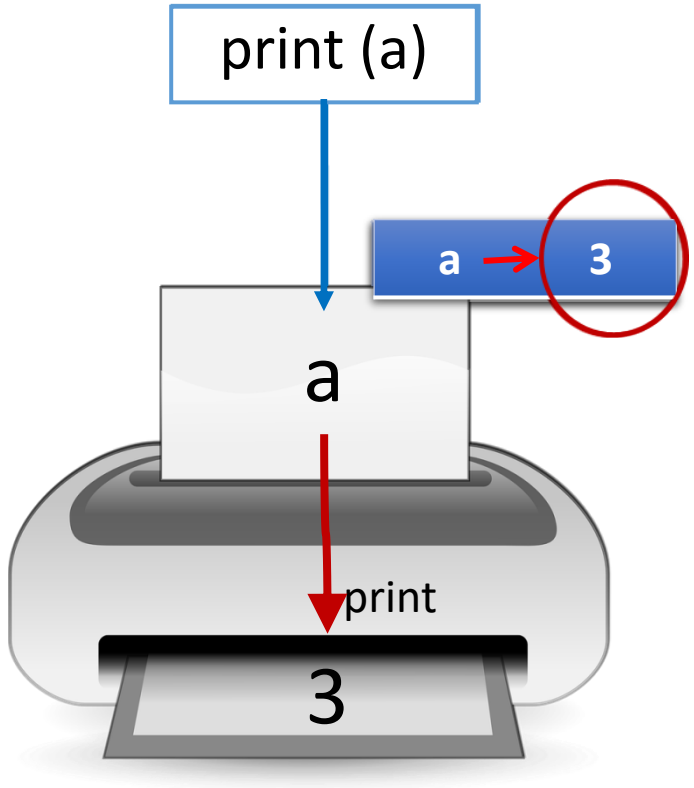
註解3：關於資料的輸出

- 雙引號內的會視為一般文字印出



- 不是放在雙引號內的會視為變數，程式會先找變數存放的值，再印出找到的那個存放值

```
a=3
print(a)
```



狀況1：在同一行輸出2個以上變數

```
a = 11
b = 12
print(a, b)
```

表示
空格

執行結果

```
11 12
```

狀況 2：指定兩個變數並且分兩行輸出

```
a = 11
b = 12
print(a)
print(b)
```

執行結果

```
11
12
```

非常直覺，
與指定1個變數與輸出1個變數
用法相同

狀況3：使用format解決輸出問題

- **第一種寫法**

```
print("{}\n{}".format(a, b))
```

- 雙引號
  - 後面接 .format(變數串)
  - 裡面用大括號 {} 代表要取 format() 裡面放的變數來替換在這個大括號的位置

- 雙引號內的大括號由左到右會依序對應 format() 裡面由左到右填的變數

```
a = 11
b = 12
print("{}\n{}".format(a, b))
```

執行結果

```
11
12
```

<缺點> 同一個變數要印10次就要寫10次，
雖然不用寫那麼多行的 print，但還是很麻煩！

狀況3：在同一行輸出2個換行的變數

第二種寫法

```
a = 11
b = 12
print("{0}\n{1}".format(a, b))
```

表示
換行

執行結果

```
11
12
```

- `print("{0}\n{1}".format(a, b))`
- {}代表未知的變數
- 0與1指的是後面format裡相對應的變數
  注意: 編號是由 **0** 開始喔！

狀況3：使用format解決輸出問題

- **第三種寫法**

```
print("{a}\n{c}".format(a=a, c=b))
```

- 第三種寫法的加強版
  - 將編號改用變數名稱
  - format() 內指定大括號內的變數名是對應哪個變數

- 注意: 大括號內的變數名稱是對應 format() 內的變數名稱，不是直接對應外面的變數名稱喔

```
a = 11
b = 12
print("{a}\n{c}".format(a=a, c=b))
```

執行結果

```
11
12
```

資料輸出的4種方法總整理

- 第一種寫法

```
print(a)
print(b)
```

- 第二種寫法

```
print("{}\n{}".format(a, b))
```

- 第三種寫法

```
print("{0}\n{1}".format(a, b))
```

- 第四種寫法

```
print("{a}\n{c}".format(a=a,
c=b))
```

狀況4：除了印變數內容外，加了其他字詞

```
a=11
b=12
print("我家有{0}隻狗".format(a))
Print("我家有{0}隻貓".format(b))
```

執行結果

```
我家有11隻狗
我家有12隻貓
```

狀況5：
在同一行輸出兩個變數，並且加上其他說明

```
a=11
b=12
print("我家有{0}隻狗{1}隻貓".format(a, b))
```

執行結果

我家有11隻狗12隻貓

概念 4：
固定寬度的輸出格式

整數 int與輸出

- int ：英文單字 integer 的縮寫
- 不含小數點的數字為整數
- 輸出整數資料時，可以使用
  {輸出位置:[寬度]d}
  控制輸出的格式
- d is decimal，十進位的意思

狀況6：固定寬度的輸出格式範例

寫一個程式，
指定一個整數為 160，代表萬元，印出其兩種輸出格式。
第一次輸出：直接印出
第二次輸出：印出時總數字寬度10

```
dollar = 160
print("金額{0}萬元".format(dollar))
print("金額{0:10d}萬元".format(dollar))
```

整數的輸入
與輸出範例

執行結果

```
金額160萬元
金額       160萬元
```

狀況6 ：整數 int輸出格式加強版

```
num = 10
print('num=<{num}>'.format(num=num))
print('num=<{num:d}>'.format(num=num))
"""
若變數值本身的輸出寬度空間大於設定的輸出寬度空間，
會沒有效果喔！

變數值(num=10)寬度空間為 2
因此若設定的寬度空間小於 2 就無效喔！
"""
print('num=<{num:1d}>'.format(num=num))
print( 'num=<{num:3d}>' .format(num=num)) # 輸出寬度
空間 為3
print( 'num=<{num:5d}>' .format(num=num)) # 輸出寬度
空間 為5
```

寬度(10) = 2
寬度(310) = 3
寬度(8) = 1
寬度(99) = 2
寬度(1000) = 4

執行結果

```
num=<10>
num=<10>
num=<10>
num=< 10>
num=<   10>
```