

CSS 112 Computer Programming

Lecture IV: Input, Control, and Conditional Statements

Suvil Chomchaiya (Ph.D.)



Contents

- ⬡ Input Function in Python
- ⬡ Control Statement (Loop)
- ⬡ Conditional Statement

Input Function



Input Function in Python

- ❖ คำสั่ง `input()` ใน Python คือ คำสั่งสำหรับรับค่าจากผู้ใช้งานผ่าน command prompt เพื่อทำการรับ input เข้ามาสู่ระบบ
- ❖ สามารถรับเป็นข้อมูลใดๆ ก็ได้ เช่น
 - ตัวอักษร (Character)
 - ข้อความ (String)
 - ตัวเลข (Number)
 - หรืออักขระพิเศษ (Special Character..เช่น @ \$ &)

Input Function in Python

- ❖ การใช้ `input()` นั้น ทำให้ข้อมูลที่ได้รับเข้ามา จะถูกเปลี่ยนไปเป็นประเภท `string` ทั้งหมด เพื่อป้องกันการเก็บข้อมูลผิดประเภท
- ❖ ถ้าหากต้องการเปลี่ยนประเภทของ `input` เป็นอย่างต่างๆ (เช่น `float`, `boolean`, `integer`) จะต้องทำการ `Data type Casting`

Input Function in Python

❖ ตัวอย่างการใช้คำสั่ง input()

- `text = input("Enter some text ")`
- `number = int(input("Enter a integer: "))`
- `name = input("Enter your name: ")`
`print("Hello " + name)`

❖ การรับค่าเลขทศนิยมใน Python ต้องใช้ float มาใช้งานกับ input() เพื่อกำหนดให้รับค่าข้อมูลที่เป็นเลขทศนิยมได้

- `floating = float(input("Enter a floating point number: "))`

Input Function in Python

- ❖ กรณีที่เขียนโปรแกรมโดยใช้คำสั่ง `input()` ไปแล้วต้องการแปลงค่าข้อมูลที่รับมาเป็นตัวเลขเพื่อนำไปคำนวณ
 - #กำหนดตัวแปร `number` ให้รับค่าข้อมูล
 - `number = input("Enter some text ")`
 - #เปลี่ยนข้อมูลที่รับมาในตัวแปร `number` ให้สามารถแปลงเป็น เลขทศนิยมเพื่อนำไปคำนวณ
 - `anumber = float(number)`
 - #หากต้องการกำหนดให้เป็นจำนวนเต็มแค่เปลี่ยน `float` เป็น `int`


Control Statement (Loop)



Loop in Python

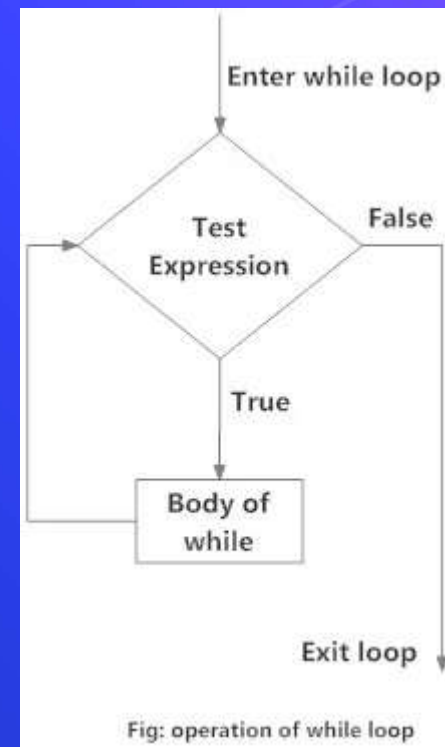
- Loop คือคำสั่งวนซ้ำ ซึ่งคำสั่งเหล่านี้สามารถควบคุมโปรแกรมให้ทำงานซ้ำๆ ในเงื่อนไขที่กำหนดและเพิ่มความสามารถของการเขียนโปรแกรม
- Loop ใน Python มี 2 แบบ คือ FOR และ WHILE

Loop in Python: While Loop

- while loop เป็นคำสั่งวนซ้ำที่ง่ายและพื้นฐานที่สุดในภาษา Python
 - คำสั่ง while loop จะสั่งให้โปรแกรมให้ทำงานซ้ำๆ ตามที่ต้องการ ในขณะที่เงื่อนไขยังคงเป็นจริง
 - ในรูปแบบการใช้งานคำสั่ง while loop นั้น จะประกอบด้วยคำสั่ง while และตามด้วยการกำหนด expression ซึ่งเป็นเงื่อนไขที่จะให้โปรแกรมทำงาน
-  โปรแกรมจะทำงานจนกว่าเงื่อนไขจะเป็น False และสิ้นสุดการทำงานของลูป

Loop in Python: While Loop

Flowchart ของ While Loop



Source: www.programiz.com

Loop in Python: While Loop

- ❖ รูปแบบทั่วไปของ While Loop

```
while expression:  
    # statements
```

- ❖ ภายในบล็อกคำสั่ง while นั้นประกอบไปด้วยคำสั่งการทำงานของโปรแกรม

Loop in Python: While Loop

- ❖ ตัวอย่างโปรแกรมในการแสดงตัวเลข 1 ถึง 10 โดยใช้คำสั่ง while loop

```
i = 1
while i <= 10:
    print(i, end = ', ')
    i = i + 1
```

1, 2, 3, 4, 5, 6, 7, 8, 9, 10,

ผลลัพธ์

Loop in Python: While Loop

- ตัวอย่างโปรแกรมในการลดตัวเลขที่เริ่มต้นด้วย 50 แล้วลงทีละ 5 จนเหลือ 0 โดยใช้คำสั่ง while loop

```
i = 50
while i >= 0:
    print(i, end = ', ')
    i = i - 5
```

50, 45, 40, 35, 30, 25, 20, 15, 10, 5, 0,

ผลลัพธ์

- ตัวอย่างโปรแกรมในการแสดงลำดับของอนุกรมเลขคณิตจากสูตร $2n + 1$ โดย n มีค่าตั้งแต่ 1 ถึง 10

```
while n <= 10:
    print(2 * n + 1, end = ', ')
    n = n + 1
```

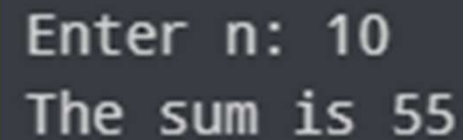
3, 5, 7, 9, 11, 13, 15, 17, 19, 21,

ผลลัพธ์

Loop in Python: While Loop

❖ ตัวอย่างโปรแกรมในการรวมผลตัวเลข 1+2+3+4+5..... จนถึง 10

```
# To take input from the user,  
    int(input("Enter n: "))  
# initialize sum and counter  
    sum = 0  
    i = 1  
  
    while i <= n:  
        sum = sum + i  
        i = i+1  # update counter  
  
# print the summation  
print("The sum is", sum)
```



Enter n: 10
The sum is 55

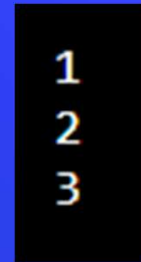
ผลลัพธ์: กรณีผู้ใช้งานป้อนค่า 10

Loop in Python: While Loop

while loop สามารถหยุดการทำงานโดยใช้คำสั่ง break

ตัวอย่าง:

```
i = 1
while i < 6:
    print(i)
    if i == 3:
        break
    i += 1
```



1
2
3

ผลลัพธ์: โปรแกรมหยุดการทำงานที่ค่า = 3
ถึงแม้ว่าตั้งค่าตอนแรกไว้ไม่เกิน 6

Loop in Python: While Loop

while loop สามารถข้ามการทำงานของ loop ปัจจุบัน โดยใช้คำสั่ง `continue`

```
i = 0
while i < 6:
    i += 1
    if i == 3:
        continue
    print(i)
```

1
2
4
5
6

ผลลัพธ์: โปรแกรมกระโดดข้ามการทำงานที่ค่า = 3
แต่ยังทำงานต่อไปจนครบเงื่อนไข

Loop in Python: While Loop

while loop สามารถใช้งานร่วมกับคำสั่ง else

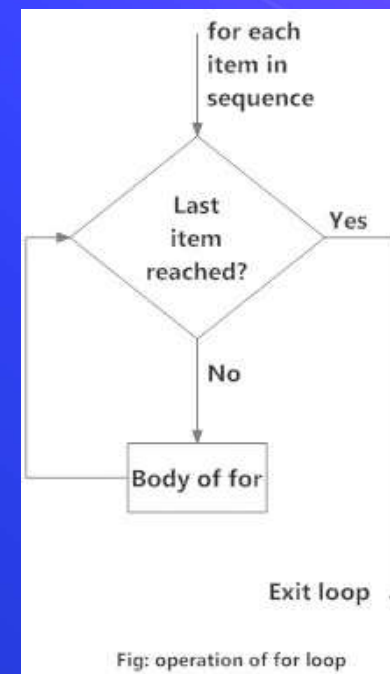
```
counter = 0
while counter < 3:
    print("Inside loop")
    counter = counter + 1
else:
    print("Inside else")
```

```
Inside loop
Inside loop
Inside loop
Inside else
```

ผลลัพธ์: โปรแกรมแสดงผลจนครบเงื่อนไข 3 ครั้ง
จากนั้นก็แสดงผลตามคำสั่ง else

Loop in Python: For Loop

Flowchart ของ While Loop



Source: www.programiz.com

Loop in Python: For Loop

- ❖ รูปแบบทั่วไปของ For Loop

```
for val in sequence:  
    loop body
```

- ❖ คำสั่ง for loop เป็นคำสั่งวนซ้ำที่ใช้ควบคุมการทำงานซ้ำๆ ในจำนวนรอบที่แน่นอน

Loop in Python: For Loop

ตัวอย่างโปรแกรม

```
# loop through string
```

```
site = 'marcuscode'
```

```
for n in site:
```

```
    print(n)
```

```
# loop through list
```

```
names = ['Mateo', 'John', 'Eric', 'Mark', 'Robert']
```

```
for n in names:
```

```
    print(n)
```

```
numbers = [10, 20, 30, 40, 50, 60, 70, 80]
```

```
for n in numbers:
```

```
    print(n)
```

Loop in Python: For Loop

ตัวอย่างโปรแกรม

```
# Program to find the sum of all numbers stored in a list
```

```
# List of numbers
```

```
numbers = [6, 5, 3, 8, 4, 2, 5, 4, 11]
```

```
# variable to store the sum
```

```
sum = 0
```

```
# iterate over the list
```

```
for val in numbers:
```

```
    sum = sum+val
```

```
print("The sum is", sum)
```

The sum is 48

ผลลัพธ์

Loop in Python: Nested Loop

ตัวอย่างโปรแกรม

```
adj = ["red", "big", "tasty"]  
fruits = ["apple", "banana", "cherry"]
```

```
for x in adj:  
    for y in fruits:  
        print(x, y)
```

```
red apple  
red banana  
red cherry  
big apple  
big banana  
big cherry  
tasty apple  
tasty banana  
tasty cherry
```

ผลลัพธ์

Conditional Statement



Conditional Statement in Python

- Conditional Statement เป็นการควบคุมทิศทางการทำงานของโปรแกรม เพื่อให้โปรแกรมที่เขียนสามารถทำงานตามลำดับตามที่ได้ออกแบบไว้

Conditonal Statement in Python

Control Statement ใน Python:

- คำสั่ง **if**: ใช้เพื่อสร้างเงื่อนไขให้โปรแกรมทำงานตามที่ต้องการเมื่อเงื่อนไขนั้นตรงกับที่กำหนด
- คำสั่ง **if...else** โปรแกรมจะทำงานในคำสั่ง **else** ถ้าหากเงื่อนไขในคำสั่ง **if** นั้นไม่เป็นจริง
- คำสั่ง **elif** นั้นเป็นคำสั่งที่ใช้สำหรับสร้างเงื่อนไขแบบหลายทางเลือกให้กับโปรแกรมที่มึการทำงานเช่นเดียวกับ **switch case** ในภาษาอื่นๆ
 - คำสั่ง **elif** นั้นต้องใช้หลังจากคำสั่ง **if** เสมอและสามารถมี **else** ได้ในเงื่อนไขสุดท้าย

Conditional Statement in Python: if

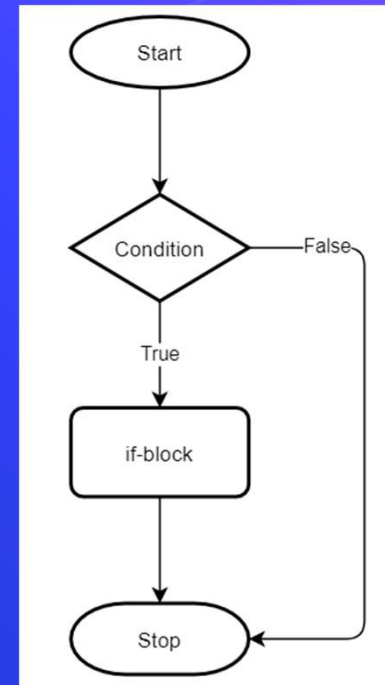
- ❖ คำสั่ง if เป็นคำสั่งที่ใช้ควบคุมการทำงานของโปรแกรมที่เป็นพื้นฐานและง่ายที่สุด
- ❖ ใช้คำสั่ง if เพื่อสร้างเงื่อนไขให้โปรแกรมทำงานตามที่ต้องการเมื่อเงื่อนไขนั้นตรงกับที่กำหนด
- ❖ if ใช้สำหรับเงื่อนไขเดียว
- ❖ รูปแบบทั่วไป

```
if expression:  
    # statements
```

ต้องมี **Indentation!!!!**

Conditional Statement in Python: if

Flowchart ของ if



Conditional Statement in Python: if

ตัวอย่างโปรแกรม

```
n = 10
if n == 10:
    print('n equal to 10')
logged_in = False
if not logged_in:
    print('You must login to continue')
m = 4
if m % 2 == 0 and m > 0:
    print('m is even and positive numbers')
if 3 > 10:
    print('This block isn\'t executed')
```

```
n equal to 10
You must login to post
m is even and positive numbers
```

ผลลัพธ์

Conditional Statement in Python: if

ตัวอย่างโปรแกรม

```
age = input('Enter your age:')  
if int(age) >= 18:  
    print("You're eligible to vote.")  
    print("Let's go and vote.")
```

```
Enter your age:21  
You're eligible to vote.  
Let's go and vote.
```

ผลลัพธ์

Conditional Statement in Python: if

ตัวอย่างโปรแกรม

```
age = input('Enter your age:')  
if int(age) >= 18:  
    print("You're eligible to vote.")  
    print("Let's go and vote.")
```

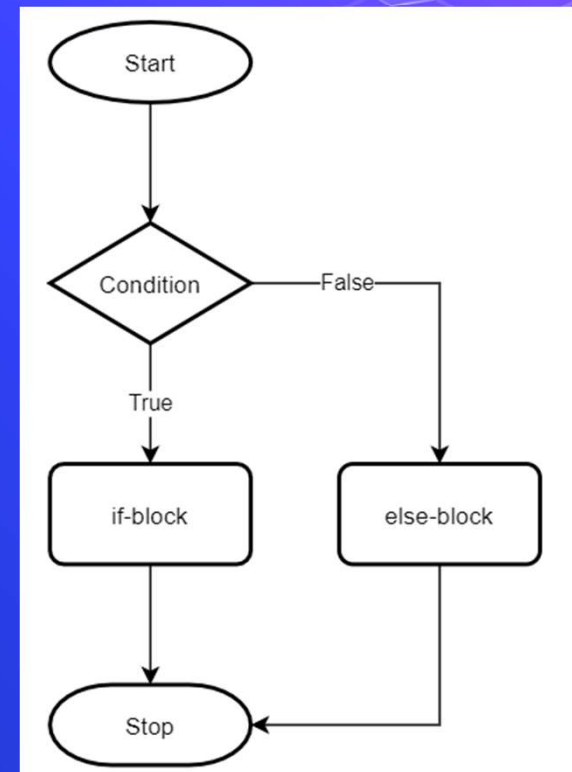
Try this!!!!!!

Conditional Statement in Python: if...else

- if - else ใช้สำหรับ 2 เงื่อนไข
- โปรแกรมจะทำงานในคำสั่ง else ถ้าหากเงื่อนไขในคำสั่ง if นั้นไม่เป็นจริง
- else จะเป็นเสมือนเงื่อนไขที่เป็น Default เมื่อเงื่อนไขก่อนหน้านี้ทั้งหมดไม่เป็นจริง

Conditional Statement in Python: if...else

Flowchart ของ if...else



Conditional Statement in Python: if...else

ตัวอย่างโปรแกรม

```
age = input('Enter your age:')  
if int(age) >= 18:  
    print("You're eligible to vote.")  
else:  
    print("You're not eligible to vote.")
```

```
Enter your age:11  
You're not eligible to vote.
```

ผลลัพธ์

Conditional Statement in Python: if...else

ตัวอย่างโปรแกรม

```
n = 5
if n == 10:
    print('n equal to 10')
else:
    print('n is something else except 10')
name = 'James'
if name == 'Mateo':
    print('Hi, Mateo.')
else:
    print('Who are you?')
money = 300
if money >= 350:
    print('You can buy an iPad')
else:
    print('You don\'t have enough money to buy an iPad')
```

```
n is something else except 10
Who are you?
You don't have enough money to buy an iPad
```

ผลลัพธ์

Conditional Statement in Python: if...else

ตัวอย่างโปรแกรม

```
var1 = 100
if var1:
    print "1 - Got a true expression value"
    print var1
else:
    print "1 - Got a false expression value"
    print var1
```

```
var2 = 0
if var2:
    print "2 - Got a true expression value"
    print var2
else:
    print "2 - Got a false expression value"
    print var2
```

```
1 - Got a true expression value
100
2 - Got a false expression value
0
Good bye!
```

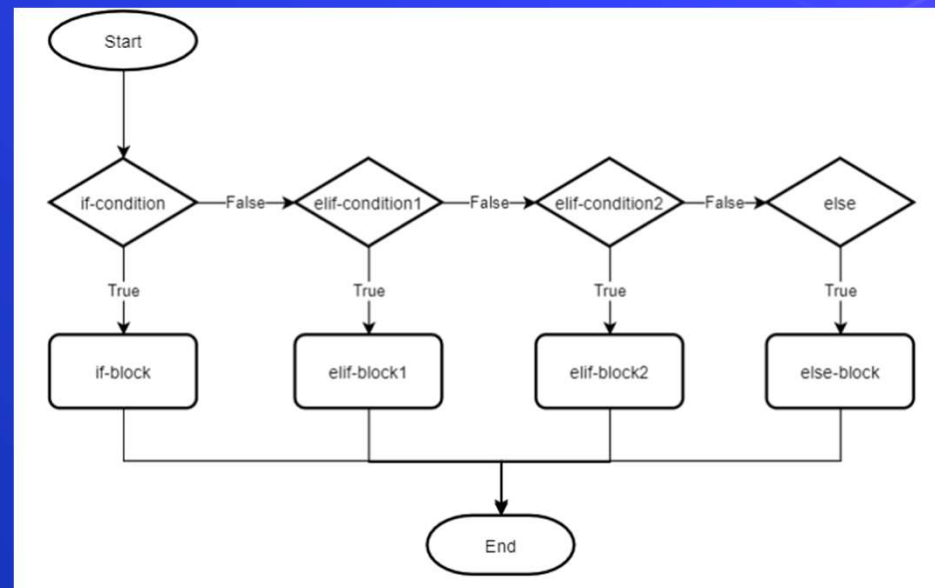
ผลลัพธ์

Conditional Statement in Python: if...elif

- ❖ คำสั่ง elif นั้นเป็นคำสั่งที่ใช้สำหรับสร้างเงื่อนไขแบบหลายทางเลือกให้กับโปรแกรมที่มีการทำงาน
- ❖ เทียบเท่า switch case ในภาษาอื่นๆ
- ❖ คำสั่ง elif นั้นต้องใช้หลังจากคำสั่ง if เสมอและสามารถมี else ได้ในเงื่อนไขสุดท้าย

Conditional Statement in Python: if...elif

Flowchart ของ if...elif



Conditional Statement in Python: if...elif

ตัวอย่างโปรแกรม

```
print('Welcome to CSS 112 course')  
level = input('Enter level (1 - 4): ')
```

```
if level == '1':  
    print('Easy')  
elif level == '2':  
    print('Medium')  
elif level == '3':  
    print('Hard')  
elif level == '4':  
    print('Expert')  
else:  
    print('Invalid level selected')
```

```
Welcome to CSS 112 Course  
Enter level (1 - 4): 2  
Medium
```

ผลลัพธ์

Conditional Statement in Python: if...elif

ตัวอย่างโปรแกรม

```
age = input('Enter your age:')  
# convert the string to int  
your_age = int(age)  
# determine the ticket price  
if your_age < 5:  
    ticket_price = 5  
elif your_age < 16:  
    ticket_price = 10  
else:  
    ticket_price = 18  
# show the ticket price  
print(f'You'll pay ${ticket_price} for the ticket')
```

```
Enter your age:30  
You'll pay $18 for the ticket
```

ผลลัพธ์

Conditional Statement in Python:

เงื่อนไขทางตรรกศาสตร์ ที่ Conditional Statement รองรับ

- ⬡ Equals: $a == b$
- ⬡ Not Equals: $a != b$
- ⬡ Less than: $a < b$
- ⬡ Less than or equal to: $a \leq b$
- ⬡ Greater than: $a > b$
- ⬡ Greater than or equal to: $a \geq b$

Conditional Statement in Short Form

Short Hand If

- Example:
 - `if a > b: print("a is greater than b")`

Short Hand If ... Else

- Example:
 - `a = 2`
 - `b = 330`
 - `print("A") if a > b else print("B")`

Conditional Statement and Logical Operators

Test if a is greater than b, **AND** if c is greater than a:

- Example:

```
a = 200
```

```
b = 33
```

```
c = 500
```

```
if a > b and c > a:
```

```
    print("Both conditions are True")
```

Conditional Statement and Logical Operators

Test if a is greater than b, **OR** if a is greater than c:

- Example:

```
a = 200
```

```
b = 33
```

```
c = 500
```

```
if a > b or a > c:
```

```
    print("At least one of the conditions is True")
```

Conclusion

- Control Statement (Loop) คือคำสั่งวนซ้ำ ซึ่งคำสั่งเหล่านี้สามารถควบคุมโปรแกรมให้ทำงานซ้ำๆ ในเงื่อนไขที่กำหนด
- Conditional Statement เป็นการควบคุมทิศทางการทำงานของโปรแกรม
- ใน Python Conditional Statement สามารถใช้งานร่วมกับ Logical and Arithmetic Operator ได้

The End.....
Questions or Comments????