

Function

Sirion Vittayakorn



Functions

- ฟังก์ชัน คือ โปรแกรมย่อย (*subprogram หรือ subroutine*) ที่ออกมาจากโปรแกรมหลัก
 - เป็นการแยกส่วนคำสั่งที่ซ้ำๆ กัน หรือเข้าใจยาก ทำให้โปรแกรมอ่านง่าย เข้าใจง่าย
 - ทำให้โปรแกรมหลักเรียกใช้ฟังก์ชันได้ โดยไม่ต้องเขียนคำสั่งหลายรอบ

Functions



- 1 Baby shark
- 2 Happy Birthday
- :

```
def jukeboxSongs():  
    sid = int(input('Please select the song number (1-100):'))  
    if sid == 1:  
        sing('Baby shark, doo doo doo doo doo doo')  
        sing('Baby shark, doo doo doo doo doo doo')  
        sing('Baby shark, doo doo doo doo doo doo')  
        sing('Baby shark!')  
        ...  
    elif sid == 2:  
        sing('Happy birthday to you')  
        sing('Happy birthday to you')  
        sing('Happy birthday, happy birthday')  
        sing('Happy birthday to you')  
        ...  
    elif sid == 3:  
        ...
```

Functions



- 1 Baby shark
- 2 Happy Birthday
- ⋮

Jukebox₁ @ Cafeteria



- 1 Happy Birthday
- 2 Baby shark
- ⋮

Jukebox₂ @ Lab304

```
def jukeboxSongs():
    sid = int(input('Please select the song number (1-100):'))
    if sid == 1:
        sing('Baby shark, doo doo doo doo doo doo')
        sing('Baby shark, doo doo doo doo doo doo')
        sing('Baby shark, doo doo doo doo doo doo')
        sing('Baby shark!')
        ...
    elif sid == 2:
        sing('Happy birthday to you')
        sing('Happy birthday to you')
        sing('Happy birthday, happy birthday')
        sing('Happy birthday to you')
        ...
    elif sid == 3:
        ...
```

```
def jukeboxSongs():
    sid = int(input('Please select the song number (1-100):'))
    if sid == 2:
        sing('Happy birthday to you')
        sing('Happy birthday to you')
        sing('Happy birthday, happy birthday')
        sing('Happy birthday to you')
        ...
    elif sid == 1:
        sing('Baby shark, doo doo doo doo doo doo')
        sing('Baby shark, doo doo doo doo doo doo')
        sing('Baby shark, doo doo doo doo doo doo')
        sing('Baby shark!')
        ...
    elif sid == 3:
        ...
```

Functions



- 1 Baby shark
- 2 Happy Birthday
- ⋮

Jukebox₁ @ Cafeteria



- 1 Happy Birthday
- 2 Baby shark
- ⋮

Jukebox₂ @ Lab304

```
def jukeboxSongs():  
    sid = int(input('Please select the song number (1-100):'))  
    if sid == 1:  
        babyShark()  
    elif sid == 2:  
        happyBirthDay()  
    elif sid == 3:  
        ...
```

```
def jukeboxSongs():  
    sid = int(input('Please select the song number (1-100):'))  
    if sid == 1:  
        happyBirthDay()  
    elif sid == 2:  
        babyShark()  
    elif sid == 3:  
        ...
```

```
def babyShark():  
    sing('Baby shark, doo doo doo doo doo doo')  
    sing('Baby shark, doo doo doo doo doo doo')  
    sing('Baby shark, doo doo doo doo doo doo')  
    sing('Baby shark!')  
    ...
```

```
def happyBirthDay():  
    sing('Happy birthday to you')  
    sing('Happy birthday to you')  
    sing('Happy birthday, happy birthday')  
    sing('Happy birthday to you')  
    ...
```

ทำไมต้องเขียนฟังก์ชัน

- เพราะการเขียนโปรแกรมซ้ำๆกัน หรือมีการทำงานเหมือนกันมากกว่า 1 ที่ ทำให้ยุ่งยากต่อการจัดการ
 - ต้องเขียนโปรแกรมซ้ำๆกัน มากกว่า 1 ครั้ง
 - การเปลี่ยนแปลงโปรแกรมทำได้ยาก (ต้องทำการเปลี่ยนแปลงโปรแกรมหลายจุด)

ข้อดีของฟังก์ชัน

- ฟังก์ชันช่วยในการ
 - ลดความซ้ำซ้อนของโปรแกรม
 - ทำให้โปรแกรมเข้าใจง่ายขึ้น และง่ายต่อการจัดการ

Functions

- ฟังก์ชันสามารถแบ่งออกได้หลายประเภท ดังนี้
 - ส่วนการทำงานหลักของโปรแกรม เช่น `main()`

```
def main():  
    name = input('Please enter your name:')  
    print('Hello,' , name)  
  
if __name__ == '__main__':  
    main()
```

- Build-in functions: `abs()`, `eval()`, `int()`, etc.

```
number = '-123.45'           # string  
print(number)                 # print() displays a value on screen  
print(pow(float(number), 2))  # float() converts a string to float  
                               # and pow(x, y) return x to the power y  
print(abs(float(number)))     # abs() returns an absolute value of input
```


Build-in function

Built-in Functions

<u>abs()</u>	<u>delattr()</u>	<u>hash()</u>	<u>memoryview()</u>	<u>set()</u>	<u>compile()</u>	<u>classmethod()</u>
<u>all()</u>	<u>dict()</u>	<u>help()</u>	<u>min()</u>	<u>setattr()</u>	<u>complex()</u>	<u>getattr()</u>
<u>any()</u>	<u>dir()</u>	<u>hex()</u>	<u>next()</u>	<u>slice()</u>	<u>globals()</u>	<u>locals()</u>
<u>ascii()</u>	<u>divmod()</u>	<u>id()</u>	<u>object()</u>	<u>sorted()</u>	<u>hasattr()</u>	<u>repr()</u>
<u>bin()</u>	<u>enumerate()</u>	<u>input()</u>	<u>oct()</u>	<u>staticmethod()</u>	<u>map()</u>	<u>zip()</u>
<u>bool()</u>	<u>eval()</u>	<u>int()</u>	<u>open()</u>	<u>str()</u>	<u>max()</u>	<u>frozenset()</u>
<u>breakpoint()</u>	<u>exec()</u>	<u>isinstance()</u>	<u>ord()</u>	<u>sum()</u>	<u>reversed()</u>	<u>list()</u>
<u>bytearray()</u>	<u>filter()</u>	<u>issubclass()</u>	<u>pow()</u>	<u>super()</u>	<u>round()</u>	<u>range()</u>
<u>bytes()</u>	<u>float()</u>	<u>iter()</u>	<u>print()</u>	<u>tuple()</u>	<u>__import__()</u>	<u>vars()</u>
<u>callable()</u>	<u>format()</u>	<u>len()</u>	<u>property()</u>	<u>type()</u>	<u>chr()</u>	

Functions

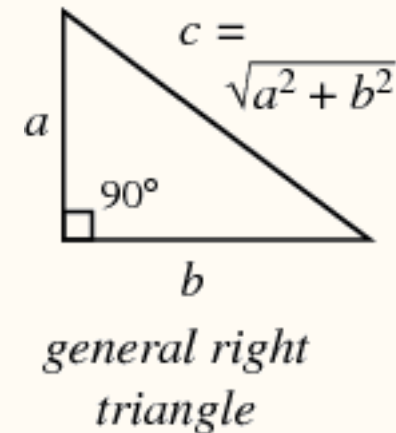
- Functions จาก Standard libraries เช่น math, string, datetime, random, enum, statistics, array, etc.

Import library

```
import math
def hypotenuseLength(a,b):
    #  $c^2 = a^2 + b^2$ 
    lenC = math.sqrt(pow(a,2)+pow(b,2))
    return (lenC) เรียกใช้งาน function จาก math library
```

```
hypotenuseLength(3,4)
```

5.0



The Python Standard Library: <https://docs.python.org/3/library/index.html>

องค์ประกอบของฟังก์ชัน

- ชื่อฟังก์ชัน มีข้อกำหนดเหมือนการตั้งชื่อตัวแปร
- ค่าที่รับเข้ามา หรือ "พารามิเตอร์" (ไม่จำเป็นต้องมีก็ได้)
- การคืนค่าจากฟังก์ชันด้วยคำสั่ง `return` (ไม่จำเป็นต้องมีก็ได้)

ชื่อฟังก์ชัน ค่าที่รับเข้ามา

```
def gen_password(username):  
    password = username[0:6]  
    password = password[::-1]*2  
    return (password)
```

การคืนค่าจากฟังก์ชัน

การเรียกใช้ฟังก์ชัน

```
tmp = gen_password('abcDE')  
print(tmp)
```

EDcbaEDcba

ชื่อฟังก์ชัน

```
def countTen():  
    for i in range(10,0, -1):  
        print(i)
```

countTen()

10
9
8
7
6
5
4
3
2
1

Example1

- จงเขียนโปรแกรมดังรายละเอียดต่อไปนี้
 - สร้างฟังก์ชันชื่อ **main** คอยทำหน้าที่
 - รับชื่อคน.จากผู้ใช้งาน แล้วเก็บไว้ในตัวแปร **sname**
 - แสดงข้อความ
 - Happy birthday to you
 - Happy birthday to you
 - Happy birthday dear(**sname**)
 - Happy birthday to you

```
def main():  
    sname = input('Enter your name:')  
    print('Happy birthday to you')  
    print('Happy birthday to you')  
    print('Happy birthday dear', sname)  
    print('Happy birthday to you')
```

Example 1

- เรียกใช้งานฟังก์ชัน `main` แล้วใส่ข้อมูลนำเข้าเป็นชื่อตัวเอง

```
main()
```

Enter your name:

Example 1

- เราทำการสร้างฟังก์ชันชื่อ main

```
def main():  
    sname = input('Enter your name:')  
    print('Happy birthday to you')  
    print('Happy birthday to you')  
    print('Happy birthday dear', sname)  
    print('Happy birthday to you')
```

- เราทำการเรียกใช้ฟังก์ชัน main
- ฟังก์ชัน main ไม่มีพารามิเตอร์ และไม่มีการคืนค่า

```
main()
```

```
Enter your name:Sirion  
Happy birthday to you  
Happy birthday to you  
Happy birthday dear Sirion  
Happy birthday to you
```

Example 2

- จงเขียนโปรแกรมดังรายละเอียดต่อไปนี้
 - สร้างฟังก์ชันชื่อ **main** คอยทำหน้าที่
 - รับชื่อคน.จากผู้ใช้งาน แล้วเก็บไว้ในตัวแปร **sname**

```
def main():  
    sname = input('Enter your name:')
```

Example 2

- สร้างฟังก์ชันชื่อ `displaySong` คอยทำหน้าที่
 - ฟังก์ชัน `displaySong` จะรับพารามิเตอร์ 1 ตัว ตั้งชื่อว่า `uname`
 - แสดงข้อความ
 - Happy birthday to you
 - Happy birthday to you
 - Happy birthday dear(`uname`)
 - Happy birthday to you

```
def displaySong(uname):  
    print('Happy birthday to you')  
    print('Happy birthday to you')  
    print('Happy birthday dear', uname)  
    print('Happy birthday to you')
```


Example 2

- ในฟังก์ชัน `main` ให้เรียกใช้งานฟังก์ชัน `displaySong` โดยส่งค่า `sname` ให้เป็นข้อมูลนำเข้า

```
def main():  
    sname = input('Enter your name:')  
    displaySong(    )
```

Example 2

- เรียกใช้งานฟังก์ชัน `main` แล้วใส่ข้อมูลนำเข้าเป็นชื่อตัวเอง

```
main()
```

Enter your name:

Example 2

- เราทำการสร้างฟังก์ชันชื่อ displaySong
 - ฟังก์ชัน displaySong มีพารามิเตอร์ 1 ตัว ชื่อว่า uname แต่ไม่มีการคืนค่า

```
def displaySong(uname):  
    print('Happy birthday to you')  
    print('Happy birthday to you')  
    print('Happy birthday dear', uname)  
    print('Happy birthday to you')
```

Example 2

- ในฟังก์ชัน `main` เรียกใช้งานฟังก์ชัน `displaySong` โดยส่ง ค่าของตัวแปร `sname` ให้เป็นข้อมูลนำเข้า

```
def main():  
    sname = input('Enter your name:')  
    displaySong(sname)
```

- เมื่อมีการเรียกใช้งานฟังก์ชัน `main` และใส่ข้อมูลนำเข้าเป็นชื่อตัวเอง

```
main()  
  
Enter your name:Sirion  
Happy birthday to you  
Happy birthday to you  
Happy birthday dear Sirion  
Happy birthday to you
```

Example 3

- จงเขียนโปรแกรมดังรายละเอียดต่อไปนี้
 - สร้างฟังก์ชันชื่อ **main** คอยทำหน้าที่รับตัวเลข 2 ตัวจากผู้ใช้งาน แล้วนำค่าไปเก็บไว้ในตัวแปร **num_x** และ **num_y**

```
def main():  
    num_x = int(input('Input X: '))  
    num_y = int(input('Input Y: '))
```

Example 3

- สร้างฟังก์ชัน calSum:
 - รับพารามิเตอร์ 2 ตัว ตั้งชื่อว่า num_one, num_two
 - คำนวณหาผลรวมของ num_one และ num_two
 - คืนค่าผลรวมกลับไป

```
def calSum(num_one, num_two):  
    summation = num_one + num_two  
    return (summation)
```

Example 3

- ภายในฟังก์ชัน `main`:
 - เรียกใช้งานฟังก์ชัน `calSum` โดยส่งค่า `num_x` และ `num_y`

```
def main():  
    num_x = int(input('Input X: '))  
    num_y = int(input('Input Y: '))
```

```
main()
```

```
Input X: 1  
Input Y: 2
```

- รับค่าผลลัพธ์ที่คืนมาจากฟังก์ชัน `calSum` แล้วนำมาเก็บไว้ที่ตัวแปร `num_z` เพื่อแสดงผล

```
def main():  
    num_x = int(input('Input X: '))  
    num_y = int(input('Input Y: '))
```

```
main()
```

```
Input X: 1  
Input Y: 2  
1+2 = 3
```

Example 3

- เราทำการสร้างฟังก์ชันชื่อ calSum
 - ฟังก์ชัน calSum มีพารามิเตอร์ 2 ตัวชื่อว่า num_one และ num_two
 - มีการคืนค่ากลับมายังฟังก์ชัน main (ซึ่งเป็นฟังก์ชันต้นทางที่เรียกใช้งานฟังก์ชัน calSum)

```
def calSum(num_one, num_two):  
    summation = num_one + num_two  
    return (summation)
```


Example 3

- พารามิเตอร์(parameter) คือ ข้อมูลที่รับจากภายนอก (input) เข้ามาใช้งานภายในฟังก์ชัน
 - พารามิเตอร์สามารถมีหลายตัวได้
 - ฟังก์ชันเดียวกัน หากค่าพารามิเตอร์เปลี่ยนไป ผลลัพธ์อาจจะเปลี่ยนไป

```
def calSum(num_one, num_two):  
    summation = num_one + num_two  
    return (summation)
```

```
calSum(10, -3)
```

7 $\text{parameter}_1 = 10$
 $\text{parameter}_2 = -3$

```
calSum(245, 1245)
```

1490 $\text{parameter}_1 = 245$
 $\text{parameter}_2 = 1245$

Multiple parameters

```
def occupation(name, age):  
    if (age <= 22):  
        print(name.title(), 'is a student.')  
    else:  
        print(name.title(), 'is an employee.')
```

```
occupation('adam', 19)
```

Adam is a student.

Arguments:
name = 'adam'
age = 19

```
occupation('jack', 30)
```

Jack is an employee.

Arguments:
name = 'jack'
age = 30

parameters are matched up based on *position*

Example

```
def main():  
    name = 'Adam'  
    printName(name)  
    print('Name: ', name)
```

```
def printName(name):  
    print('Name: ', name)  
    name = 'Jack'
```

```
def printName(param):  
    print('Name: ', param)
```

Call function: `main()`

A)

```
Name:  Adam  
Name:  Adam
```

B)

```
Name:  Adam  
Name:  Jack
```

Function that returns value

- จงเขียนฟังก์ชันหาค่าผลรวมของเลข 2 จำนวน:

Define function:

```
def sumNumber(input_x, input_y):  
    output = input_x + input_y
```

Call function:

```
sumNumber(10, 5)
```

Result:

Function that returns value

- จงเขียนฟังก์ชันหาค่าผลรวมของเลข 2 จำนวน :

Define function

```
def sumNumber(input_x, input_y):  
    output = input_x + input_y  
    return output
```

```
In [58]: sumNumber(10,5)  
Out[58]: 15
```

```
In [57]: result = sumNumber(10, 5)  
         print(result)  
15
```

Function that returns multiple values

```
def plusMinus(input_x, input_y):  
    plus = input_x + input_y  
    minus = input_x - input_y  
    return plus, minus
```

```
In [61]: plusMinus(10, 5)
```

```
Out[61]: (15, 5)
```

```
In [62]: a, b = plusMinus(10, 5)  
print('a:', a, 'b:', b)
```

```
a: 15 b: 5
```

Example

- กำหนดให้ฟังก์ชัน `addInterest` คำนวณยอดเงินฝากพร้อมดอกเบี้ย จากยอดเงินต้น(`balance`) และอัตราดอกเบี้ย (`rate`) ที่กำหนดให้:

```
def addInterest(balance, rate):  
    newBalance = balance * (1 + rate)  
    return newBalance
```

```
def main():  
    amount = 1000  
    rate = 0.05  
    addInterest(amount, rate)  
    print(amount)
```

A) 1000.00

B) 1050.00

Example

- กำหนดให้ฟังก์ชัน `addInterest` คำนวณยอดเงินฝากพร้อมดอกเบี้ย จากยอดเงินต้น(`balance`) และอัตราดอกเบี้ย (`rate`) ที่กำหนดให้:

```
def addInterest(balance, rate):  
    newBalance = balance * (1 + rate)  
    return newBalance
```

```
def main():  
    amount = 1000  
    rate = 0.05  
    amount = addInterest(amount, rate)  
    print(amount)
```

B) 1050.00

การส่งค่าระหว่างฟังก์ชันผ่านพารามิเตอร์

- การส่งค่าระหว่างฟังก์ชันผ่านพารามิเตอร์ สามารถแบ่งออกได้เป็น 2 รูปแบบ คือ
 - **Pass by value** คือ การคัดลอกค่า ที่เก็บจากตัวแปรที่ส่ง ไปยังตัวแปรของฟังก์ชัน
 - **Pass by reference** คือ การใช้งาน ข้อมูลที่เก็บจากตำแหน่งในหน่วยความจำ (address) ของตัวแปรที่ส่งไป ในฟังก์ชันนั้น

Pass by value

ตัวแปร (variables) คือ การจองพื้นที่ในหน่วยความจำเพื่อเก็บค่าของตัวแปรนั้นๆ



Memory		
1	2	3
4	5	6
7	8	9

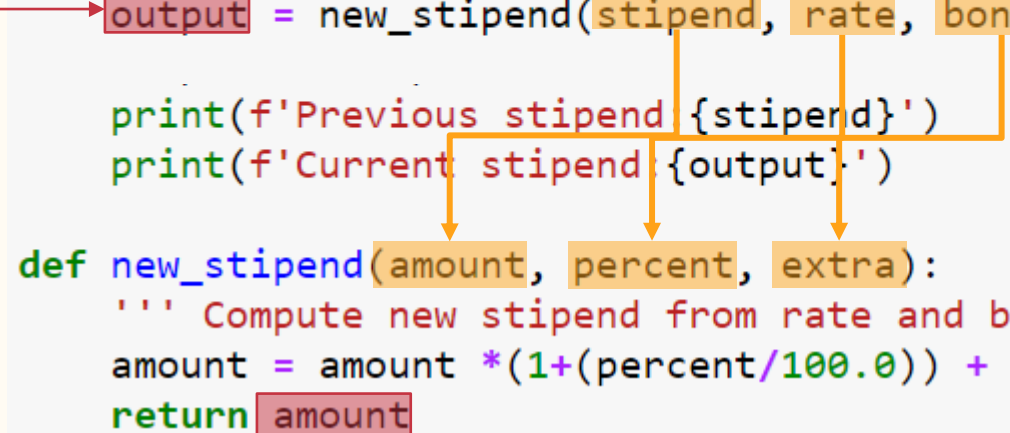
Address

Pass by value

```
def main():
    stipend = float(input('Please enter your stipend: '))
    rate = float(input('Please enter your rate: '))
    bonus = float(input('Please enter your bonus: '))
    output = new_stipend(stipend, rate, bonus)

    print(f'Previous stipend {stipend}')
    print(f'Current stipend {output}')

def new_stipend(amount, percent, extra):
    ''' Compute new stipend from rate and bonus '''
    amount = amount * (1 + (percent/100.0)) + extra
    return amount
```



main()

```
Please enter your stipend: 30000
Please enter your rate: 4
Please enter your bonus: 5000
Previous stipend:30000.0
Current stipend:36200.0
```

main()

```
Please enter your stipend: 30000
Please enter your rate: 4
Please enter your bonus: 5000
Previous stipend:36000.0
Current stipend:36200.0
```

Pass by value

```
def main():
```

```
    stipend = input(...)
```

```
    output = new_stipend(stipend, ...)
```

```
    stipend = stipend*1.2
```

1. stipend = 10 @ address 1

2. output = 15 @ address 3

```
def new_stipend(amount, ...):
```

```
    amount *= 1.5
```

```
    return amount
```

1. amount = 10 @ address 5

2. Update amount = 15 @ address 5

Memory		
stipend	2	output
10		15
4	amount	6
	15	
7	8	9

Pass by reference

```
def main():
    stipend = [float(x) for x in input('Please enter stipends: ').split(',')]
    rate = [float(x) for x in input('Please enter rate: ').split(',')]
    bonus = float(input('Please enter bonus: '))

    output = new_stipend(stipend, rate, bonus)
    print(f'Previous stipend:{stipend}')
    print(f'Current stipend:{output}')

def new_stipend(amount, percent, extra):
    ''' Compute new stipend from rate and bonus '''
    for i in range(len(amount)):
        amount[i] = amount[i] * (1 + (percent[i]/100.0)) + extra
    return amount
```

```
main()
```

```
Please enter stipends: 30000,32000,35000
Please enter rate: 2,3,4
Please enter bonus: 5000
Previous stipend:[35600.0, 37960.0, 41400.0]
Current stipend:[35600.0, 37960.0, 41400.0]
```

Pass by reference

```
def main():
```

```
→ stipend = [30000, 32000, 35000]
```

```
→ rate = [2, 3, 4]
```

```
→ bonus = 5000
```

```
→ new_stipend(stipend, rate, bonus)
```

1. stipend = [30000, ...] @ address 1

2. rate = [2, 3, 4] @ address 2

3. bonus = 5000 @ address 3

```
def new_stipend(amount, percent, extra):
```

```
    for i in range(len(amount)):
```

```
        amount[i] = amount[i] * (1 + (percent[i]/100)) + extra
```

```
    return amount
```

1. amount = [30000, ...] @ address 1

2. percent = [2, 3, 4] @ address 2

3. extra = 5000 @ address 6

Memory

stipend/ amount	rate/ percent	bonus
30000	2	5000
4	5	extra

Pass by reference

```
def main():  
    stipend = [30000, 32000, 35000]  
    rate = [2, 3, 4]  
    bonus = 5000  
    → new_stipend(stipend, rate, bonus)
```

1. stipend = [35600, ...] @ address 1
2. rate = [2, 3, 4] @ address 2
3. bonus = 5000 @ address 3

```
def new_stipend(amount, percent, extra):  
    for i in range(len(amount)):  
        amount[i] = amount[i] * (1 + (percent[i]/100)) + extra  
    return amount
```

1. amount: [35600, ...] @ address 1
2. percent = [2, 3, 4] @ address 2
3. extra = 5000 @ address 3

Memory		
stipend/ amount	rate/ percent	bonus
35600	2	5000
4	5	extra
		5000

Pass by reference

```
def main():
    stipend = [float(x) for x in input('Please enter stipends: ').split(',')]
    rate = [float(x) for x in input('Please enter rate: ').split(',')]
    bonus = float(input('Please enter bonus: '))

    output = new_stipend(stipend, rate, bonus)
    print(f'Previous stipend:{stipend}')
    print(f'Current stipend:{output}')

def new_stipend(amount, percent, extra):
    ''' Compute new stipend from rate and bonus '''
    for i in range(len(amount)):
        amount[i] = amount[i] * (1 + (percent[i]/100.0)) + extra
    return amount
```

***^สไม่มีการ update ค่า stipend
แต่ค่าของ stipend ก็ยังคงมีการเปลี่ยนแปลง

```
main()
```

```
Please enter stipends: 30000,32000,35000
```

```
Please enter rate: 2,3,4
```

```
Please enter bonus: 5000
```

```
Previous stipend:[35600.0, 37960.0, 41400.0]
```

```
Current stipend:[35600.0, 37960.0, 41400.0]
```


Pass by reference

```
def main():
    stipend = [float(x) for x in input('Please enter stipends: ').split(',')]
    rate = [float(x) for x in input('Please enter rate: ').split(',')]
    bonus = float(input('Please enter bonus: '))

    new_stipend(stipend, rate, bonus)
    print(f'Previous stipend:{stipend}')

def new_stipend(amount, percent, extra):
    ''' Compute new stipend from rate and bonus '''
    for i in range(len(amount)):
        amount[i] = amount[i] *(1+(percent[i]/100.0)) + extra
```

***ไม่มีการ update ค่า stipend

***ไม่มีการคืนค่าจากฟังก์ชัน

```
main()
```

```
Please enter stipends: 30000,32000,34000
```

```
Please enter rate: 2,3,4
```

```
Please enter bonus: 5000
```

```
Previous stipend:[35600.0, 37960.0, 40360.0]
```

Function: passing argument

	Pass by value	Pass by reference
1. definition	Pass the value of the argument to parameter.	Pass the address of the argument to parameter.
2. work with	Immutable types: <ul style="list-style-type: none">- bool- integer- float- tuple- string	Mutable types: <ul style="list-style-type: none">- list- set- dictionary

Hint: to quickly test if a type is mutable or not, is to use id() function.

Return the "identity" of an object. This is an integer which is guaranteed to be unique and constant for this object during its lifetime. (The address of the object in memory.)

Exercise

```
def main():
    stipend = [float(x) for x in input('Please enter stipends: ').split(',')]
    rate = [float(x) for x in input('Please enter rate: ').split(',')]
    bonus = float(input('Please enter bonus: '))

    output = new_stipend(stipend, rate, bonus)
    print(f'Previous stipend:{stipend}')
    print(f'Current stipend:{output}')
```

```
def new_stipend(amount, percent, extra):
```

ให้นัก.ลองทำการแก้ไขโปรแกรมในฟังก์ชัน new_stipend เพื่อให้ผลลัพธ์ดังที่แสดงด้านล่าง
***โดยห้ามแก้ไขฟังก์ชัน main โดยเด็ดขาด

```
main()
```

```
Please enter stipends: 30000,32000,34000
```

```
Please enter rate: 2,3,4
```

```
Please enter bonus: 5000
```

```
Previous stipend:[30000.0, 32000.0, 34000.0]
```

```
Current stipend:[35600.0, 37960.0, 40360.0]
```



Questions?

Function