

Lecture 16 Modules

17 ตุลาคม 2557 8:56

เราได้เรียนเกี่ยวกับการเก็บข้อมูลใน list การสร้าง function และการรวมกันของก้อนข้อมูลและฟังก์ชัน คือ object ในบทนี้ เราเรียนเกี่ยวกับการสร้าง โมดูล Module เราจะได้เรียกใช้โมดูล Pygame ในการวาดภาพกราฟฟิก

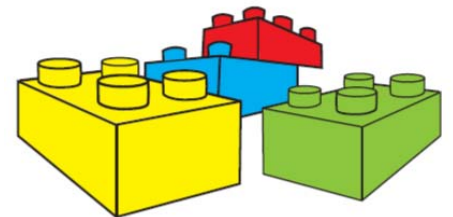
โมดูลคืออะไร

โมดูลคือชิ้นส่วนหนึ่งในการประกอบบางสิ่งบางอย่าง ชิ้นส่วนย่อยอย่าง Lego ที่มารวมกันแล้วสร้างเป็นชิ้นงานขึ้นมาได้ การเขียนโปรแกรมเราไม่สามารถที่จะเขียนโปรแกรมทั้งหมดในไฟล์เดียวแล้วให้ทำงานที่ใหญ่มาได้ เราจำเป็นต้องแยกไฟล์ออกเป็นหลายไฟล์แล้วนำมาเชื่อมต่อให้เป็นโปรแกรมเดียว เพื่อง่ายต่อการจัดการโปรแกรม การแยกเป็นส่วนเล็กจะทำให้เราค้นหาส่วนที่มีปัญหาได้เข้าใจง่ายยิ่งขึ้น และการแยกเป็นส่วนย่อย ยังช่วยให้เรานำกลับมาใช้ใหม่ได้ง่าย (reuseable) กับ การเขียนโปรแกรมใหม่ต่อไป

เราจะสร้างโมดูลอย่างไร

เราจะสร้างไฟล์ชื่อ myModule.py ที่ด้านในมีฟังก์ชันการคำนวณ องศาเซลเซียสเป็นฟาเรนไฮต์ ดังนี้

```
7% myModule.py - C:/Python33/ComLang/chapte...
File Edit Format Run Options Windows Help
#this is the file "myModule.py"
#we're going to use it in another program
def c_to_f(celsius):
    fahrenheit = celsius * 9 / 5 + 32
    return fahrenheit
Ln: 6 Col: 0
```



เราได้สร้างโมดูลชื่อ myModule และข้างในมีฟังก์ชันให้เรียกใช้คือ

การเรียกใช้โมดูลอย่างไร

การเรียกใช้โมดูล เข้ามาในโปรแกรม เราจะใช้คำสั่ง import ดังนี้

```
import myModule
```

เมื่อเรา import เราสามารถที่จะเรียกใช้ ฟังก์ชัน c_to_f() ที่ให้ส่งอุณหภูมิองศาเซลเซียสเข้าไป เราจะทดลองเขียนโปรแกรมใช้ดังนี้

```
import myModule

cel = float(input("Enter a temperature in Celsius: "))
fah = c_to_f(cel)
print("That's", fah, "degrees Fahrenheit")
```

เมื่อเราทดสอบรันผลลัพธ์ดู

```
>>>
Enter a temperature in Celsius: 100
Traceback (most recent call last):
  File "C:/Python33/ComLang/chapter 15/testMyModule.py", line 4, in <module>
    fah = c_to_f(cel)
NameError: name 'c_to_f' is not defined
^^^
```

จะเกิด error ขึ้น บอกว่า c_to_f() ไม่ได้ประกาศ แต่เรารู้ว่าเราได้พิมพ์ ฟังก์ชันนี้ในโมดูล myModule แล้ว ในภาษา Python เราจำเป็นต้องเพิ่มชื่อของโมดูลเข้าไปด้วยดังนี้ จากเดิม

```
fah = c_to_f(cel)
```

เป็นโค้ดดังนี้

```
fah = myModule.c_to_f(cel)
```

โดยเพิ่มเติมชื่อโมดูลแล้วจึงเรียกใช้ฟังก์ชันที่อยู่ในโมดูลอีกที

Namespaces เนมสเปซ

ทำไมต้องมีชื่อโมดูลไว้ด้านหน้าฟังก์ชันที่เราเรียกใช้ ทั้งที่เรา import โมดูล myModule ด้านบนโค้ดแล้ว นี่คือการจัดการของ namespaces ขอบเขตของชื่อ ต้องเข้าใจว่าการเขียนโปรแกรมอาจจะต้องมีหลายโมดูลเข้ามาเกี่ยวข้องในโปรแกรมของเรา และในบางโมดูลอาจมีชื่อของฟังก์ชันที่เหมือนกัน ในสองโมดูล ซึ่งอาจจะทำให้การทำงาน เกิดความสับสนหรือผิดพลาดได้เนื่องจากความเข้าใจผิด ดังนั้นเพื่อหลีกเลี่ยงปัญหาการใช้ชื่อฟังก์ชัน เหมือนกัน การใช้ namespaces หรือ

ในภาษา Python จะใช้ชื่อโมดูลไว้ด้านหน้า เนื่องจากชื่อโมดูลมีความเฉพาะเจาะจงมากกว่า การใช้ชื่อฟังก์ชัน ทำให้โปรแกรมแบ่งแยกโค้ดได้ว่า ฟังก์ชันนี้มาจากโมดูลไหน

```
fah = yourModule.c_to_f(cel)
```

ชื่อฟังก์ชันของสองโมดูลอาจจะเหมือนกัน แต่ใช้ชื่อโมดูล ด้านหน้า ที่แตกต่างกันทำให้ โปรแกรมสามารถแยกแยะได้ ทั้งนี้ การทำงานในฟังก์ชันของแต่ละโมดูลอาจจะแตกต่างกันด้วย

การนำเข้า Namespace

บางครั้งอาจเป็นการยุ่งยากที่เราจะเขียนชื่อของโมดูลไว้ด้านหน้าทุกครั้งที่เรียกใช้ในโปรแกรม และเราได้แยกแยะแล้วว่า โมดูลนี้ค่อนข้างใช้งานเฉพาะ ไม่มีการซ้ำกันเกินขึ้นแน่ เพื่ออำนวยความสะดวก แทนที่โค้ดเดิมเราจะ

```
import myModule
```

เราจะเจาะจง เข้าไปดังนี้

```
from myModule import c_to_f
```

เมื่อเราเรียกใช้เราสามารถเรียกได้โดยตรงโดยไม่มีชื่อโมดูลดังนี้

```
fah = c_to_f(cel)
```

แต่บางครั้งในโมดูลมีหลายสิบฟังก์ชันในนั้น เราคงจะไม่ import ทีละฟังก์ชัน จะเป็นการเสียเวลา เราสามารถที่จะเรียกใช้ ฟังก์ชันในโมดูลนั้นทั้งหมดดังนี้

```
from myModule import *
```

Standard Module โมดูลมาตรฐาน

เราจะสังเกตได้ว่าคำสั่งเฉพาะ ในภาษา Python อาทิเช่น print for if เราไม่จำเป็นต้อง import โมดูลใดๆ มาใช้ใน โปรแกรม เราสามารถเรียกใช้ได้โดยตรง แต่ Python เมื่อเราติดตั้ง จะมีโมดูลสำเร็จรูปที่เราสามารถเรียกใช้ได้ แต่จะต้อง import อาทิเช่น

- Time เวลา
- Random การสุ่ม
- Math คณิตศาสตร์

เราจะลองเรียกใช้โมดูลกัน ดังนี้

Time

คำสั่ง sleep() จะหยุดเวลาไว้กี่วินาที

```
import time
print("How")
time.sleep(2)
print("are")
time.sleep(2)
print("you")
time.sleep(2)
print("today?")
```

จะเห็นได้ว่าเราต้องมี time.sleep() แต่ถ้าต้องการใช้แค่ sleep()

```
>>> import time
>>> sleep(2)
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    sleep(2)
NameError: name 'sleep' is not defined
```

จะ Error เราจะแก้โดยใช้

```
>>> from time import sleep
>>> sleep(2)
```

Random การสุ่ม

การสุ่มจะใช้เยอะมากในการเขียนเกมส์ เพราะเราไม่ต้องการให้เกมส์ ออกมาแบบเดิมซ้ำๆ จะต้องมีการเปลี่ยนแปลงตลอด ด้วยวิธีการสุ่ม

การสุ่มตัวเลขจำนวนเต็ม ด้วย randint(a, b) จะระบุขอบเขต 0 - 100

```
>>> import random
>>> print(random.randint(0,100))
75
>>> print(random.randint(0,100))
55
```

การสุ่มทศนิยม ด้วย random() จะได้ค่าระหว่างช่วง 0-1

```
>>> print( random.random() )
0.19043437890294068
>>> print( random.random() )
0.7595996779009125
```

ถ้าเราอยากได้ค่าทศนิยมระหว่าง 0-10 จะต้องทำอย่างไร

```
>>> print( random.random()*10)
5.6288289624545795
>>> print( random.random()*10)
8.327135249991537
```

บทบทวน

- โมดูลคืออะไร
- สร้างโมดูลอย่างไร
- เราจะใช้โมดูลในโปรแกรมอย่างไร
- Namespaces คืออะไร
- การใช้ from เพื่อนำ namespaces เข้ามาในโปรแกรม

ทดสอบความรู้

1. จงบอกข้อดีของการใช้โมดูล
2. จงยกตัวอย่างโค้ดการเขียนโมดูล
3. คำสั่งอะไร นำเข้าโมดูลในโปรแกรม
4. เราจะนำเข้าโมดูล time ทุกอย่างที่มีได้อย่างไร (ทุก function, variable และ objects)

จงเขียนโปรแกรมต่อไปนี้

1. จงสร้างโมดูลที่สามารถพิมพ์ชื่อตัวเองเป็นตัวหนังสือใหญ่ๆ (lecture 14 ข้อหนึ่ง) และเรียกใช้โมดูลและฟังก์ชันนั้น
2. จงแก้โค้ดในตัวอย่างให้สามารถใช้คำสั่งในโมดูลแบบนี้ได้

```
fah = c_to_f(cel)
```
3. สร้างโปรแกรมที่สร้าง list ของตัวเลขที่จำนวนเต็มที่มีการสุ่มค่าตั้งแต่ 1 - 20
4. สร้างโปรแกรมที่มีการสุ่มตัวเลขทศนิยม ทุกๆ 3 วินาที เป็นเวลารวมทั้งหมด 30 วินาที