

Python Basic Syntax

01 打印

■ 基本打印輸出 `print(value, ... , sep="", end = "\n")`

- `value`：表示想要輸出的資料，可以多筆資料輸出，並以逗號隔開
- `sep`：放多個字串，預設會使用空白串成一行。可透過變更分隔符號，改變打印方式
- `end`：當資料輸出結束時所插入的字元，預設式插入換行字元

■ 格式化打印輸出 `print("輸出格式區"%(變數系列區))`

- `%d`：整數
- `%f`：浮點數
- `%s`：字串
- `%e`：科學記號

■ 增強格式化打印輸出 `print("輸出格式區".format(變數系列區))`

- 可透過編號改變順序

■ 不用迴圈式來打印出重複字串

01 基本打印	<code>print("Hello! Python")</code>
02 字串連接	<code>print('A', 'B', 'C')</code>
03 分隔符號	<code>print('A', 'B', 'C', sep = '/')</code> <code>print('A', 'B', 'C', sep = '\n')</code>
04 格式化打印	<code>print('我是{:s}，我的年齡是{:d}歲'.format('小明', 18))</code> <code>print('我的身高是%0.1f 公分，體重是%0.2f 公斤' %(175.86, 68.65))</code> <code>print('我的興趣是{0}、{2}和{1}'.format("打程式", "玩遊戲", "閱讀"))</code>
05 重複打印	<code>text = "我喜歡打程式 \t" * 5</code> <code>print(text)</code>

02 註解

- 「#」 可作註解，該行後面不會顯示
- 「'''」 三個雙引號可作註解，中間的不會顯示
- 「"""」 三個單引號可作註解，中間的不會顯示

06 註解	<code>#print("我不會打印出來，因為我是註解")</code>
07 三雙引號	<code>""" print("我不會打印出來，因為我是註解") """</code>
08 三單引號	<code>''' print("我不會打印出來，因為我是註解") '''</code>

03 變數的命名原則

- Python 系統保留字

Python 系統保留字				
False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

■ Python 系統內建函數

Python 系統內建函數				
abs()	delattr()	hash()	memoryview()	set()
all()	dict()	help()	min()	setattr()
any()	dir()	hex()	next()	slice()
ascii()	divmod()	id()	object()	sorted()
bin()	enumerate()	input()	oct()	staticmethod()
bool()	eval()	int()	open()	srt()
breakpoint()	exec()	isinstance()	ord()	sum()
bytearray()	fitter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	_import_()
complex()	hasattr()	max()	round()	

04 基本數學運算

■ 加、減、乘 與 除

■ 餘數 和 整除

■ 次方

■ 指派運算子

■ 等號多重指定使用

■ 刪除變數

09 加減乘除	<pre>a = 10; b = 3 print("加:", a+b) ; print("減:", a-b) print("乘:", a*b) ; print("除:", a/b)</pre>
10 餘數與整除	<pre>print("餘數:", a%b) ; print("整除:", a//b)</pre>
11 次方	<pre>print("次方:", a ** b)</pre>
12 等號多重指定 與 指派運算子	<pre>a1 = a2 = a3 = a4 = a5 = a6 = a a1 += 2 ; print("加等於:", a1) a2 -= 2 ; print("減等於:", a2) a3 *= 2 ; print("乘等於:", a3) a4 /= 2 ; print("除等於:", a4) a5 //= 2 ; print("整除等於:", a5) a6 **= 2 ; print("次方等於:", a6)</pre>
13 刪除變數	<pre>del a6 print(a6)</pre>

05 基本的資料型態

- 數值資料型態：整數、浮點數 與 複數 (不常用)
- 布林值資料型態
- 文字序列型態
- 字元組、序列型態、對映型態 與 集合型態(後面介紹)

Python 中不相同資料型態間，部分也可以執行運算，例如：整數與浮點數之間

14 數值資料型態	<pre>a = 10 ; b = 0.5 print(type(a)) ; print(type(b))</pre>
15 強制轉換	<pre>b = int(b); print(type(b)) ; print(b) b = str(b); print(type(b))</pre>
16 數值運算中的 常見函數	<pre>print(abs(-2)) #絕對值 print(pow(2,3)) #次方 print(round(4.7777,1)) #四捨五入 print(round(6.2222,3))</pre>

17 布林資料型態	<pre> a = False ; b = True print(b) ; print(type(b)) print(a+b) ; print(type(a+b)) </pre>
18 字串資料型態	<pre> a = 'I'm Peter' #字串出現單引號有問題 a = "I'm Peter." #字串出現單引號，使用雙引號 print(a) a = 'I\'m Peter' #字串出現單引號，使用逸出字元 print(a) b = "I love Python." c = a+"\t"+b #字串透過加號來串成新字串 print(c) d = r"123\n321" #使用 r 防止逸出字元被轉譯 print(d) </pre>
19 ASCII 碼	<pre> e = 97 ; print(chr(e)) e -= 32 ; print(chr(e)) </pre>

06 開啟檔案/輸入輸出資料

06-1 開啟檔案

```
file_Obj = open(file, mode="r")
```

■ **file**：用字串列出欲開啟的檔案

■ **mode**：開啟檔案的模式

■ 第一個字母操作意義

- **r** 是預設，開啟檔案供讀取
- **w** 開啟檔案供寫入，如果原先檔案有內容將被覆蓋
- **a** 開啟檔案供寫入，如果原先檔案有內容，新寫入資料將附加在後面
- **x** 開啟一個新的檔案供寫入，如果所開啟的檔案已經存在會產生錯誤

■ 第二個字母操作意義

- b 開啟二進位檔案模式
- t 是預測，開啟文字檔案模式

■ file_Obj：不使用要正常關閉，要使用 `file_Obj.close()`

06-2 輸入輸出資料

```
value = input("prompt:")
```

■ value：是變數，所輸入的資料會儲存在此變數內，不管回傳的資料是何種型態，回傳至 value

時皆為字串資料，如果要執行數學運算要用強制轉換等方式 `result = eval(expression)`

■ eval：可以回傳字串內數學表達式的計算結果，expression 必須為字串

20 開啟並寫入	<pre>text1 = open("text_1.txt", mode="wt") print("Python", file=text1) ; text1.close text2 = open("text_1.txt", mode="at") print("C++", file=text2) ; text2.close</pre>
21 資料輸入	<pre>name = input("請輸入姓名：") h = input("請輸入身高：") w = input("請輸入體重：") print("我是%s，身高是%s 公分，體重是%s 公斤" %(name, h, w)) print("資料型態是%s, %s, %s" %(type(name), type(h), type(w))) bmi = float(w)/((float(h)/100)**2) print("我的 BMI 是%0.2f" %(bmi))</pre>
22 公式輸入	<pre>eq_str = input("請輸入公式：") eq = eval(eq_str) print("計算結果為：%0.2f" %(eq))</pre>

07 運算子

07-1 關係運算子

運算符號	描述
<	小於
<=	小於或等於
>	大於
>=	大於或等於
==	比較值是否相等
!=	比較值是否不相等

07-2邏輯運算子

- and 且 / or 或 / not 反

23 關係運算子	<pre> a = 2 print(a > 3) print(a == 2) print(a != "2") print(str(a) == "2") </pre>
24 邏輯運算子	<pre> b = (4>2) and (-1>-5) print(b) c = not(2>1) print(c) </pre>

08 流程控制

- if
- if else
- if elif else
- 巢狀 if

25 if	<pre> if(True): print("程式執行了!") </pre>
26 if else	<pre> year = input("請輸入年齡：") if(int(year)<20): print("你年齡太小，需要年滿 20 歲才能購買菸酒") else: print("歡迎購買菸酒") </pre>
27 if elif else	<pre> year_name = ["未成年", "青年人", "老年人"] year = int(input("請輸入年齡：")) if(0<year<18): print("你是", year_name[0]) elif(18<=year<65): print("你是", year_name[1]) elif(65<=year<150): print("你是", year_name[2]) else: print("你不是人!") </pre>

09 列表(list)

- 切片
- 索引
- 統計資料：最大值/最小值/總和/個數
- 更改列表元素
- 刪除列表元素
- 列表多重指定
- 增加、插入與刪除列表的元素
- 列表的排序
- 列表內特定元素出現的次數

28 切片與索引	<pre> fruits = ['蘋果', '香蕉', '橘子'] print(fruits) ; print(fruits[0]) number = [0,1,2,3,4,5,6,7,8,9,10] matrix = [[0,1,2],[3,4,5],[6,7,8]] print(number[2:5]) print(number[:3]) print(number[:-3]) print(number[7:]) print(number[-7:]) print(number[-1]) print(matrix[0][2]) print(matrix[1][1:]) print(matrix[-1][-1]) print(matrix.index([0,1,2])) </pre>
29 統計資料	<pre> a = [1,2,3] print(min(a)) ; print(max(a)) print(sum(a)) ; print(len(a)) </pre>
30 更改列表元素	<pre> a[0] = 4 print(a) </pre>
31 刪除列表元素	<pre> del a[1] print(a) </pre>
32 多重指定	<pre> a, *b, c = 1,2,3,4,5,6 print(a,b,c) </pre>
33 增加、插入與 刪除列表中的元 素	<pre> b = [3,4,5] ; c = [1,1,1] print(b+c) c2 = [2] c2.extend(c) print(c2) c.append(2) print(c) b.insert(1,3.5) print(b) </pre>

	<pre>d1, d2 = [1,2,3,4], [1,2,3,4] print(d1.pop()) ; print(d1) print(d2.pop(2)) ; print(d2)</pre>
34 列表的排序	<pre>e1, e2 = [3,1,2,5,4], [3,1,2,5,4] e1.sort() e2.reverse() print(e1) ; print(e2)</pre>
35 列表元素次數	<pre>f = [0,0,1,0,1] print(f.count(0))</pre>

10 迴圈設計

10-1 for 迴圈

- for var in 可迭代物件：

「程式區塊」

可迭代物件可以是串列、元組、字典、集合或 `range()`，上述與法可將可迭代物件的元素當作 `var` 重複執行，直到每個元素皆被執行一次，整個迴圈才會停止

- `range(start, stop, step)`

`range()` 函數可以產生一等差級序列，常用於 for 迴圈的計數器

- 列表生成之應用
- `enumerate` 物件：可以讓索引值與元素值配對出現
- 巢狀 for 迴圈

- 強制離開 for 迴圈指令：break
- for 迴圈不往下執行，跳至下一次迴圈中：continue

10-2 while 迴圈

- while 條件運算：

「程式區塊」

- 巢狀 while 迴圈
- 強制離開 while 迴圈指令：break
- while 迴圈不往下執行，跳至下一次迴圈中：continue

36 基本 for 迴圈	<pre> a = [1,2,3,4] for i in a: i += i print(i) for i in a[1:]: print(i) </pre>
37 基本 for 迴圈	<pre> name = [[26, 'Tim',], [66, 'Peter'], [30, 'Tom'], [18, 'Tiny']] young_name = [] for y, n in name: if (20<=y<65): young_name.append(n) print(young_name) </pre>
38 range	<pre> for i in range(5): print(i,end=" ") for i in range(0,10,2): print(i,end=" ") </pre>
39 列表	<pre> save_number = [] for i in range(0,10,1): </pre>

	<pre> save_number.append(i) print(save_number) save_number = list(range(0,10,1)) print(save_number) </pre>
40 enumerate	<pre> alphabet = ['A', 'B', 'C', 'D'] for alp in enumerate(alphabet): print(alp) for index, alp in enumerate(alphabet): print(index, alp) for index, alp in enumerate(alphabet,10): print(index, alp) </pre>
41 break	<pre> for i in range(20): if (i>8): break print(i,end=' ') </pre>
42 continue	<pre> score_name = [['Kint',100],['Peter',60], ,['Sam', 45],['Tom', 59],['Tim',70]] good = 0 save_name = [] for name, score in score_name: if score < 60: continue good += 1 save_name.append(name) print("及格有%d個人，名字為：%s" %(good,save_name)) </pre>
43 while	<pre> a = 0 while a<10: a += 1 print(a,end=' ') </pre>

11 元組(Tuple)

■ tuple=(元素 1, ..., 元素 n)

- 與列表類似，最大差異是元素值與元素個數不可更動，不可以再給予賦值
- 列表與元組資料互換

- 使用 zip 打包多個物件

■ 優點

- 可以更安全保護資料
- 加快程式執行速度，占用較少的系統資源

44 建立元組	<pre>number = (1,2,3,4,5) print(type(number)) alphabet = 'A', 'B', 'C' #快速建立元組的方法 print(alphabet)</pre>
45 無法修改元素	<pre>number[0] = 1</pre>
46 元組切片(與列表相同)	<pre>number = (0,1,2,3,4,5,6,7,8,9,10) matrix = ((0,1,2),(3,4,5),(6,7,8)) print(number[2:5]) print(number[:3]) print(number[:-3]) print(number[7:]) print(number[-7:]) print(number[-1]) print(matrix[0][2]) print(matrix[1][1:]) print(matrix[-1][-1])</pre>
47 列表與元組 資料互換	<pre>number = list(number) print(number) number = tuple(number) print(number)</pre>
48 zip 打包	<pre>year = (20, 18, 30, 42, 25) name = ('Kint', 'Peter', 'Sam', 'Tom', 'Tim') zipdata = zip(year, name) print(type(zipdata))</pre>

```
print(list(zipdata))
for i in zip(year, name):
    print("年齡：%d, 姓名：%s" %(i[0],i[1]))
```

12 字典(Dict)

■ 定義字典：

- 字典是一個非序列的資料結構，元素以"鍵(key):值(value)"方式配對儲存

```
dict={key1:value1, ..., key2:value2}
```

- key 通常以字串或數字當作是鍵，在一個字典中不可有重複的 key 出現。value 則可以任何 python 的資料物件

■ 基本操作

- 建立/讀取
- 增加字典元素
- 更改字典元素內容
- 刪除字典的 key 與元素
- 字典拷貝
- 驗證 key 是否存在
- 合併字典
- 列表/元組轉為字典

49 建立/讀取

```
fruits = {'西瓜':15, '香蕉':20, '水蜜桃':25}
print(fruits)
print(type(fruits))
```

	<pre>print("水蜜桃一斤 = ", fruits['水蜜桃'], "元") print("水蜜桃一斤 = ", fruits.get('水蜜桃'), "元") print("蘋果一斤 = ", fruits.get('蘋果'), "元")</pre>
50 增加字典元素	<pre>fruits['橘子'] = 18 print(fruits)</pre>
51 更改字典元素	<pre>fruits['水蜜桃'] = 12 print("新價格水蜜桃一斤 = ", fruits['水蜜桃'], "元")</pre>
52 刪除 del	<pre>del fruits['水蜜桃'] print(fruits)</pre>
53 刪除 pop	<pre>fruits.pop('西瓜') print(fruits)</pre>
54 清除整個字典	<pre>fruits.clear() print(fruits)</pre>
55 字典拷貝	<pre>fruits = {'西瓜':15, '香蕉':20, '水蜜桃':25} new_fruits = fruits.copy() print(fruits == new_fruits) print(fruits is new_fruits) print("fruits 位址 = ", id(fruits)) print("new_fruits 位址 = ", id(new_fruits))</pre>
56 驗證 key 是否 已存在	<pre>fruits = {'西瓜':15, '香蕉':20, '水蜜桃':25} key = input("請輸入 key = ") value = input("請輸入 value = ") if key in fruits: print("%s 已經在字典了" %key) else: fruits[key] = value print("新的 fruits 字典內容 = ", fruits)</pre>
57 合併字典	<pre>num1 = {1:"one", 2:"two"} num2 = {10:"ten", 6:"sex"} num1.update(num2) print(num1)</pre>
58 列表/元組轉 為字典	<pre>country_list = [['日本','東京'], ['泰國','曼谷'], ['英國','倫敦']] country_dict = dict(country_list) print(country_dict)</pre>

13 函數(Function)

- 函數設計
- 單個參數與多個參數
- 關鍵字參數(參數名稱=值)
- 參數預設值
- 傳回值
- 全域變數與區域變數

59 函數設計	<pre>def fun_1(): print("這是我第一個 python 函數") fun_1()</pre>
60 參數設計	<pre>def calculator(num1, num2, cal): if cal == "+": print(num1+num2) elif cal == "-": print(num1-num2) elif cal == "*": print(num1*num2) elif cal == "/": print(num1/num2) else: print("你輸入的值為非運算子") calculator(8, 2, "+") calculator(16, 4, "/") calculator(16, 4, 5)</pre>
61 傳回值	<pre>def fun_2(val): val += 1 return val num = fun_2(2) print(num)</pre>

62 全域變數與區域變數

域變數

```
def var():  
    msg = '區域變數'  
    print(msg)  
  
msg = '全域變數'  
print(msg)  
  
var()
```

14 類別(Class)

- 封裝
- 繼承
- 多型

15 程式異常處理

...

#沒補充到的 Python 基礎程式，請學生自我回家學習！以下有數個網站推薦：

1. Python Tutorial (Python 2.7 and 3.5)

<https://openhome.cc/Gossip/CodeData/PythonTutorial/index.html>

2. Python 入門教學課程：彭彭的課程

https://www.youtube.com/playlist?list=PL-g0fdC5RMboYEyt6QS2iLb_1m7QcgfHk

3. Python 基础教程

<https://www.runoob.com/python/python-tutorial.html>