

## Description

ต่อไปฉันจะนำเสนอการใช้งานฟังก์ชันในรูปแบบที่มีความหลากหลายมากขึ้น  
โดยวิธีการส่งข้อมูลเข้าฟังก์ชัน

ที่ผ่านมาฟังก์ชันของคุณมีลักษณะการทำงานตายตัว ไม่มีการทำงานที่หลากหลาย  
ทำให้ผลลัพธ์ของฟังก์ชันของคุณมีเพียงค่าเดียว

ลองมาดูตัวอย่างที่อธิบายได้เข้าใจง่ายกว่าการอธิบายของฉัน

```
"""Source code's docstring"""  
def functionName() :  
    """Function's docstring"""  
    print("Output")  
    print("Another Output")  
    print("Other Output")
```

คุณคุ้นเคยอยู่แล้วแน่นอน

ฉันต้องการให้คุณสังเกตคำสั่ง print ทั้งสามตัว

ใน Python; print คือฟังก์ชันการแสดงผลลัพธ์อะไรก็ได้ของภาษา Python  
ฟังก์ชัน print นั้นมีรับค่าหนึ่ง หรือหลายตัวก็ได้

รับค่าเพียงค่าเดียวเช่น print("Output")

รับค่าหลายค่าเช่น print("Output1", "Output2", "Output3", "Output4")

ค่าข้อมูลที่ส่งให้ฟังก์ชันเพื่อนำไปใช้งานนั้นเรียกว่า **Argument**

"Output", "Output1", "Output2", "Output3", "Output4" ล้วนมีสถานะเป็น **Argument** ของฟังก์ชัน print ทั้ง  
สิ้น

จะเห็นได้ว่าฟังก์ชัน print นั้น แสดงผลลัพธ์ออกมาแตกต่างกันขึ้นกับ **Argument** ที่รับเข้าไป (**Input**)

ต่อไปคุณต้องสร้างฟังก์ชันเพื่อขึ้นมาอีกตัวหนึ่ง

ใน **Source code** python 1 ไฟล้นั้น สามารถสร้างฟังก์ชันได้หลายตัว

อย่ายึดติดว่าในหนึ่งไฟล์สร้างฟังก์ชันได้เพียงตัวเดียว

จากรูปนี้ ฉันมีคำอธิบายมากมายที่จะบอกคุณ คุณอาจจะรู้สึกเบื่อได้

และบางทีคุณอาจจะรู้สึกไม่จำเป็นต้องทำตาม

แต่เชื่อฉันเถอะ คุณต้องการมันในภายหลัง

ไม่ได้ชู้

ห่าห่า

```

"""Source code's docstring"""
def function1():
    """Function1's docstring"""

    function2("Test")

    input_value1 = "Output"
    function2(input_value1)

    input_value2 = "Another Output"
    function2(input_value2)

    input_value3 = "Other Output"
    function2(input_value3)

def function2(parameter):
    """Function2's docstring"""
    print(parameter)

function1()

```

ในโปรแกรมหลัก (Main)

มีการเรียกใช้ function1 เท่านั้น

โดยที่ function1 ไม่มีการรับข้อมูลเข้า

และ function2

มีการรับค่าข้อมูลเข้าฟังก์ชัน 1 ตัวเท่านั้น

ตัวแปรที่ใช้รับข้อมูลเข้าฟังก์ชันเรียกว่า **Parameter**

ซึ่งชื่อตัวแปร ฉันก็ตั้งชื่อว่า parameter เหมือนกัน; หาก你不ชอบชื่อนี้ คุณสามารถตั้งชื่อตามใจคุณ  
หลักการตั้งชื่อเป็นไปตามหลักการตั้งชื่อตัวแปรทั่วไป

function2 นั้นมีคำสั่งเดียวคือให้ print ตัวแปร parameter

เมื่อคุณพิจารณาคำสั่งในฟังก์ชัน function1 จะมีชุดคำสั่งอยู่หลายบรรทัด

ฉันแยกไว้ให้เป็น 4 ส่วนให้คุณลองวิเคราะห์

ส่วนที่ 1

function2("Test")

ฉันใช้ข้อมูล "Test" ส่งเป็น **Argument** เข้า function2 โดยตรง

เพื่อต้องการให้แสดงผลลัพธ์ "Test" จากคำสั่ง print ข้างใน function2 อีกทีหนึ่ง

คุณอาจจะเห็นว่ามีการทำงานซ้ำซ้อน แต่นี้เป็นพื้นฐานที่เข้าใจง่ายที่สุดในความคิดฉัน

คุณควรจะเข้าใจหลักการทำงานของการทำงานส่งข้อมูลเข้าฟังก์ชัน

ส่วนที่ 2-3-4

ฉันสร้างตัวแปรขึ้นมา 3 ตัว และส่งให้กับ function2

เพื่อแสดงผลลัพธ์ออกมาสามแบบที่ไม่เหมือนกัน

---

จนถึงตรงนี้คุณจะสามารถเขียนสร้างฟังก์ชันให้มีการรับข้อมูลเข้าไปในฟังก์ชันได้แล้ว  
งานของคุณคือให้ดัดแปลงโปรแกรมตัวอย่างของฉัน เพื่อแสดงผลลัพธ์ให้ตรงตามที่คุณต้องการ

ฉันไม่บังคับวิธีการของคุณ

แต่คุณควรดัดแปลงโปรแกรมของฉันให้เหมาะสม

by Chotipat Pornavalai (<https://ejudge.it.kmitl.ac.th/account/6>)

8 August 2017, 13:45

## Specification

### Input Specification

ไม่มี

ให้สร้างข้อมูลใช้งานภายใน function1

### Output Specification

12 บรรทัด ให้ดูจาก **Sample Output**

การแสดงผลลัพธ์ควรมาจาก function2

## Sample Case

### Sample Input

### Sample Output

Output1  
Output1  
Output1  
Output2  
Output2  
Output2  
Output3  
Output3  
Output3  
Output4  
Output4  
Output4

## Time Remaining

4

Days

3

Hours

34

Minutes

29

Seconds

718

## Information

### Time Limit

1 Second