|  |
| --- |
| **Worksheet 6** |

นักเรียนจะได้เรียนรู้เกี่ยวกับฟังก์ชันในภาษา Python เราจะพูดถึงการสร้างและการใช้งานฟังก์ชันในเบื้องต้น และการเรียกใช้งานฟังก์ชันในรูปแบบต่างๆ

**การสร้างฟังก์ชันในภาษา Python**

ฟังก์ชัน (Function) คือส่วนของโค้ดหรือโปรแกรมที่ทำงานเพื่อวัตถุประสงค์บางอย่าง ในภาษา Python เราสามารถสร้างฟังก์ชันของเราเองเพื่อให้ทำงานที่ต้องการ ในการเขียนโปรแกรมเรามักจะแยกโค้ดที่มีการทำงานเหมือน ๆ กันเป็นฟังก์ชันเอาไว้ และเรียกใช้ฟังก์ชันนั้นซ้ำ ๆ ซึ่งเป็นแนวคิดของการ reuse โค้ด นี่เป็นรูปแบบของการประกาศฟังก์ชันในภาษา Python ตัวอย่างรูปแบบการเขียนเช่น

def function\_name(args...):  
 # statements  
  
def function\_name(args...):  
 # statements  
 return value

ในรูปแบบของการประกาศฟังก์ชันในภาษา Python นั้นจะใช้คำสั่ง def และหลังจากนั้น function\_name เป็นชื่อของฟังก์ชัน และในวงเล็บ ( ) เป็นการกำหนดพารามิเตอร์ของฟังก์ชัน พารามิเตอร์ของฟังก์ชันนั้นสามารถมีจำนวนเท่าไหร่ก็ได้หรือไม่มีก็ได้ และเช่นเดียวกับภาษาอื่นๆ ฟังก์ชันอาจจะมีหรือไม่มีการส่งค่ากลับ ต่อไปมาดูตัวอย่างการประกาศและใช้งานฟังก์ชันในภาษา Python

def mypersonalinfo():

print('My name is Trump Tonga')

print('Address: Neverland')

def hello(name):  
 print('Hello %s' % name)  
  
def count\_vowel(str):

vowel = 0  
 for c in str:  
 if c in ('A', 'E', 'I', 'O', 'U', 'a', 'e', 'i', 'o' 'u'):  
 vowel = vowel + 1  
 return vowel  
  
def area(width, height):  
 result = width \* height  
 return result

**การเรียกใช้งานฟังก์ชันในภาษา Python**

หลังจากเราได้สร้างฟังก์ชันในตัวอย่างก่อนหน้าแล้ว ต่อไปเราจะมาเรียกใช้งานฟังก์ชันเหล่านั้น ในการเรียกใช้ฟังก์ชันนั้นเราจะใช้ชื่อของฟังก์ชันและส่งอาร์กิวเมนต์ (arguments) ให้สอดคล้องกับพารามิเตอร์ (parameters) ที่กำหนดไว้ในฟังก์ชัน ดังนั้นอาร์กิวเมนต์คือค่าที่ส่งเข้าไปในฟังก์ชันตอนใช้งาน ส่วนพารามิเตอร์นั้นคือตัวแปรทีกำหนดไว้ในฟังก์ชันเพื่อรับค่าจากอาร์กิวเมนต์ มาดูตัวอย่างการเรียกใช้งานฟังก์ชันในภาษา Python (จากโค้ดตัวอย่างด้านบน)

# calling functions

mypersonalinfo()  
hello('Danny')  
hello('Mateo')  
print('Vowel in string = %d' % count\_vowel('marcuscode.com'))  
print('Vowel in string = %d' % count\_vowel('Python'))  
print('Area = %d' % area(8, 4))

ผลที่ได้คือ

My name is Trump Tonga

Address: Neverland

Hello Danny  
Hello Mateo  
Vowel in string = 5  
Vowel in string = 1  
Area = 32

def area(width, height):  
 result = width \* height  
 return result

result = area(4, 5)

print(‘The area is’, result)

ตัวอย่างนี้ฟังก์ชัน area( ) ส่งค่ากลับ (ผ่าน return) มายังตัวแปรที่ชื่อว่า result และสั่งพิมพ์ค่านอกฟังก์ชัน

def area(width, height):  
 result = width \* height  
 print(‘The area is’, result)

area(4, 5)

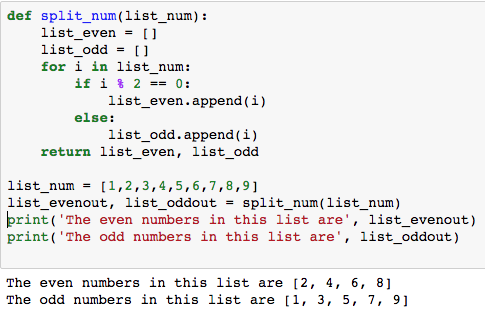
ตัวอย่างนี้เราไม่มีการส่งค่ากลับ และสั่งพิมพ์ในตัวฟังก์ชัน

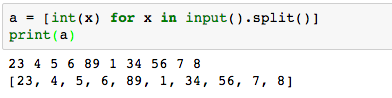
**1) ให้นักเรียนเขียนฟังก์ชันสำหรับคำนวณค่า Body Mass Index (BMI) โดยกำหนดพารามิเตอร์สองตัวคือ height และ weight โดยมีชื่อฟังก์ชันคือ calculate\_BMI(height, weight) และสำหรับฟังก์ชันแบบแรกให้ทำการพิมพ์ค่า BMI ในฟังก์ชันเลย และฟังก์ชันแบบที่สองให้ return ค่า BMI ไปยังตัวแปรชื่อ result และทำการพิมพ์ค่า BMI นอกฟังก์ชัน**

**2) ให้นักเรียนเขียนฟังก์ชันสำหรับแปลงหน่วยจากกิโลเมตรเป็นไมล์ โดยกำหนดชื่อฟังก์ชันคือ KmtoMi(distance) และมีพารามิเตอร์หนึ่งตัวคือ distance โดยฟังก์ชันแบบแรกให้พิมพ์ค่าที่แปลงแล้วในฟังก์ชัน และในแบบที่สองให้ส่งค่าที่แปลงแล้วออกมานอกฟังก์ชัน และทำการพิมพ์ค่า**

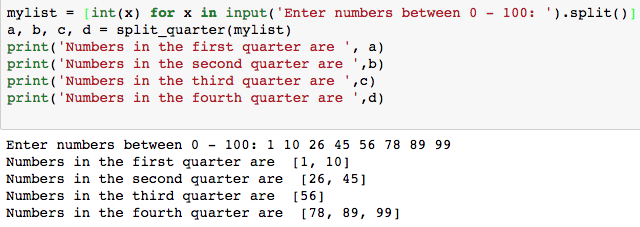
ตัวอย่างถัดไป เป็นฟังก์ชันชื่อ split\_num( ) ซึ่งรับค่าที่เป็นตัวแปรประเภท list และหน้าที่ของฟังก์ชันคือแยกเลขคู่เลขคี่ออกจากกัน จะเห็นได้ว่าเราต้องกำหนดตัวแปร list ไว้สองตัวเพื่อรองรับค่าที่เป็นเลขคู่ และเลขคี่

for i in list\_num: ตัวแปร i จะมีค่าเริ่มต้นจากค่าแรกใน list จนถึงค่าสุดท้าย และเราก็ทำการเช็คว่าเป็นเลขคู่หรือเลขคี่ แล้วก็เอาค่าไปใส่ไว้ในตัวแปร list ที่สร้างไว้ด้วยคำสั่ง .append(i) หลังจากนั้นเราก็ส่งค่ากลับทั้งหมดสองค่า

****

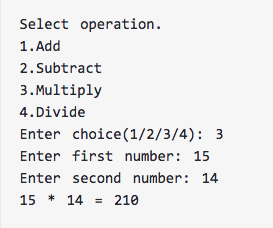
**3) จากตัวอย่างด้านบน ให้นักเรียนเขียนฟังก์ชันสำหรับแบ่งเลขจำนวนเต็มที่อยู่ใน list เป็นสองชุดคือเลขที่ค่าอยู่ระหว่าง 0 – 50 และเลขที่มากกว่า 50 และส่งค่าทั้งสองชุดออกมา นักเรียนสามารถกำหนดชื่อฟังก์ชันตามสมควร อาทิ split\_50( ) หลังจากนั้นให้เขียน Python code เพื่อรับเลขจำนวนเต็มหลาย ๆ ค่าจากคียบอร์ดโดยใช้คำสั่ง  และทำการเรียกฟังก์ชันที่นักเรียนเขียน และพิมพ์ค่าที่อยู่ใน list ทั้งสองออกมา (a จะเป็นตัวแปรประเภท list นักเรียนสามารถตั้งชื่ออื่นได้ตามความเหมาะสม เช่น num\_list หรือ int\_list)**

**4) ให้นักเรียนเขียนฟังก์ชันสำหรับแบ่งเลขจำนวนเต็มที่อยู่ใน list เป็นสี่ชุดคือเลขที่ค่าอยู่ระหว่าง 0 – 25, 26 – 50, 51 - 75 และ 75 - 100 และส่งค่าทั้งสี่ชุดออกมา ให้นักเรียนกำหนดชื่อฟังก์ชันคือ split\_quarter( ) หลังจากนั้นให้เขียน Python code เพื่อรับเลขจำนวนเต็มหลาย ๆ ค่าจากคียบอร์ด (ค่าที่ใส่ไม่เกิน 100) และเรียกฟังก์ชัน split\_quater โดยผ่านพารามิเตอร์เป็น list ที่เก็บค่าตัวเลข จากนั้นพิมพ์ค่าของ list ทั้งสี่ชุดออกมา ตัวอย่าง inputs/outputs**

****

**5) ให้นักเรียนเขียนฟังก์ชันสำหรับ บวก ลบ คูณ หาร (ทั้งหมด 4 ฟังก์ชัน) และเขียน Python code เพื่อให้เลือกว่าจะคำนวณอะไร (แสดงดังตัวอย่างด้านล่าง) พร้อมรับค่าตัวเลขสองตัว แล้วเรียกฟังก์ชันตาม option ที่เลือกพร้อมแสดงผลลัพธ์ และให้ทำการถามว่าจะ Continue หรือ Quit (ให้ประยุกต์ใช้ while\_loop จาก worksheet 5)**

**ตัวอย่าง input/output**

****

**หลักการในการเขียน Python code สำหรับแบบฝึกหัดนี้คือ นักเรียนต้องเขียนฟังก์ชันบวก ลบ คูณ หาร ทั้งหมดสี่ฟังก์ชัน จากนั้นเขียน code ถามผู้ใช้ว่าต้องการจะทำอะไร เช่น เลือก 1 เพื่อบวกเลข (หรือนักเรียนจะทำเป็น A เพื่อบวกเลข S เพื่อลบ) จากนั้นเขียนเงื่อนไข if-else เพื่อตรวจสอบว่าผู้ใช้เลือกจะทำอะไร ถ้าเลือก 1 ก็ให้เรียก add() เป็นต้น**

**choice = int(input(‘Please choose your option…’))**

**if choice == 1:**

**add()**

**elif choice == 2:**

**substract()**

**ในแต่ละฟังก์ชันก็จะถามผู้ใช้ให้ใส่ first number และ second number เช่น**

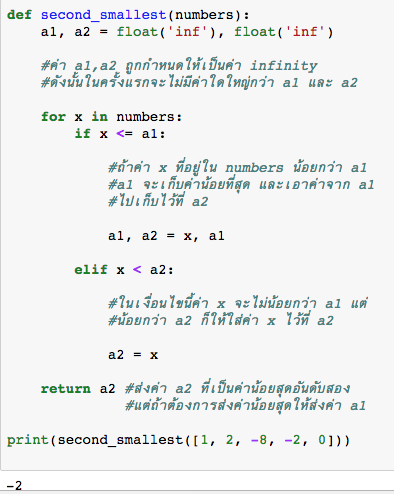
**def add():**

**first\_number = int(input(‘Enter first number’))**

**….**

**6) ให้นักเรียนเขียน Python code โดยมีฟังก์ชันสำหรับคำนวณหาพื้นที่แบบต่าง ๆ เช่น square, circle, triangle และอื่น ๆ หลักจากนั้นถามผู้ใช้ว่าจะเลือกคำนวณหาพื้นที่แบบใด หลักจากเรียกฟังก์ชันก็ให้ผู้ใช้ใส่ค่าเพื่อคำนวณหาพื้นที่พร้อมแสดงผล และให้ทำการถามว่าจะ Continue หรือ Quit (ให้ประยุกต์ใช้ while\_loop จาก worksheet 5)**

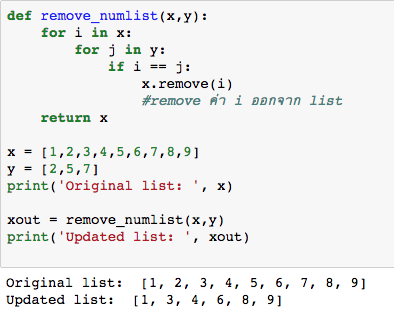
**ตัวอย่างถัดไป**

****

**หมายเหตุ ในการเปรียบเทียบครั้งแรกในการหาค่าที่เล็กที่สุด เนื่องจากเราไม่ทราบว่าควรจะกำหนดค่าใหญ่สุดที่เท่าใดสำหรับ a1 และ a2 เนื่องจากเราอาจไม่รู้ค่าที่ใหญ่ใน list เช่นถ้าเรากำหนดให้ค่า a1 และ a2 เป็น 1000000 แต่ใน list มีค่า 1000002 เราก็จะไม่สามารถเปรียบเทียบค่าได้ถูกต้อง ดังนั้นในการหาค่าที่เล็กที่สุด เราควรกำหนดค่าเริ่มต้นที่ต้องการจะเปรียบเทียบให้เป็น infinity**

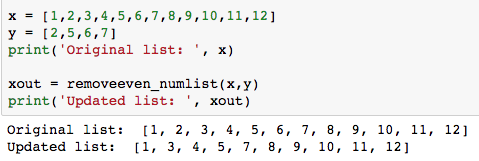
**7) ให้นักเรียนเขียนฟังก์ชัน second\_largest( ) เพื่อหาค่าที่มากเป็นอันดับสอง (ใช่ float(‘-inf’) สำหรับกำหนดค่าเริ่มต้นให้เป็น – infinity**

**จากตัวอย่างด้านล่างซึ่งเป็นฟังก์ชันที่ลบค่าตัวเลขที่อยู่ใน list y ออกจาก x**

****

**8) ให้นักเรียนเขียนฟังก์ชันสำหรับ remove เฉพาะเลขคู่ที่ปรากฏอยู่ใน list y ออกจาก list x**

**ตัวอย่าง inputs/outputs**

****

**จะเห็นได้ว่าเฉพาะเลขคู่ที่อยู่ใน list y ได้แก่ 2 และ 6 ได้ถูก remove ออกจาก list x**