

Basic function Lab

Recap: function call

- การเรียกใช้ฟังก์ชัน ทำได้สองแบบ
- แบบแรก คือเรียกชื่อฟังก์ชันโดยตรง เช่น
 - `len(string)` คือการหาความยาวของสตริง
 - ฟังก์ชันที่ถูกระบุแบบนี้ สามารถกระทำกับ object หรือตัวแปร ได้หลายชนิด
- แบบที่สอง เรียกฟังก์ชันด้วย dot operator
 - `string.upper()`
 - ฟังก์ชันที่ถูกระบุแบบนี้ จะสามารถกระทำเฉพาะกับ object หรือตัวแปร 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 หรือตัวแปรชนิดสตริงที่ต้องการว่ามีกี่ตัว
- ตัวอย่างเช่น

```
>>> 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 ในตัวแปรชนิดสตริง ที่ต้องการ (** โดยจะเริ่มนับจากตำแหน่ง 0)
- ตัวอย่างเช่น



The diagram shows two red boxes with arrows pointing to the string 'fish cat fish fish'. The first box is labeled 'ตำแหน่ง 0' and points to the first 'f' of the first 'fish'. The second box is labeled 'ตำแหน่ง 5' and points to the first 'f' of the second 'fish'.

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

Now !!!

String manipulation: checking utilities

- `str.isalpha()`

- ตรวจสอบว่า character ทุกตัวในสตริงเป็น alphabet หรือไม่

- `str.isdigit()`

- ตรวจสอบว่า character ทุกตัวในสตริงเป็น ตัวเลข หรือไม่

- `str.upper()`

- ปรับ `str` ให้เป็นอักษรตัวใหญ่ (**Uppercase** ~~lowercase~~) ให้หมด

- `str.lower()`

- ปรับ `str` ให้เป็นอักษรตัวเล็ก (~~lowercase~~) ให้หมด

Experiment 2

- ทดลองการเรียกใช้ checking utilities

1.

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

2.

```
>>> str = 'alphan3t'  
>>> 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 หรือตัวแปร 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 ให้แสดงผล 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 ~]$ |
```

Hint for Exercise 1

- ฟังก์ชันที่เกี่ยวข้องที่นำมาใช้แก้ปัญหาคือ 1 นี้
- Built-in ฟังก์ชัน
 - **len()** → นับความยาวของสตริง
 - **ตัวแปรสตริง.upper()** → ทำให้สตริงเป็นตัวพิมพ์ใหญ่
 - **ตัวแปรสตริง.lower()** → ทำให้สตริงเป็นตัวพิมพ์เล็ก

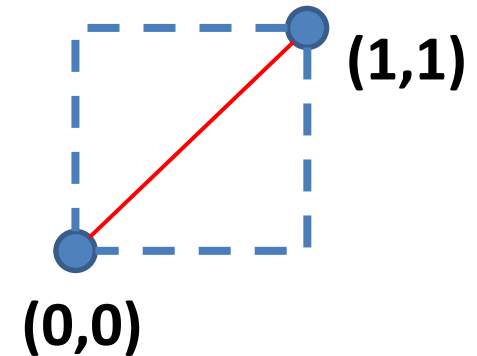
Exercise 2

- จงเขียนโปรแกรมเพื่อรับข้อมูลในรูปแบบจุดคู่อันดับสองจุด ในรูปแบบ x_1, y_1, x_2, y_2 เข้ามา เพื่อคำนวณหาพื้นที่สี่เหลี่ยมจัตุรัสซึ่งมีความยาวเส้นทแยงมุมเท่ากับ ระยะทางระหว่างจุด (x_1, y_1) ถึงจุด (x_2, y_2)

$$\text{พท.สี่เหลี่ยมด้านเท่า} = \frac{1}{2} * (\text{เส้นทแยงมุม})^2$$

- 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
```



Hint for Exercise 2

- ฟังก์ชันที่เกี่ยวข้องที่นำมาใช้แก้ปัญหาคือ 2 นี้

- Built-in ฟังก์ชัน

token



- ตัวแปรสตริง `split('สัญลักษณ์ที่ใช้แยก')` → แยก substring ที่รับเข้ามาโดยใช้สัญลักษณ์ที่กำหนดไว้ ผลลัพธ์ที่แยกแต่ละตัวได้เก็บไว้ในตัวแปรชุด (array) แต่ละช่อง
เช่น `s_in = 'hello,python,mid-exam'`
`s_out = s_in.split(',')`

ตัวแปรชุด `s_out`

`'hello'`

`'python'`

`'mid-exam'`

`s_out[0]`

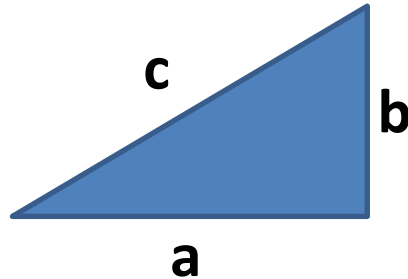
`s_out[1]`

`s_out[2]`

- ฟังก์ชัน `float()`

Hint for Exercise 2

- ความรู้เกี่ยวกับสามเหลี่ยมมุมฉาก



$$c^2 = a^2 + b^2$$

$$c = \sqrt{a^2 + b^2}$$

- ต้องการใช้ฟังก์ชัน square root มาใช้ในการคำนวณ!!
- ฟังก์ชัน `sqrt()` คือฟังก์ชันหาค่า square root ที่อยู่ในโมดูล **math**
- ต้องประกาศเรียกใช้งานฟังก์ชันนี้ก่อนใช้งานในการคำนวณ

```
from math import sqrt
```

ตัวอย่างเช่น `c = sqrt(a**2 + b**2)`

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 3

```
[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
```

Hint for Exercise 3

- ฟังก์ชันที่เกี่ยวข้องที่นำมาใช้แก้ปัญหาข้อ 3 นี้
- Built-in ฟังก์ชันที่เกี่ยวกับการจัดการสตริงหรือข้อมูลสายอักขระ
 - ตัวแปรชนิดสตริง.count()
 - ตัวแปรชนิดสตริง.find()
 - ตัวแปรชนิดสตริง.split()
- การเปรียบเทียบสตริง
 - ตัวอย่างเช่น `a_in = 'Worawut.Sri'`
`a_in == 'Worawut.Srisukkham'` ผลลัพธ์คือ **False**

Hint for Exercise 3

- ฟังก์ชันที่เกี่ยวข้องที่นำมาใช้แก้ปัญหาคือ 3 นี้

- Built-in ฟังก์ชัน

token

- ตัวแปรสตริง `split('สัญลักษณ์ที่ใช้แยก')` → แยก substring ที่รับเข้ามาโดยใช้สัญลักษณ์ที่กำหนดไว้ ผลลัพธ์ที่แยกแต่ละตัวได้เก็บไว้ในตัวแปรชุด (array) แต่ละช่อง

เช่น `s_in = 'hello,python,mid-exam'`
`s_out = s_in.split(',')`

ตัวแปรชุด `s_out`

<code>'hello'</code>	<code>'python'</code>	<code>'mid-exam'</code>
<code>s_out[0]</code>	<code>s_out[1]</code>	<code>s_out[2]</code>