

# python 物件導向

## 設計與構想

類別(class)

- 類別擁有初始值(\_\_init\_\_)
- 建構式
- 屬性 — 變數
- 方法 — 函式

```
class Example():  
    Example_attribute = 0  
  
    def Example_method():  
        return "This is Example"
```

```
class Example():  
    def __init__(self, example_variable):  
        self.example_variable = example_variable # 定義屬性  
  
    def Example_method(self): # 定義方法  
        print("this is instance ", self.example_var)  
  
Instance_Boject = Example(10) # 以 Example 類別創建一個名為 Instance_Boject 的物件  
# p.example_variable = 10  
print(Instance_Boject.example_variable)
```

封裝

設定方法(method)、屬性為 public(公有) 或是 private(私有)

- 方法: 在屬性或方法前面增加兩個底線(eg. \_\_名字)
- 用途: 避免外部直接使用特定的方法或屬性, 導致錯誤

```
class Example():  
    def __init__(self, example_variable):  
        self.example_variable = example_variable # 定義屬性  
        self.__private_attribute = 22  
  
    def Example_method(self): # 定義方法  
        print("this is instance ", self.example_variable)  
        print("this is private attribute", self.__private_attribute)  
  
    def __private_method(self):  
        print("this is private method")  
  
Instance_Boject = Example(10) # 以 Example 類別創建一個名為 Instance_Boject 的物件  
Instance_Boject.Example_method()  
# Instance_Boject.__private_method() 錯誤!!
```

擴展類別(class)的方法

繼承

在一個新建的類別, 引入其他類別的 method 與 attribute

- 方法: 在類別(class)後面的括號內放入要被繼承的類別名稱
- 用途: 可以不用重複建立相似的類別
- 在不同檔案裡要記得先 import
- 觀念: 被繼承的為「父類別」, 繼承者為「子類別」

```
class Son_class(Example):  
    def son_method(self):  
        print("this is son instance ", self.example_variable)  
        # print("this is private attribute", self.__private_attribute) 但私有屬性無法繼承
```

我們可以使用 Super() 取得私有方法的回傳值, 所以我們只要將私有的 attribute 放入私有 method 並回傳即可

多型

同一個方法名稱卻有不同的功能

- 方法: 將方法(method)放在不同的類(class)
- 用途: 易於擴充及降低類別間的相依性

```
class Login():  
    def login(self):  
        pass  
# Facebook登入機制  
  
class FacebookLogin(Login):  
    def login(self):  
        print("Facebook login implementation.")  
#Google登入機制  
  
class GoogleLogin(Login):  
    def login(self):  
        print("Google login implementation.")  
  
fb = FacebookLogin()  
fb.login()  
google = GoogleLogin()  
google.login()
```

## 實際使用

產生物件

- 實例 — Son\_Boject = Son\_class(10) # 以 Example 類別創建一個名為 Instance\_Boject 的物件
- 將 class 設給我們的變數