

物件與類別

基礎程式設計 - Python

俞伯傑@YZU

複習

- `tmp = calendar.monthcalendar(2019,12)`
- 讓使用者輸入年月日，可以輸出那天是禮拜幾？

類別 (class)

- 類別？什麼是類別？怎麼定義？
- 物件？怎麼建立物件？

類別 (class)

- 用來製造物件的”模板”
- 可以自行定義此種類別的屬性與方法
- 例如: 人, 車, 學生...etc.

類別的內容

- 以”人”為例
- 屬性：性別/姓名/年齡/身高/體重
- 方法：計算 BMI

範例

`__init__` 是用來初始化物件的方法

- **`class person():`** 在定義類別的第一個參數必須是 **`self`**

`def __init__(self, gender, name, age, height, weight):`

`self.gender = gender`

`self.name = name`

`self.age = age`

`self.height = height`

`self.weight = weight`

屬性

`def BMI(self):`

`return self.weight/((self.height/100.0)2)`**

方法

- 定義物件範例:
- **a = person('Male','PCYu',20,75,180)**
- 查看person類別的定義
- 建立一個新物件，呼叫 `_init_` 來初始化，傳遞給 `self`，且把參數個別傳給 `gender`, `name`, `age`, `weight`, `height` 儲存
- 回傳此物件指派給 `a`
- 此時 `a` 是一個物件，它的類別是 `person`
- `a`擁有 `person`類別所定義的所有性質

print看看

因為a是一個person類別的物件，可以存取所有屬性跟方法，利用 **a.屬性** 或是 **a.方法**：

```
print(a.name)
print(a.gender)
print(a.age)
print(a.weight)
print(a.height)
print(a.BMI())
```


練習

- 試著定義一個名為 student 的類別
- 屬性包含: name (姓名), gender (性別), id(學號) ,
而且還有空串列 grades[]
- 方法包含: (1) add(grade)新增成績到 grades;
(2) 回傳 grade的平均分數

```
class student():
    def __init__(self,name,gender,id):
        self.gender = gender
        self.name = name
        self.id = id
        self.grades = []
    def add(self,newgrade):
        self.grades.append(newgrade)
    def avg(self):
        return (sum(self.grades)/len(self.grades))
```

```
a=student('ABC','male','s1045555')
a.add(90)
a.add(60)
a.add(75)
print(a.name)
print(a.gender)
print(a.grades)
print(a.avg())
```

練習

- 定義一個銀行賬戶的類別
- 屬性有: 姓名，賬戶號碼，餘額
- 方法有: (1) 存款後顯示餘額 (2) 提款檢查餘額是否足夠，提款後顯示餘額

```
class account():
    def __init__(self,name,sid):
        self.name = name
        self.sid = sid
        self.balance = 0
    def deposit(self,amount):
        if amount > 0:
            self.balance += amount
            print('Deposit successfully, current balance: '+str(self.balance))
        else:
            print('ERROR! Deposit amount must > 0')
    def withdraw(self,amount):
        if amount <= self.balance:
            self.balance -= amount
            print('Withdraw successfully, current balance: '+str(self.balance))
        else:
            print('ERROR! Balance not enough! Please save more money!!!')
```

```
a=account('Jack','123-444')
a.deposit(10000)
a.deposit(-200)
a.withdraw(3000)
a.withdraw(200000)
```

練習

- 定義一個檔案的類別叫 my_file
- 屬性有: 檔名，文字內容
- 方法有: (1) 覆蓋檔案內容 (2) 新增檔案內容 (3) 修改檔名