

## Lab 3 - ฟังก์ชันและโปรแกรมย่อย

### 1. ฟังก์ชัน (Functions)

ฟังก์ชันเปรียบเสมือนคำสั่งพิเศษที่ช่วยให้เราสามารถทำงานเฉพาะเจาะจงหนึ่ง ๆ ได้อาทิเช่น ฟังก์ชัน print() ใช้ในการแสดงข้อความออกมาทางหน้าจอ ฟังก์ชัน input() ใช้ในการรับข้อมูลชนิดข้อความจากผู้ใช้ผ่านแป้นพิมพ์ ฟังก์ชัน int() ใช้ในการแปลงข้อมูลใด ๆ ให้เป็นข้อมูลชนิดจำนวนเต็ม เป็นต้น

อย่างไรก็ตามภาษา Python ไม่อาจสร้างฟังก์ชันทุกอย่างให้เพียงพอต่อความต้องการของนักเขียนโปรแกรมทั้งโลกได้ เช่นถ้าเราต้องการใช้ฟังก์ชันที่เฉพาะเจาะจงมาก ๆ อาทิ ฟังก์ชันตรวจสอบจำนวนเฉพาะ ฟังก์ชันหา ค.ร.น. ของตัวเลข ภาษา Python อาจไม่ได้สร้างฟังก์ชันเฉพาะเจาะจงให้เราไว้ แต่ภาษา Python ก็อนุญาตให้นักเขียนโปรแกรมสร้างฟังก์ชันขึ้นมาใหม่ได้ โดยนิยามลักษณะและความสามารถต่าง ๆ ของฟังก์ชันที่เราต้องการไว้ก่อนที่จะใช้เสมอ ดังนี้

```
def ชื่อฟังก์ชัน(พารามิเตอร์) :  
    การทำงานต่าง ๆ ของฟังก์ชัน  
    return ค่าที่ต้องการคืนกลับ (ถ้ามี)
```

โดยที่

- พารามิเตอร์คือค่าใด ๆ ที่จะถูกส่งเข้ามาเก็บไว้ในตัวแปรนี้เพื่อใช้เป็นข้อมูลสำหรับคำนวณในฟังก์ชันต่อไป ฟังก์ชันใด ๆ อาจมีหรือไม่มีพารามิเตอร์ก็ได้
- การทำงานต่าง ๆ ของฟังก์ชัน คือส่วนที่เราเขียนโปรแกรมให้ฟังก์ชันทำงานตามที่เรต้องการ
- ค่าที่ต้องการคืนกลับ คือค่าที่ฟังก์ชันจะส่งกลับไปให้ส่วนที่เรียกฟังก์ชันนี้ อาจจะมีหรือไม่มีก็ได้ ขึ้นอยู่กับการทำงานของฟังก์ชันนั้น ๆ ว่าต้องการส่งค่ากลับหรือไม่

**ตัวอย่างที่ 1.1 :** ตัวอย่างการนิยามฟังก์ชัน compute\_area() ทำหน้าที่คำนวณพื้นที่รูปสามเหลี่ยมเมื่อทราบความยาวฐานและความสูง

```
def compute_area(base, height) :  
    area = 0.5*base*height  
    return area
```

เมื่อนิยามฟังก์ชัน<sup>1</sup> เรียบร้อยเราก็จะสามารถนำฟังก์ชันนั้น ๆ ไปใช้งานได้บ่อยครั้งตามที่เรต้องการโดยการเรียกใช้ฟังก์ชันดังกล่าว ดังตัวอย่าง

**ตัวอย่างที่ 1.2 :** การเรียกใช้ฟังก์ชัน compute\_area()

```
# นิยามฟังก์ชัน compute_area()  
def compute_area(base, height):  
    area = 0.5*base*height  
    return area  
  
#แสดงพื้นที่ของรูปสามเหลี่ยมที่มีความยาวฐานเท่ากับ 5 และความสูงเท่ากับ 3  
print(compute_area(5, 3))  
  
#แสดงพื้นที่ของรูปสามเหลี่ยมที่มีความยาวฐานเท่ากับ 10.5 และความสูงเท่ากับ 3.8  
print(compute_area(10.5, 3.8))  
  
#คำนวณพื้นที่ของรูปสามเหลี่ยมที่มีความยาวฐานเท่ากับ 8 และความสูงเท่ากับ 6 เก็บใส่ตัวแปร A  
A = compute_area(8, 6)  
print("Price per area is $5.")  
print("Your price is", A*5)
```

<sup>1</sup> หมายเหตุ ฟังก์ชัน compute\_area() ในตัวอย่างที่ 1.1 อาจเขียนย่อได้อีกแบบดังนี้

```
def compute_area(base, height) :  
    return 0.5*base*height
```

**แบบฝึกหัดที่ 1.1 : ฟังก์ชันกำลังสอง**

ให้นักเรียนนิยามฟังก์ชัน square ที่รับจำนวนเต็ม x เป็นพารามิเตอร์แล้วคืนค่า  $x^2$  ให้สมบูรณ์

โปรแกรมที่กำหนดให้ (เติมโปรแกรมในช่องว่างให้สมบูรณ์)	ตัวอย่างการทำงาน
<pre>import math  def square(x):     _____     _____  x = int(input("Input x: ")) print ("x^2 = %.0f" % square(x))</pre>	<p>ตัวอย่างที่ 1 Input x: <u>2</u> <math>x^2 = 4</math></p> <hr/> <p>ตัวอย่างที่ 2 Input x: <u>11</u> <math>x^2 = 121</math></p>

**แบบฝึกหัดที่ 1.2 : ฟังก์ชันพหุนาม**

ให้เขียนฟังก์ชัน poly ที่รับอาร์กิวเมนต์ x ที่คำนวณค่าของฟังก์ชัน  $f(x) = 5x^2 - 121x + 7$

โปรแกรมที่กำหนดให้ (เติมโปรแกรมในช่องว่างให้สมบูรณ์)	ตัวอย่างการทำงาน
<pre>import math  _____ _____ _____  x = int(input("Input x: ")) a = poly(x) print ("f(x) =", a)</pre>	<p>ตัวอย่างที่ 1 Input x: <u>4</u> <math>f(x) = -397</math></p> <hr/> <p>ตัวอย่างที่ 2 Input x: <u>15</u> <math>f(x) = -683</math></p>

**แบบฝึกหัดที่ 1.3 : ปริมาตรทรงกระบอก**

ปริมาตรของทรงกระบอกใด ๆ สามารถคำนวณได้จากสูตร  $V = \pi \cdot r^2 \cdot h$  โดยที่ r คือรัศมีที่ฐานของทรงกระบอก และ h คือความสูงของทรงกระบอก

จงนิยามฟังก์ชัน find\_cylinder\_volume() ที่รับความสูงและรัศมีเป็นพารามิเตอร์ตามลำดับแล้วคืนปริมาตรของทรงกระบอกที่สอดคล้องกับสูตรข้างต้นในบรรทัดที่ 3 - 5 พร้อมทั้งเรียกใช้ฟังก์ชันดังกล่าวในบรรทัดที่ 9 เพื่อให้โปรแกรมทำงานสอดคล้องกับตัวอย่างที่กำหนดให้

โปรแกรมที่กำหนดให้ (เติมโปรแกรมในช่องว่างให้สมบูรณ์)	ตัวอย่างการทำงาน
<pre>1: import math 2: 3: def find_cylinder_volume(_____) 4: _____ 5: _____ 6: 7: r = float(input("radius = ")) 8: h = float(input("height = ")) 9: V = _____ 10: print("Volume = %.2f" %V) 11:</pre>	<p>ตัวอย่างที่ 1 radius = <u>3</u> height = <u>5</u> Volume = 141.37</p> <hr/> <p>ตัวอย่างที่ 2 radius = <u>4.2</u> height = <u>1.5</u> Volume = 83.13</p>

### แบบฝึกหัดที่ 1.4 : คำนวณระยะทางการเคลื่อนที่

ให้นักเรียนนิยามฟังก์ชัน distance() สำหรับคำนวณการเคลื่อนที่ โดยรับพารามิเตอร์ 2 ตัว คือ a (ความเร่งในหน่วยเมตรต่อวินาทีกำลังสอง) และ t (เวลาในหน่วยวินาที) โดยกำหนดให้อัตราเร็วเริ่มต้นเป็น 0 ในบรรทัดที่ 1 - 3 แล้วเรียกใช้ในบรรทัดที่ 7 เพื่อให้โปรแกรมทำงานสอดคล้องกับตัวอย่างที่กำหนดให้

	โปรแกรมที่กำหนดให้ (เติมโปรแกรมในช่องว่างให้สมบูรณ์)	ตัวอย่างการทำงาน
1: 2: 3: 4: 5: 6: 7:	<pre>_____ _____ _____ _____ acceleration = int(input("Acceleration : ")) time = int(input("Time : ")) print(_____)</pre>	ตัวอย่างที่ 1 Acceleration : <u>9</u> Time : <u>10</u> 450.0 <hr/> ตัวอย่างที่ 2 Acceleration : <u>4</u> Time : <u>16</u> 512.0

### แบบฝึกหัดที่ 1.5 : เมื่อไหร่ถึงน้ำจะเต็ม

เขียนโปรแกรมเพื่อคำนวณเวลาในการเติมน้ำลงถังทรงกระบอกที่ว่างเปล่า โดยถามข้อมูลผู้ใช้ดังนี้

- เส้นผ่านศูนย์กลางของถัง หน่วยเป็นนิ้ว
- ความสูงของถัง หน่วยเป็นนิ้ว
- อัตราการไหลของน้ำ หน่วยเป็นซีซีต่อวินาที

จากนั้นแสดงเวลาในการเติมน้ำให้เต็มถึง ในหน่วย "นาที"

#### ตัวอย่างการทำงาน

```
Enter tank's diameter (inch): 24
Enter tank's height (inch): 3
Enter water flow rate (cc/s): 15
Time to fill the tank is 24.711110 minutes
```

#### โปรแกรมที่เขียนได้คือ

## 2. ฝึกเขียนโปรแกรมแสนสนุก - ปัญหาการหาค่ารากสมการกำลังสอง

เราต้องการหาคำตอบของรูปแบบหนึ่งของสมการกำลังสอง พิจารณาสมการ

$$ax^2 + bx + c = 0$$

เราทราบว่าคำตอบ x สองคำตอบที่เป็นรากของสมการกำลังสอง คือ

$$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{และ} \quad x = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \quad \text{โดยที่ } b^2 - 4ac > 0$$

เพื่อหาคำตอบของสมการกำลังสองนี้เราจะแบ่งงานพัฒนาโปรแกรมนี้ออกเป็นหลาย ๆ ส่วน

### แบบฝึกหัดที่ 2.1 : รากสมการกำลังสองภาคที่หนึ่ง (แสดงพหุนาม)

ให้เขียนฟังก์ชัน show\_poly(a, b, c) ที่รับพารามิเตอร์ 3 ตัวคือ a, b, และ c จากนั้นให้แสดงผลในรูปแบบตามตัวอย่างดังนี้

ตัวอย่างการเรียกใช้ฟังก์ชัน	การแสดงผลทางหน้าจอ
show_poly(1, 2, 3)	1 x^2 + 2 x + 3
show_poly(5, 0, -10)	5 x^2 + 0 x + -10

#### ฟังก์ชัน show\_poly(a, b, c) ที่เขียนได้คือ

```
def show_poly(a, b, c):
```

### แบบฝึกหัดที่ 2.2 : รากสมการกำลังสองภาคที่สอง (คำนวณเทอมย่อย)

สังเกตว่าในการหาราก เราจำเป็นต้องคำนวณค่า  $\sqrt{b^2 - 4ac}$  หลายครั้ง ในส่วนนี้เราจะเขียนฟังก์ชันเพื่อคำนวณส่วนดังกล่าว โดยให้นักเรียนเขียนฟังก์ชันชื่อ cal\_term ที่รับพารามิเตอร์สามตัวคือ a, b และ c แทนสัมประสิทธิ์ต่าง ๆ ของ x ในพหุนาม

ให้ฟังก์ชันนี้คืนค่า  $\sqrt{b^2 - 4ac}$  ดังตัวอย่าง

ตัวอย่างการเรียกใช้ฟังก์ชัน	การแสดงผลทางหน้าจอ
print( cal_term(1,2,1) )	0.0
print( cal_term(1,4,1) )	3.46410161514

#### ฟังก์ชัน cal\_term ที่เขียนได้คือ

```
import math
def cal_term _____
_____
_____
```

### แบบฝึกหัดที่ 2.3 : รากสมการกำลังสองภาคที่สาม (คำนวณหารากที่สอง)

เขียนฟังก์ชัน root\_plus(a,b,c) ที่รับสัมประสิทธิ์ a,b, และ c แล้วคำนวณค่า x เมื่อ

$$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

และเขียนฟังก์ชัน root\_minus(a,b,c) ที่รับสัมประสิทธิ์ a,b, และ c แล้วคำนวณค่า x เมื่อ

$$x = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

โดยฟังก์ชันทั้งสองให้เรียกใช้ฟังก์ชัน cal\_term ที่เขียนไว้ในแบบฝึกหัดที่ 2.2 ด้วย

#### ฟังก์ชัน root\_plus และ root\_minus ที่เขียนได้คือ

```
import math
```

**แบบฝึกหัดที่ 2.4 : รากสมการกำลังสองภาคที่สาม (ส่วนโปรแกรมหลัก)**

ให้เขียนโปรแกรมรับสัมประสิทธิ์ของสมการกำลังสอง จากนั้นแสดงรากของสมการดังกล่าว ให้แสดงโดยจัดรูปแบบเป็นทศนิยม 3 ตำแหน่ง โดยพิมพ์คำตอบที่ได้จาก root\_plus(a,b,c) ก่อน ทั้งนี้ให้ใช้ฟังก์ชันต่าง ๆ ที่เขียนไว้ในแบบฝึกหัดที่ 2.1 – 2.4 ให้เกิดประโยชน์

โปรแกรมที่เขียนได้คือ	ตัวอย่างการทำงาน
	<pre>Enter a: 1 Enter b: 2 Enter c: 1 The solutions of 1 x^2 + 2 x + 1 are -1.000 and -1.000</pre>

**3. ฝึกเขียนโปรแกรมแสนสนุก - ปัญหาการเคลื่อนที่แบบโพรเจกไทล์**

ลูกกระสุนปืนใหญ่ถูกยิงจากปืนที่กระบอกทำมุม  $\theta$  องศา กับพื้นระนาบ ด้วยความเร็วต้น  $u$  เมตรต่อวินาที เป้าหมายของกระสุนปืนคือกำแพงที่อยู่ห่างไป  $m$  เมตร ให้เขียนโปรแกรมคำนวณว่าเมื่อกระสุนเคลื่อนที่ถึงระยะดังกล่าว จะมีความสูงจากพื้นระนาบเท่าใด



**แบบฝึกหัดที่ 3.1 : การเคลื่อนที่แบบโพรเจกไทล์ภาคที่หนึ่ง (แปลงมุม)**

เนื่องจากมุมที่ใช้ในการรับข้อมูลเข้าเป็นหน่วยองศา แต่มุมที่นำไปคำนวณเพื่อหาคำตอบผ่านฟังก์ชัน math ต่าง ๆ มักจะเป็นหน่วยเรเดียน เราจึงจำเป็นต้องแปลงมุมที่รับเข้ามาจากหน่วยองศาให้เป็นเรเดียนเสียก่อน ฉะนั้นให้นักเรียนเขียนฟังก์ชัน deg\_to\_rad(d) ที่คืนค่ามุมในหน่วยเรเดียน จากมุม d ในหน่วยองศา

ตัวอย่างการเรียกใช้ฟังก์ชัน	การแสดงผลทางหน้าจอ
<code>print( deg_to_rad(30) )</code>	0.523598775598
<code>print( deg_to_rad(90) )</code>	1.57079632679

ฟังก์ชัน deg_to_rad ที่เขียนได้คือ



ฟังก์ชัน height ที่เขียนได้คือ

แบบฝึกหัดที่ 3.5 : การเคลื่อนที่แบบโพรเจกไทล์ภาคที่ห้า (ส่วนโปรแกรมหลัก)

ลูกกระสุนปืนใหญ่ถูกยิงจากปืนที่กระบอกทำมุม  $\theta$  องศา กับพื้นระนาบ ด้วยความเร็วต้น  $u$  เมตรต่อวินาที เป้าหมายของกระสุนปืนคือกำแพงที่อยู่ห่างไป  $m$  เมตร ให้เขียนโปรแกรมคำนวณว่าเมื่อกระสุนเคลื่อนที่ถึงระยะดังกล่าว จะมีความสูงจากพื้นระนาบเท่าใด ทั้งนี้อย่าลืมนำฟังก์ชันที่เขียนไว้ในแบบฝึกหัดที่ 3.1 - 3.4 มาใช้ให้เกิดประโยชน์



ตัวอย่างการทำงานที่ 1	Enter u: <u>20</u> Enter angle: <u>60</u> Enter distance: <u>15</u> The height of the cannon ball is 14.945 meters
ตัวอย่างการทำงานที่ 2	Enter u: <u>12</u> Enter angle: <u>45</u> Enter distance: <u>10</u> The height of the cannon ball is 3.188 meters

ส่วนของโปรแกรมหลักที่เขียนได้คือ