

[Exercise] Functions-1

1 สร้างฟังก์ชัน `div()` ที่คำนวณค่า อินพุตตัวแรกหารด้วยอินพุตตัวที่สอง

```
In [6]: def div(a, b):  
        c = a / b  
        return(c)  
        print('answer is ')  
  
        result = div(int(input("first number : ")), int(input("second number : ")))  
        print(result)
```

6.0

2 จากฟังก์ชัน `con()` ที่กำหนดต่อไปนี้ จงตอบคำถาม

```
In [55]: # Use the con function for the following question  
def con(a, b):  
    return(a + b)
```

2.1 ฟังก์ชัน `con` ที่ประกาศไป สามารถใช้บวกตัวเลขจำนวนเต็ม (int) สองจำนวนหรือสตริง (str) สองข้อความได้หรือไม่

```
In [56]: # Write your code below and press Shift+Enter to execute  
(con("2", "2"))#ได้  
(con(6,5))#ได้
```

Out[56]: 11

2.2 ฟังก์ชัน `con` ที่ประกาศไป สามารถใช้กับข้อมูลประเภท Collections เช่น lists หรือ tuples ได้หรือไม่?

```
In [68]: # Write your code below and press Shift+Enter to execute  
list1 = [1,2,3,4,5]  
list2 = [12,13,14,15]  
con(list1, list2)#ได้
```

Out[68]: [1, 2, 3, 4, 5, 12, 13, 14, 15]

3 จงเขียนฟังก์ชันหาเศษของการหารระหว่างจำนวนเต็มสองจำนวน `remainder1(num1, num2)` โดยมีพารามิเตอร์ `num1` และ `num2` เป็นเลขจำนวนเต็ม และคืนค่ากลับเป็นเศษที่ได้จากการหาร `num1` ด้วย `num2` เช่น เศษของ `10%3` คือ `1`

```
In [18]: # Write your code below and press Shift+Enter to execute  
def remainder1(num1, num2):  
    ## BEGIN SOLUTION  
    output = num1%num2  
    return output
```

```
### END SOLUTION
print(f"{remainder1(10, 3) = }")
```

remainder1(10, 3) = 1

4 จงเขียนฟังก์ชันหาเศษใหม่ remainder2(num1, num2) โดยไม่ต้องใช้ตัวดำเนินการ %

In [26]: # Write your code below and press Shift+Enter to execute

```
def remainder2(num1, num2):
    ## BEGIN SOLUTION
    pass
    result = num1
    while result >= num2:
        result -= num2
    return result
    ### END SOLUTION

print(f"{remainder2(99, 10) = }")
```

remainder2(99, 10) = 9

5 จงเขียนฟังก์ชันหาเศษใหม่ remainder2(num1, num2) โดยไม่ต้องใช้ตัวดำเนินการ *, // และ %

In [27]: # Write your code below and press Shift+Enter to execute

```
def remainder2(num1, num2):
    ## BEGIN SOLUTION
    pass
    result = num1
    while result >= num2:
        result -= num2
    return result
    ### END SOLUTION

print(f"{remainder2(99, 10) = }")
```

remainder2(99, 10) = 9

6 จงเขียนฟังก์ชันหา Root Mean Square (RMS) ของตัวเลข 5 จำนวน cal_rms1(num1, num2, num3, num4, num5) โดยคืนค่ากลับเป็นค่า RMS และมี Docstring ดังนี้

Function: Calculate the Root Mean Square (RMS) of five numbers.

Input: Five numeric values (num1, num2, num3, num4, num5)

Output: RMS value

$$\text{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}$$

In [64]: # Write your code below and press Shift+Enter to execute

```
def cal_rms1(num1, num2, num3, num4, num5):
    ## BEGIN SOLUTION
    sum = num1**2 + num2**2 + num3**2 + num4**2 + num5**2
    rms = ((1/5)*sum)**(1/2)
```

```

pass

### END SOLUTION
return rms

# ตัวอย่างการใช้งาน
num1 = 3
num2 = 5
num3 = 7
num4 = 9
num5 = 11

rms_result = cal_rms1(num1, num2, num3, num4, num5)
print(f"Root Mean Square (RMS) of {num1}, {num2}, {num3}, {num4}, {num5} is {rms}

```

Root Mean Square (RMS) of 3, 5, 7, 9, 11 is 7.54983443527075

[Exercise] Functions-2

7 จากฟังก์ชันคำนวณหา RMS ข้อก่อนหน้า จงเขียนฟังก์ชันใหม่ที่มีพารามิเตอร์ `decimal_places` เพิ่ม `cal_rms2(num1, num2, num3, num4, num5, decimal_places)` เพื่อระบุจำนวนตำแหน่งทศนิยมที่ต้องการปัด โดยกำหนดค่าเริ่มต้นคือ 5

In [67]: *# Write your code below and press Shift+Enter to execute*

```

def cal_rms2(num1, num2, num3, num4, num5, decimal_places=5):
    """
    ฟังก์ชัน: คำนวณค่า Root Mean Square (RMS) ของห้าจำนวน พร้อมกำหนดจำนวนตำแหน่งทศนิยม
    อินพุต: ห้าค่าตัวเลข (num1, num2, num3, num4, num5), decimal_places (ค่าเริ่มต้นคือ 5)
    เอาท์พุต: ค่า RMS ที่ถูกปัดเศษตามจำนวนตำแหน่งทศนิยมที่ระบุ
    """

    ## BEGIN SOLUTION
    sum = num1**2 + num2**2 + num3**2 + num4**2 + num5**2
    rms = ((1/5)*sum)**(1/2)
    rms = round(rms, decimal_places)
    return rms
    ### END SOLUTION

# ตัวอย่างการใช้งาน
# [3, 5, 7, 9, 11]: 7.550
# [30, 50, 70, 90, 110]: 75.498
num1 = 3
num2 = 5
num3 = 7
num4 = 9
num5 = 11

decimal_places = 3 # ระบุจำนวนทศนิยมที่ต้องการปัด

rms_result = cal_rms2(num1, num2, num3, num4, num5, decimal_places)
print(f"ค่า Root Mean Square (RMS) ของ {num1}, {num2}, {num3}, {num4}, {num5} คือ

```

ค่า Root Mean Square (RMS) ของ 3, 5, 7, 9, 11 คือ 7.55

8 จากฟังก์ชันคำนวณหา RMS ข้อก่อนหน้า จงเขียนฟังก์ชันใหม่ที่สามารถรับตัวเลขได้ไม่จำกัดจำนวน `cal_rms3(*numbers, decimal_places=5)` โดยกำหนดค่าเริ่มต้นของพารามิเตอร์ `decimal_places` เป็น 5

In [61]: *# Write your code below and press Shift+Enter to execute*

```
def cal_rms3(*numbers, decimal_places=5):
    """
    ฟังก์ชัน: คำนวณค่า Root Mean Square (RMS) พร้อมกำหนดจำนวนตำแหน่งทศนิยม
    อินพุต: ตัวเลข (อาจมีจำนวนมากหรือน้อยกว่า 5), decimal_places (ค่าเริ่มต้นคือ 5)
    เอาท์พุต: ค่า RMS ที่ถูกปัดเศษตามจำนวนตำแหน่งทศนิยมที่ระบุ
    """

    ## BEGIN SOLUTION
    pass
    if len(numbers) > 0:
        for i in range(len(numbers)):
            min = numbers[0]
            if numbers[i] < min:
                min = numbers[i]

        else :
            return None
        return min
    ### END SOLUTION

# ตัวอย่างการใช้งาน
nums1 = [3, 5, 7, 9, 11]
nums2 = [30, 50, 70, 90, 110]

print(f"ค่า Root Mean Square (RMS) ของ {nums1} คือ {cal_rms3(*nums1)}")
print(f"ค่า Root Mean Square (RMS) ของ {nums2} คือ {cal_rms3(*nums2)}")
```

ค่า Root Mean Square (RMS) ของ [3, 5, 7, 9, 11] คือ 3

ค่า Root Mean Square (RMS) ของ [30, 50, 70, 90, 110] คือ 30

[Exersice] Scope of variables

1. หลังจากรันโค้ดต่อไปนี้ Output บนหน้าจอเป็นอย่างไร? (เพราะเหตุใด จงอธิบาย)

```
def f(x):
    x *= 5
    return x

def g(x):
    y = f(x**2)
    z = f(x**3)
    return y + z

print(g(2))
```

In [28]: *# Write your code below and press Shift+Enter to execute*

```
def f(x):
    x *= 5
    return x

def g(x):
    y = f(x**2)
```

```

    z = f(x**3)
    return y + z

print(g(2))
# เพราะว่า เราทำการป้อนฟังก์ชัน g(2) ซึ่ง 2 เข้าไปแทนใน x กำลัง 2 เท่ากับ 4 ซึ่ง y = f(x) = 4 * 5
# ค่าที่เรานำออกมาคือ y + z = 20 + 40 = 60

```

60

► [Click here for the output](#)

2. หลังจากรันโค้ดต่อไปนี้ Output พรินท์ออกมาจะเป็นอย่างไร? (เพราะเหตุใด จงอธิบาย)

```

def f(x):
    x += 8
    return round(x / 6)

def g(x):
    x *= 15
    return 2 * f(x)

def h(x):
    x += 2
    return f(x+1) + g(x)

print(h(f(1)))

```

In [31]: *# Write your code below and press Shift+Enter to execute*

```

def f(x):
    x += 8
    return round(x / 6)

def g(x):
    x *= 15
    return 2 * f(x)

def h(x):
    x += 2
    return f(x+1) + g(x)

print(h(f(1)))
# แทนค่า ในฟังก์ชัน f = 1 ใน f(x) ได้ค่าเท่ากับ 2
# ที่ g(2) = f(5)+g(4)
# ใน def g(x) ได้ g(4) = 4 * 15 = 60 return 2 * f(60) = 11 = 11 * 2 = 22 + 2 = 24

```

24

► [Click here for the output](#)

3. หลังจากรันโค้ดต่อไปนี้ Output พรินท์ออกมาจะเป็นอย่างไร?

```

g = 110

def f(x):
    return x + g

print(f(5))

```

```
print(f(10))
print(g)
```

In [32]: *# Write your code below and press Shift+Enter to execute*
g = 110

```
def f(x):
    return x + g

print(f(5))
print(f(10))
print(g)
#f(5) 110 + 5 = 115
#f(10) 110 + 10 = 120
#f(g) = 110
```

115
120
110

► [Click here for the output](#)

4. หลังจากรันโค้ดก่อนหน้าตามด้วยการรันโค้ดต่อไปนี้ Output พรินต์ออกมาจะเป็นอย่างไร?

```
def f(x):
    global g
    g += 1
    return x + g
```

```
print(f(5))
print(f(6))
print(g)
```

In [33]: *# Write your code below and press Shift+Enter to execute*
def f(x):
 global g
 g += 1
 return x + g

```
print(f(5))
print(f(6))
print(g)
#111 + 5 = 116
#112 + 6 = 118 (นำค่าในglobalเดิมที่ g = 111+เพิ่มไปอีก 1)
#112
```

116
118
112

In []: