

Basic function Lab

Recap: function call

- การเรียกใช้ฟังก์ชัน ทำได้สองแบบ
- แบบแรก คือเรียกชื่อฟังก์ชันโดยตรง เช่น
 - `len(string)` คือการหาความยาวของสตริง
 - ฟังก์ชันที่ถูกนิยามแบบนี้ สามารถกระทำกับ object ได้หลายชนิด
- แบบที่สอง เรียกฟังก์ชันด้วย dot operator
 - `string.upper()`
 - ฟังก์ชันที่ถูกนิยามแบบนี้ จะสามารถกระทำเฉพาะกับ object นั้นๆ
 - ฟังก์ชัน `upper()` คงไม่เหมาะกับ object ประเภท ตัวเลข

Recep: module

- เราสามารถเรียกใช้ฟังก์ชันใน module ได้โดยใช้ keyword

```
from module_name import function_name
```

- หรือหากต้องการเรียกใช้ฟังก์ชันทุกตัวใน module นั้น

```
from module_name import *
```

หรือ

```
import module_name
```

} สองวิธีนี้มีความแตกต่างกัน
เรียกใช้งานฟังก์ชันอย่างไร ?

- หากต้องการเปลี่ยนชื่อฟังก์ชัน ให้ใช้ keyword: as

```
from math import log as mylog
```

Experiment 1

● ทดลองการเรียกใช้ module

1.

```
>>> import math  
>>> math.log(2)
```

2.

```
>>> import math as maths  
>>> math.log(2)  
>>> maths.log(2)
```

3.

```
>>> from math import log  
>>> math.log(2)  
>>> log(2)  
>>> exp(2)
```

4.

```
>>> from math import log as mylog  
>>> math.mylog(2)  
>>> mylog(2)
```

5.

```
>>> from math import *  
>>> exp(2)  
>>> log(2)
```

String manipulation

- Python มี built-in ฟังก์ชัน สำหรับการจัดการ object ประเภท string มากมาย
- หาข้อมูลเพิ่มเติมได้จาก

<https://docs.python.org/3/library/stdtypes.html#string-methods>

String manipulation: str.count()

- Syntax: str.count(substring)
- นับจำนวน substring ใน สตริง str ว่ามีกี่ตัว
- ตัวอย่าง

```
>>> s = 'dog dog dog'  
>>> s.count('dog')  
3
```

```
>>> s = 'fish cat fish fish'  
>>> s.count('fish')  
3  
>>> s.count('cat')  
1
```

String manipulation: str.find()

- Syntax: str.find(substring)
- หาดำแหน่งของ substring ตัวแรกสุดที่เจอ
- ตัวอย่าง

```
>>> s = 'fish cat fish fish'
>>> s.find('cat')
5
>>> s.find('fish')
0
```

String manipulation: checking utilities

- `str.isalpha()`

- ตรวจสอบว่า character ทุกตัวในสตริงเป็น alphabet หรือไม่

- `str.isdigit()`

- ตรวจสอบว่า character ทุกตัวในสตริงเป็น ตัวเลข หรือไม่

- `str.upper()`

- ปรับ `str` ให้เป็นอักษรตัวใหญ่ (uppercase) ให้หมด

- `str.lower()`

- ปรับ `str` ให้เป็นอักษรตัวเล็ก (lowercase) ให้หมด

Experiment 2

● ทดลองการเรียกใช้ checking utilities

1.

```
>>> str = 'alpha'  
>>> str.isalpha()
```

2.

```
>>> str = 'alhab3t'  
>>> str.isalpha()
```

3.

```
>>> str = '12345'  
>>> str.isdigit()
```

4.

```
>>> 'alpha'.isalpha()  
>>> '12345'.isalpha()  
>>> '12345'.isdigit()
```

5.

```
>>> str = '_myvar'  
>>> str.upper()  
>>> '123python'.upper()
```

6.

```
>>> str = 'if'  
>>> str.lower()
```

A tip for exam study

- ถ้าเราต้องการจะตรวจสอบว่า identifier เป็น reserved keyword หรือไม่ จะตรวจสอบอย่างไร
- ใช้ function `iskeyword()` ใน module `keyword`

```
>>> from keyword import iskeyword
>>> s = 'if'
>>> iskeyword(s)
True
>>> s = 'iff'
>>> iskeyword(s)
False
```

Exercise 1

- จงออกแบบโปรแกรมเพื่อรับข้อมูลเป็น string แล้วแสดงผล string นั้นในรูปตัวพิมพ์ใหญ่ทั้งหมด หาก string มีความยาวเกิน 15 หากมีความยาวน้อยกว่าหรือเท่ากับ 15 ให้แสดง string ด้วยตัวพิมพ์เล็กทั้งหมด

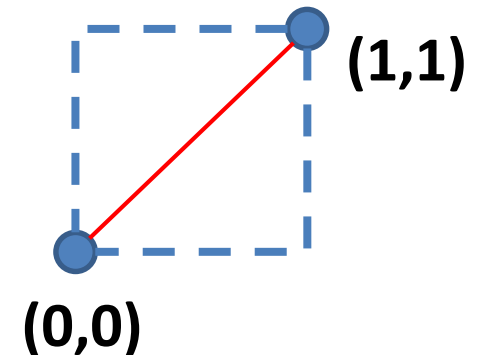
```
[jkm@ludwig ~]$ python xx.py
Please input string: This is short
this is short
[jkm@ludwig ~]$ python xx.py
Please input string: This is loooooooooong
THIS IS LOOOOOOOOONG
[jkm@ludwig ~]$ |
```

Exercise 2

- จงเขียนโปรแกรมเพื่อรับข้อมูลในรูปจุดคู่อันดับสองจุด ในรูปแบบ x_1, y_1, x_2, y_2 เข้ามา เพื่อคำนวณหาพื้นที่สี่เหลี่ยมจัตุรัสที่มีความยาวเส้นทแยงมุมเท่ากับ ระยะทางระหว่างจุด (x_1, y_1) ถึงจุด (x_2, y_2)

- ตัวอย่าง

```
[jkm@ludwig ~]$ python circle.py
Input coordinates: 1,1,0,0
area is 1.00
[jkm@ludwig ~]$ python circle.py
Input coordinates: -1,-1,0,0
area is 1.00
```



Exercise 3 (Homework)

- จงออกแบบโปรแกรมเพื่อรับ email address เป็นข้อมูลขาเข้า เพื่อตรวจสอบว่า email นั้น เป็น email ของ มหาวิทยาลัยเชียงใหม่ หรือไม่ แล้วแสดงผลว่า Valid หรือ Invalid ออกทางหน้าจอ
- กำหนดให้ email ที่ valid จะมีลักษณะดังต่อไปนี้
 - เป็น email ที่มีเครื่องหมาย @ ตัวเดียว
 - เครื่องหมาย @ ต้องไม่ใช่ตัวแรกในชื่อ email
 - หลังจากเครื่องหมาย @ จะต้องตามด้วย domain name (cmu.ac.th)
 - ชื่อ Domain name เป็นแบบ case insensitive นั่นคือ
 - Cmu.ac.th หรือ cmu.ac.th หรือ CMU.AC.TH ถือว่าเป็นตัวเดียวกันใช้แทนกันได้

ตัวอย่าง exercise 2

```
[jkm@ludwig ~]$ python mailcheck.py
Please input email: jakramate@cmu.ac.th
Valid
[jkm@ludwig ~]$ python mailcheck.py
Please input email: _jakra@cmu.ac.th
Valid
[jkm@ludwig ~]$ python mailcheck.py
Please input email: j@krA@cmu.ac.th
Invalid
[jkm@ludwig ~]$ python mailcheck.py
Please input email: @@@@
Invalid
[jkm@ludwig ~]$ python mailcheck.py
Please input email: jakra@CMU.AC.TH
Valid
[jkm@ludwig ~]$ python mailcheck.py
Please input email: jakra@cmu.com
Invalid
```