มหาวิทยาลัยราชภัฏนครปฐม
Nakhon Pathom Rajabhat University

บทที่ 12 การเขียนโปรแกรมเชิงวัตถุใน python

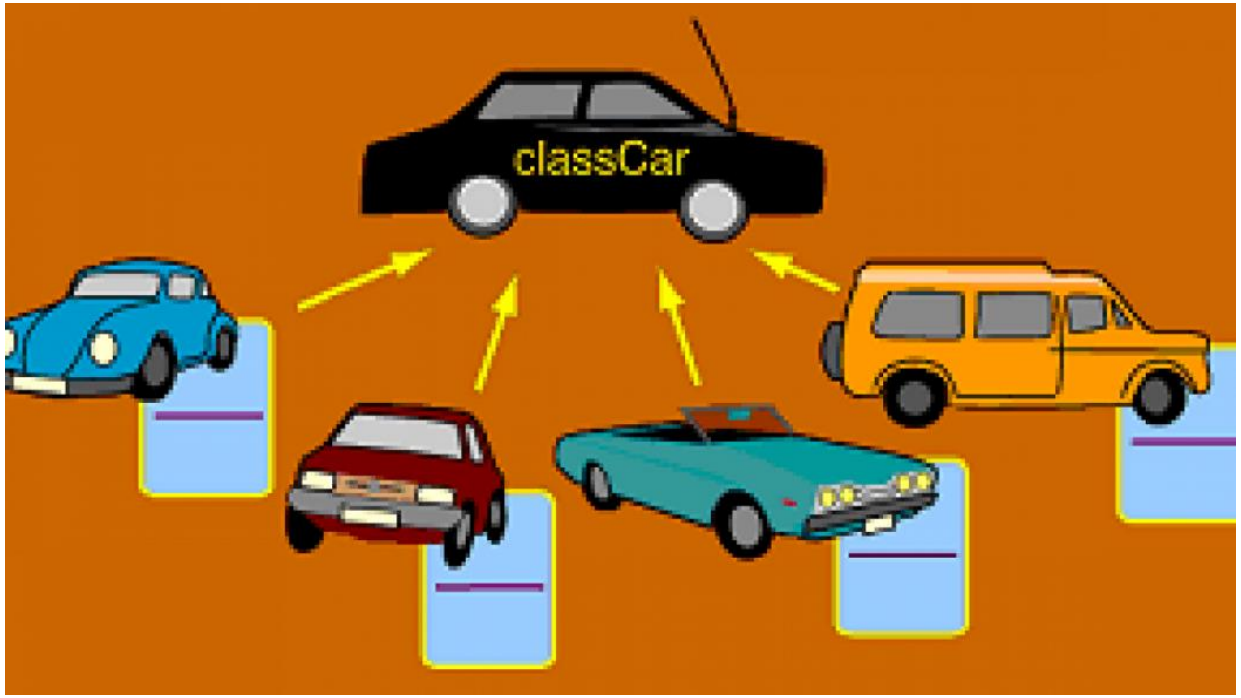
รายวิชา 3602801 การเขียนโปรแกรมคอมพิวเตอร์ทางธุรกิจ

ผศ.ดร.เดช ธรรมศิริ

Python Classes และ Objects

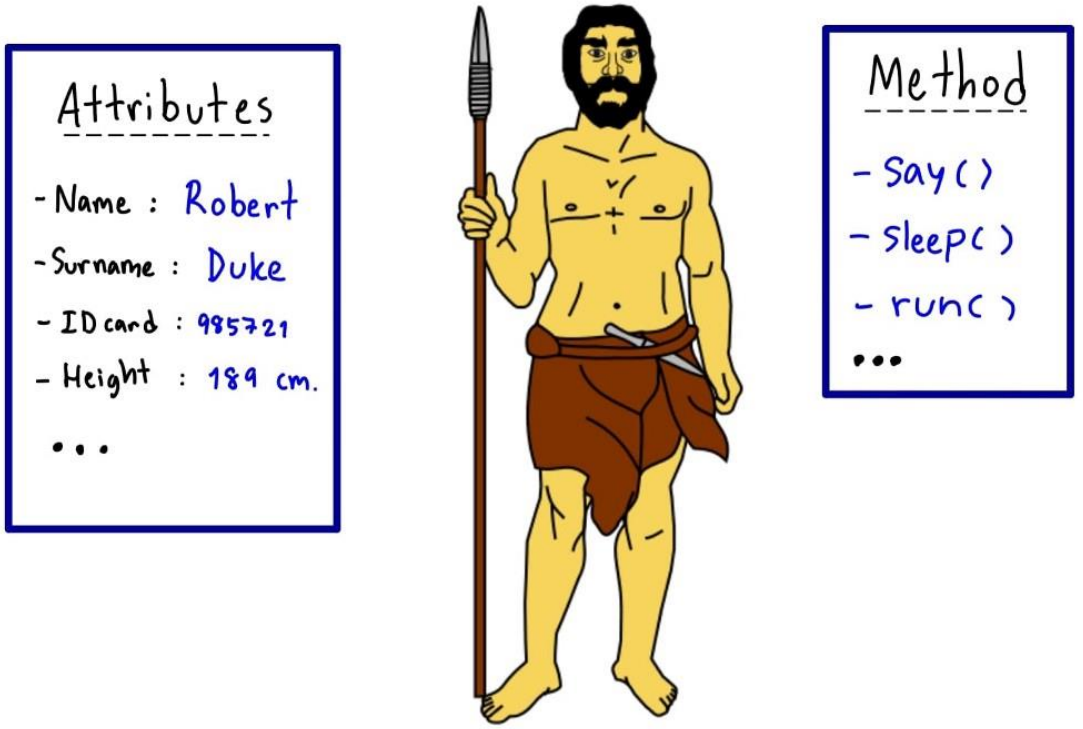
- **คลาส** คือประเภทข้อมูลที่สร้างโดยผู้ใช้ โดยจะนำไปใช้สร้างออบเจ็ค กล่าวอีกนัยหนึ่ง คลาสคือประเภทข้อมูลของออบเจ็ค
- **ออบเจ็ค** คือสิ่งที่สร้างมาจากคลาสหรือ class instances
- **แอตทริบิวต์** (instance attributes) คือข้อมูลที่เป็นสมาชิกของแต่ละออบเจ็ค โดยมักจะกำหนดไว้ในเมธอด __init__() ของคลาส
- **เมธอด** คือฟังก์ชันการทำงานที่กำหนดไว้ในคลาส
- **คลาสแอตทริบิวต์** (class attributes) คือตัวแปรที่ประกาศไว้ในคลาส ซึ่งจะแชร์กับออบเจ็คทั้งหมดที่สร้างจากคลาสนั้นๆ

Class เหมือนกับ ต้นแบบ,แม่แบบ,แม่พิมพ์,พิมพ์เขียว,ต้นฉบับ



ตัวอย่าง ให้เข้าใจง่าย ๆ

- Object ⟶ มนุษย์ ประกอบด้วย คุณสมบัติ (Attributes) คือ ชื่อ,นามสกุล,รหัสบัตรประชาชน,ส่วนสูง..... และมี พฤติกรรม (Method) เช่น พูด,นอน,นั่ง,วิ่ง



การสร้าง class

- สร้าง class ชื่อ my class และคุณลักษณะ(property) ชื่อ x

- ```
  class MyClass:
      x = 5
  ```
  ```

สร้าง object จาก คลาสต้นแบบ

- ```
  class MyClass:
      x = 5
  ```

- ```
  p1 = MyClass()
  print(p1.x)
  ```

```
5
```

การใช้งาน __init__() Function

- # ทุก class จะมี __init__() function มาให้ สำหรับกำหนดค่าให้กับ properties ของ object เมื่อเราต้องการสร้าง object ใหม่

- ```
  class Person:
    def __init__(self, name, age):
      self.name = name
      self.age = age

  p1 = Person("John", 36)

  print(p1.name)
  print(p1.age)
  ```

```
John
36
```
```

Object Methods

- #สามารถเพิ่ม function เข้าไปได้
-
```
class Person:
  def __init__(self, name, age):
    self.name = name
    self.age = age

  def myfunc(self):
    print("Hello my name is " + self.name+" , I am "+str(self.age)+ " year old")

p1 = Person("John", 36)
p1.myfunc()
```

```
Hello my name is John , I am 36 year old
```

สามารถแก้ไขค่าคุณลักษณะใน object ได้

```
class Person:
  def __init__(self, name, age):
    self.name = name
    self.age = age

  def myfunc(self):
    print("Hello my name is " + self.name+" , I am "+str(self.age)+ " year old")

p1 = Person("John", 36)
p1.age=40

p1.myfunc()
```

```
Hello my name is John , I am 40 year old
```

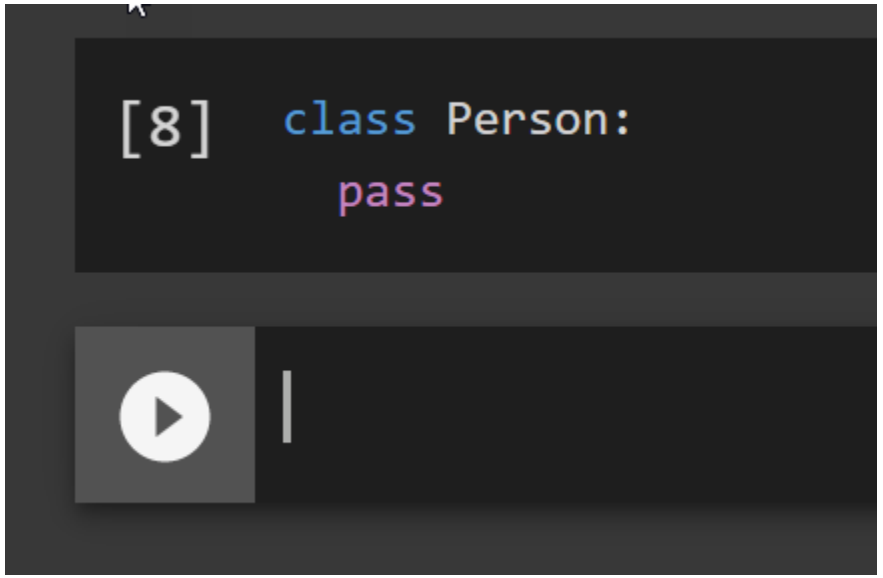
สามารถลบ Object ได้ ด้วย del

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-7-b69a132ac619> in <module>()
      9 p1 = Person("John",36)
     10 del p1
---> 11 p1.myfunc()

NameError: name 'p1' is not defined
```

```
        print("Hello my name is " + self.name)

p1 = Person("John", 36)
del p1 # บรรทัดนี้ลบ object ชื่อ p1
print(p1)
```

หากไม่มีค่าใน class ใช้ pass ได้

-
```
class Person:
    pass
```



Python Inheritance

- การที่คลาสหรือออบเจ็ค ได้รับการถ่ายทอดแอตทริบิวต์และเมธอดจากคลาสอื่น นั่นจะทำให้คลาสดังกล่าวมีแอตทริบิวต์และเมธอดเหมือนคลาสที่มันสืบทอดมาเรียกคลาสนั้นว่า super class หรือ base class ส่วนคลาสที่ได้รับการสืบทอดเรียกว่า sub class หรือ child class

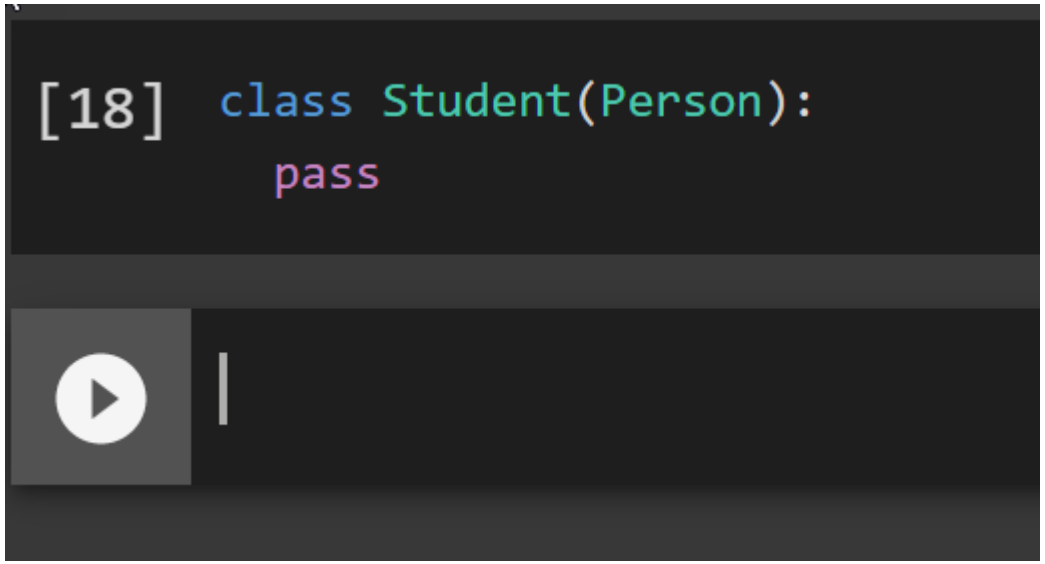
ตัวอย่าง ทำการสร้าง class ชื่อ person ประกอบด้วย คุณลักษณะ firstname และ lastname และมี printname method:

```
class Person:
  def __init__(self, fname, lname):
    self.firstname = fname
    self.lastname = lname

  def printname(self):
    print(self.firstname, self.lastname)


x = Person("เดช", "ธรรมศิริ")
x.printname()
```

```
เดช ธรรมศิริ
```

สร้าง class ลูกจาก class แม่ต้นฉบับ

- ```
  class Student(Person):
      pass
  ```


  ```

เพิ่ม object ให้แก่ class ลูกที่สร้างขึ้นมาใหม่

- ```
  x = Student("Dech", "Thammasiri")
  x.printname()
  ```

```
Dech Thammasiri
```
```

ขอบคุณครับ