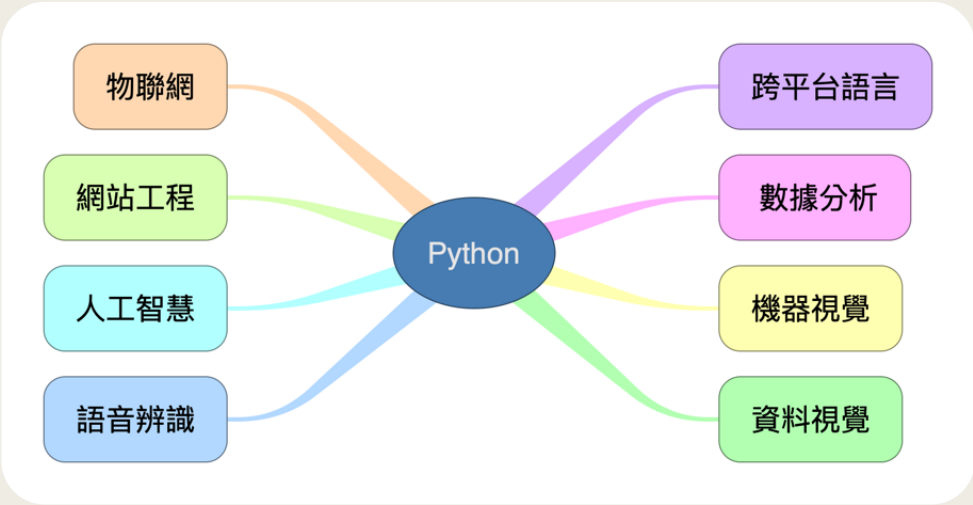
基本語法

朱克剛

簡介

- 1991年誕生的老語言，但這幾年回春，爆紅
- 目前穩居全球語言排名第四位
  - *https://www.tiobe.com/tiobe-index/*
- 應用領域



安裝

- Python 官網
  - *https://www.python.org*
- Windows 平台安裝過程必須勾選「加到環境變數」選項，若忘記勾選，建議重裝。
- 開發工具
  - *本課程僅使用標準編輯軟體，例如 notepad++ 或是 vi*
  - *其他豪華版 IDE 依個人需求自行選用，例如PyCharm*
- 執行環境
  - *Windows：命令提示字元*
  - *UNIX：terminal*

執行 Python

- 在命令提示字元（或是UNIX的terminal）下輸入 python 即可進入 Python 的命令模式（直譯環境）

```
C:\> python
Python 2.7.10 (default, Feb  6 2017, 23:53:20)
[GCC 4.2.1 Compatible Apple LLVM 8.0.0 (clang-800.0.34)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- 離開為 quit()

執行方式

- 存檔成 hello.py
  - *副檔名習慣命名為 .py*

```
print ("hello world")
```

- 執行方式
  - *指令打：python hello.py*

變成執行檔（UNIX系統）

- 存檔成 hello.py
  - *副檔名習慣命名為 .py*

```
#!/usr/bin/python

print ("hello world")
```

- chmod 755 hello.py
  - *設定 hello.py 為執行檔*
- 執行 ./hello.py

註解

- 目的：
  - *程式碼中放一些幫助人們理解程式內容的字句，但 Python 會忽略這些字句不執行他們*
- 單行註解 - #
  - *# 這是註解*
- 多行註解 – """
  - *""" 多行註解開始*
  - *內容*
  - *多行註解結束 """*

編碼

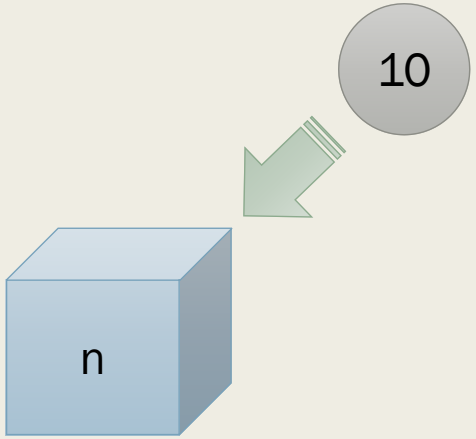
- 程式中若包含中文字，需使用 utf-8 編碼
  - *Python3預設為 utf-8*
- 以下敘述放在程式第一行（python2 必加）

```
#coding=utf-8
```

- 因各平台編碼不同，Python非常容易出現亂碼問題，請參考此網址處理
  - *https://openhome.cc/Gossip/Encoding/Python.html*

變數

- 目的：
  - *儲存資料用*
- 變數命名規則
  - *不可數字開頭*
  - *單獨一個英文字，中間不可空白*
  - *不可中文*
  - *不可使用特殊符號，但可使用底線*
- 命名慣例
  - *第一個字母小寫*
  - *兩個以上的英文組合可使用駝峰命名*



變數不需宣告

- Python 變數不需要宣告，隨叫隨用
- 變數的型別會依據等號右邊的資料來決定
  - *例如：x為 int，y為float*

```
x = 10
y = 20.3
print(x + y)
```

四種純量型別

- int 整數
  - *10, 20, 30*
- float 小數
  - *3.1415, 5.0*
- str 字串
  - *'hello world', '123'*
- bool 布林
  - *True, False*

型別轉換

- 數字轉字串

  s = str(10)

- 字串轉整數

  n = int("5")

- 字串轉小數

  f = float("3.14")

- 轉布林

  b = bool('True')

  b = bool(1)

  b = bool(0)

標準輸入輸出

- 標準輸入：鍵盤；標準輸出：螢幕
- 輸入：
  - *Python3為input()*
  - *Python2為raw_input()*
- 輸出：print()
  - *python2，print後方括號可以省略*
- 例如：

```
name = input("What's your name? ")
print ("Hello, " + name)
```

數字運算

此為python2，
python3為 float

```
>>> 17 / 3      # int / int -> int
5
>>> 17 / 3.0    # int / float -> float
5.666666666666667
>>> 17 // 3.0   # 取整數
5.0
>>> 17 % 3     # 取餘數
2
>>> 5 ** 3      # 5的3次方
125
```

字串

- 使用 '...' 或 "..." 夾住字串
- 倒斜線 \ 為跳脫字元
  - *\t：Tab*
  - *\n：換行*
- 使用 print 印出字串

注意字串修飾子 r

  - *print(r"hello\nworld")*
  - *print("5 + 3 = {0}".format(5 + 3))*
  - *print("hello world", end='')　#python3不換行*
  - *print "hello world",　#python2不換行*

多行字串

- 多行文字可使用 """...""" 或是 '''...'''

```
print ("""\
Usage: thingy [OPTIONS]
     -h                        Display this usage message
     -H hostname               Hostname to connect to
""")
```

字串相加

- 使用 + 號或是空白鍵

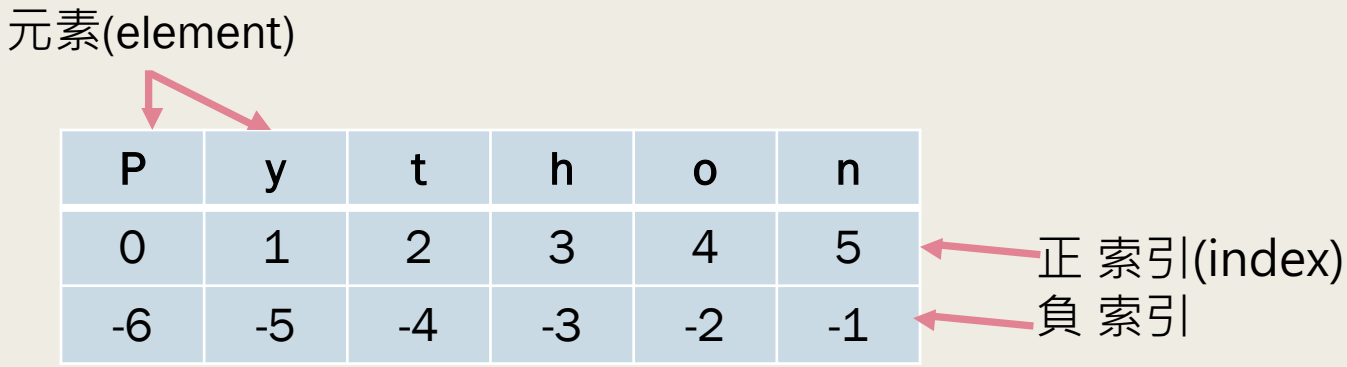
```
print ("a" + "b")

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

字串長度

- 使用 len() 函數
  - *len("hello")*

字串是字元陣列




```
>>> text = "Python"
>>> text[3]
'h'
>>> text[:2]
'Py'
>>> text[2:-2]
'th'
>>>
```

命令列參數

- 在命令列輸入的參數會放在 sys.argv 中，argv 為陣列

```
import sys

print (sys.argv[1])
```

- 測試
  - *python test.py hello*
  - *hello*

流程控制

條件判斷

```
#coding=utf-8

s = raw_input("請輸入關鍵字: ")

if "關燈" in s:
    print("關鍵字中有「關燈」")
elif "開燈" in s:
    print("關鍵字中有「開燈」")
else:
    print("沒有特定關鍵字")
```

運算子與布林值

- 布林值
  - *True / False*
- 邏輯運算子
  - *and, or , not*
- 先乘除後加減，搞不清楚的就用小刮號
  - *例如 not (a == b)*
  - *或 (not a) == b*

```
if x and y:
    // x 與 y 均成立

if x or y:
    // x 或 y 成立

if not x:
    // x 不成立
```

for

```
zoo = ['貓', '狗', '獅子', '長頸鹿']
for animal in zoo:
    print ("動物園有" + animal)
```

range()

```
>>> range(10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> range(2, 5)
[2, 3, 4]
>>> range(0, 10, 2)
[0, 2, 4, 6, 8]
```

1 + 2 + 3 + ... + 10

```
sum = 0
for i in range(1, 11):
    sum += i

print ("答案是", sum)          # 答案是 55
```

while

- 費氏數列（Fibonacci）：生兔子故事
- https://zh.wikipedia.org/wiki/斐波那契数列

```
a, b = 0, 1
while a < 50:
    print (a)
    a, b = b, a+b

# 印出 0 1 1 2 3 5 8 13 21 34
```

break, continue, else

- break 用來中斷迴圈
- continue 用來立即返回到迴圈開始處
- else是當迴圈結束（不包含用break結束的迴圈）後呼叫的地方

```
for i in range(3):
    print (i)
else:
    print ("迴圈結束")
```

```
執行後
0
1
2
迴圈結束
```

pass 語句

- pass 語句的目的是不做任何事情，例如下述是錯誤的程式碼

```
test = True
if test:
else:
    print ("測試失敗")
```

- 需改為

```
test = True
if test:
    pass
else:
    print ("測試失敗")
```

資料結構

list（串列）

- 可變長度的陣列，每個元素的資料型態可以不一致
- 建立
  - *list = ["a", "b", "c", 4, 5, 6]*
- 取出
  - *print(list[3])　　# 印出4*

list 常用函數

- len(list)：計算數量
- list.append(obj)：附加
- list.insert(index, obj)：在 index 所在位置插入元素
- list.remove(obj)：刪除某元素
- list.sort()：排序
- list.reverse()：反向
- 參考資料
  - https://docs.python.org/3/tutorial/datastructures.html

tuple

- 由一連串資料組成，有順序性且不可更動內容
- 建立
  - *tp = ("a, "b", "c", 4, 5, 6)*
- 取出
  - *print (tp[3])*

常用函數

- len(tp)：計算數量

set（集合）

- 無順序性，無重複資料
- 建立
  - *set = {"a", "b", "c"}*
- 無取出方式，只能透過 in 來確定某資料是否有在集合中
  - *例如：* *"b" in set 會傳回 True*

常用函數

- len(set)：計算數量
- set.add(obj)：將 obj 加入集合中
- set.remove(obj)：將 obj 從集合中刪除
- set.issubset(S)：判斷集合S是否為set的子集合
- set.issuperset(S)：判斷集合S是否為set的超集合
- set.isdisjoint(S)：若集合set與集合S交集為空集合時傳回True

集合運算

- set.union(S)：聯集
- set.intersection(S)：交集
- set.difference(S)：差集

dict（字典）

- Key-value形式
- 建立
  - *dict = {"uid": "A01", "salary": 50000}*
- 取出
  - *print(dict["uid"])　　# 印出 A01*
- 新增
  - *dict["sex"] = "M"*

常用函數

- len(dict)：計算數量
- del dict["uid"]：刪除 uid 這個 key
- 使用 in 與 not in 判斷某個 key 是否在字典中
- dict.keys()：傳回所有的 key
- dict.values()：傳回所有的 value
- dict.items()：傳回所有的 key-value 配對

錯誤處理

使用 try ... except

```
#coding=utf-8

try:
    n = input("請輸入一個數字: ")
    n = 1 + int(n)
    print(n)
except Exception as error:
    print("error: ", error)
```

丟出錯誤

- 使用 raise 丟出錯誤

```
#coding=utf-8

s = input("請輸入 「hello」 ")
if s <> "hello":
    raise ValueError
```