

More on Defining a function

Recap

- การนิยามฟังก์ชัน คือการตั้งชื่อให้ กลุ่มของคำสั่ง เพื่อที่เราจะสามารถนำกลุ่มของคำสั่งนั้นกลับมาใช้ได้อีก
- ในภาษา Python สามารถทำได้โดยใช้ syntax ต่อไปนี้

```
def functionName(argument1, argument2, ...):  
    process1  
    process2  
    . . .  
    processN
```

} กลุ่มคำสั่ง

- ส่วนของ argument จะมีหรือไม่มีก็ได้ แล้วแต่หน้าที่ของฟังก์ชัน
- Process แต่ละตัวที่อยู่ในฟังก์ชันต้องทำการย่อหน้าให้ถูกต้อง

Variable scope

- Global variable

หมายความว่า ตัวแปรที่ประกาศไว้ นอก function ซึ่งฟังก์ชันทุกตัวสามารถเรียกใช้งานได้

- Local variable

หมายความว่า ตัวแปรที่ประกาศไว้ ใน function รวมถึงตัวแปรที่ทำหน้าที่เป็น *argument* ด้วย จะมีแต่ฟังก์ชันที่เป็นเจ้าของเท่านั้นที่เรียกใช้งานได้

โปรแกรมต่อไปนี้ทำงานได้หรือไม่

```
1 globalGift = "Diamond"
2
3
4 def HappyBirthday(person):
5     localGift = "Rock"
6     if (person.lower() == 'je'):
7         return globalGift
8     elif (person.lower() == 'yoon'):
9         return localGift
10    else:
11        return "Nothing"
12
13 gift = HappyBirthday('je')
14 print("I get ", gift, " as a gift")
15
16 print("Yoon will get ", localGift, "as a gift")
```

โปรแกรมต่อไปนี้ทำงานได้หรือไม่

```
1 globalGift = "Diamond"
2
3
4 def HappyBirthday(person):
5     localGift = "Rock"
6     if (person.lower() == 'je'):
7         return globalGift
8     elif (person.lower() == 'yoon'):
9         return localGift
10    else:
11        return "Nothing"
12
13 gift = HappyBirthday('je')
14 print("I get ", gift, " as a gift")
15
16 print("Yoon will get ", localGift, "as a gift")
```

local

local

global

เรียกใช้
ไม่ได้
เพราะเป็น
local

What happens to function argument?

- จะเกิดอะไรขึ้นกับค่าตัวแปรที่ส่งเป็น argument เข้าไปในฟังก์ชัน แล้วมีการเปลี่ยนแปลงค่าของ argument ภายในฟังก์ชัน

Experiment 1: โปรแกรมถูกต้องไหม

```
1 def HappyBirthday(person, age):  
2     print("Happy Birthday my dear ", person)  
3     age = age + 1  
4  
5 name = input("What's your name?: ")  
6 age  = int(input("How old are you? "))  
7  
8 HappyBirthday(name, age)  
9  
10 print(name, ", you are now", age, "year-old")
```

Experiment 2: แล้วโปรแกรมนี้จะ

```
1 def getMarried(bridename, groomName):
2     brideName[1] = groomName[1]
3
4 # Input name in the form 'Firstname Lastname'
5 # What we get from split(' ') is a list in the form ['firstname', 'lastname']
6 brideName = input("What's the name of the bride: ").split(' ')
7 groomName = input("What's the name of the groom: ").split(' ')
8
9 getMarried(bridename, groomName)
10
11 # ' '.join(bridename) means joining elements of brideName with ' ' (space)
12 print("Congratulations ", ' '.join(bridename), " and ", ' '.join(groomName))
```


ตัวแปรใน Python มีสองแบบ

Immutable

- การเปลี่ยนแปลงต่อ argument ที่เกิดขึ้นใน function จะคงอยู่เฉพาะในฟังก์ชันเท่านั้น
- ตัวอย่างของตัวแปร Immutable
 - Int, float, long, complex (ใน experiment 1)
 - Str
 - Tuple
 - Bytes

Mutable

- การเปลี่ยนแปลงต่อ argument ที่เกิดขึ้นใน function จะส่งผลต่อภายนอกด้วย
- ตัวอย่างของตัวแปร mutable
 - List (ใน experiment 2)
 - Set
 - Dict

Exercise 1

- จงเขียนฟังก์ชัน `min_of_three(a,b,c)` ซึ่งรับ argument เป็นตัวเลขจำนวนเต็ม สามตัว แล้ว return ตัวเลขที่มีค่าน้อยที่สุดกลับคืนมา
- ตัวอย่าง ผลลัพธ์

```
>>> min_of_three(7,3,4)
3
>>> min_of_three(10,100,-3)
-3
>>> min_of_three(10,0,-9)
-9
```

Exercise 2

- โปรแกรม word processing อย่าง Microsoft office มีฟังก์ชันที่ผู้ใช้สามารถจัดย่อหน้าในสวยงามได้ทั้ง จัดชิดซ้าย จัดชิดขวา และจัดกึ่งกลาง แบบฝึกหัดนี้ ให้เขียนฟังก์ชันชื่อ `right_justify(string)` เพื่อรับข้อความหนึ่งข้อความ (a string) เข้ามา แล้ว ทำการจัดชิดขวา ให้สวยงาม กำหนดให้หน้ากระดาษมีความกว้าง 50 whitespaces และข้อความที่เข้ามามีความยาวไม่เกิน 50 อักขระ ฟังก์ชันนี้ไม่มีการ return ค่ากลับ

Exercise 2: Example

```
>>> right_justify("hello")
                                hello
>>> right_justify("longer than hello")
                                longer than hello
>>> right_justify("longer than the longest hello")
                                longer than the longest hello
>>> |
```

Exercise 3

- ปี 2050 มนุษยชาติค้นพบดาว UN-11S ที่น่าตื่นเต้นกว่านั้นสิ่งมีชีวิตบนดาวนี้ใช้ภาษาอังกฤษในการสื่อสาร ถึงแม้ว่าจะใช้ภาษาเดียวกันแต่วัฒนธรรมทางภาษาไม่เหมือนกัน สิ่งมีชีวิตบนดาวนี้เน้นสุขภาพมาก ฉะนั้นคำว่า butter cake chip hamburger sugar ถือเป็นคำที่ไม่สุขภาพอย่างมาก
- เพื่อการสื่อสารกับสิ่งมีชีวิตดังกล่าวให้เป็นไปอย่างราบรื่น เราจะต้อง censor คำที่มีคำต้องห้าม 5 คำดังกล่าวเป็นส่วนประกอบ โดยหากพบคำที่ว่าในประโยค ให้ทำการ censor โดยแปลงอักขรตัวที่ 2 ถึง $n-1$ เป็นเครื่องหมาย *
- เริ่มจากโครงร่างโปรแกรมที่มีมาให้ จงเขียนเติมฟังก์ชันที่ชื่อ `sensor()` ที่รับประโยคภาษาอังกฤษเข้าไปแล้ว ส่งประโยคที่ทำการเซนเซอร์แล้วกลับคืนมา
- เช่น Cake, cakes, CAKES, หรือ cheesecake ถือว่าเป็นคำไม่สุขภาพเหมือนกัน

Exercise 3: Code template

```
1 def censor():
2     # fill your code here
3
4
5
6
7 def main():
8     sentence = input("Say what: ")
9     censored = censor(sentence)
10    print("Say this instead: ", censored)
11
12 main()
```

Exercise 3 ตัวอย่าง

- Input: I will bake butter cakes for you
- Output: I will bake b****r c***s for you
- Input: I hate fast food
- Output: I hate fast food
- Input: Hamburger or cheeseburger ?
- Output: H*****r or c*****r ?

Homework

- จงเขียนฟังก์ชัน `justify(inputString, alignment)` ที่ทำหน้าที่เหมือนกับ โปรแกรม `right_justify` ในแบบฝึกหัดที่ 2 แต่มีการเพิ่มความสามารถโดย user สามารถเลือกได้ว่าจะให้จัดชิดซ้าย ชิดขวา หรือกึ่งกลาง
- กำหนดให้
 - ความกว้างของหน้ากระดาษเป็น 50 whitespaces
 - ค่าที่เป็นไปได้ของ `alignment` มีดังนี้
 - ใช้ “left” เพื่อสั่งให้จัดชิดซ้าย
 - ใช้ “right” เพื่อสั่งให้จัดชิดขวา
 - ใช้ “center” เพื่อสั่งให้จัดกึ่งกลาง (หากจัดกึ่งกลางไม่พอดี นั่นคือเนื้อที่ว่างทางซ้ายไม่เท่ากับเนื้อที่ว่างทางขวา ให้จัดเยื้องไปทางซ้าย)
 - ข้อความขาเข้า (`inputString`) สามารถมีความยาวมากกว่า 1 บรรทัด โดยแต่ละบรรทัดจะคั่นด้วยเครื่องหมาย `newline` (`'\n'`)
 - ทั้งนี้กำหนดให้ประโยคใดๆมีความยาวตั้งแต่ 1 ถึง 50 อักขระ
- หมายเหตุ มี code template มาให้

Template

```
1 def justify(string, option)
2     # fill your code here
3     #
4     #
5     #
6
7 def readlines():
8     lines = []
9     print("Input your text")
10    while True:
11        line = input("> ")
12        if line:
13            lines.append(line)
14        else:
15            break
16    return '\n'.join(lines)
17
18 def main():
19     document = readlines()
20     option = input("Specify alignment:[left, right, center] ")
21     splt = document.split('\n')
22
23     for i in range(0, len(splt)-1):
24         justify(splt[i], option)
25
26 main()
27
```

References

- Hands-on Python tutorial
 - anh.cs.luc.edu/python/hands-on/3.1/handsonHtml/functions.html
- Think Python: How to Think Like a Computer Scientist