SCHOOL OF
INFORMATION TECHNOLOGY
AND INNOVATION
BANGKOK UNIVERSITY

Python Iteration LOOP

CS310 : Computer Programming I

Sirinthorn Cheyasak, Asst. Prof.

Table of Contents

01 **Iteration/LOOP**
Getting to know python iteration and Loop.

02 **Python For Loop**
Python for loop concept and how to write a repletely program.

03 **Python List/ Tuple**
Python List concept and case of use python list. Python tuple concept and case of use python tuple.

04 **Class Activity**
Here you could describe the topic of the section

python

Iteration LOOP

01

Getting to know python iteration and Loop.

Iteration and Loop Concept

Iteration

Repetitive execution of the same block of code over and over is referred to as iteration.

Iteration คือ กระบรวนการที่มีการเริ่มต้น, การทำซ้ำ, และเงื่อนไขที่ก่อให้เกิดจุดจบของกระบรวนการนั้น เรียกว่า กระบรวนการ Iteration

Loop

Loop is a sequence of the instructions that is continually repeated until a certain condition is reached.

Loop คือ code ที่เราเขียนลงบนโปรแกรมให้เกิดกระบวนการ Iteration

Iteration Types

Iteration Types

Definite Iteration

Iteration, in which the number of repetitions is specified explicitly in advance.
จำนวนรอบของการทำซ้ำได้ถูกกำหนดไว้แล้วว่าจะให้ทำซ้ำทั้งหมดกี่ครั้ง

FOR LOOP

Infinite Loop

ระวังอย่าให้เกิดการวนไม่รู้จบ

Indefinite Iteration

Iteration, in which the code block executes until some condition is met.
จำนวนรอบของการทำซ้ำจะขึ้นอยู่กับเงื่อนไข ว่าตรงกับเงื่อนไขมีทั้งหมดกี่ครั้ง ก็จะทำซ้ำไปตามจำนวนครั้งที่ตรงกับเงื่อนไขนั้น

WHILE LOOP

02 Python FOR Loop

Python "for" Loops (Definite Iteration)

Range()

The Python range() function returns the sequence of the given number between the given range. The most common use of it is to iterate sequence type (Python range() List, string, etc.)

Syntax of range()

Parameter:

- start: integer starting from which the sequence of integers is to be returned
- stop: integer before which the sequence of integers is to be returned. The range of integers ends at stop – 1.
- step: integer value which determines the increment between each integer in the sequence

Python For Loop Syntax

SYNTAX: range(start, stop, step)

Example:

x = range(6) -> 0,1,2,3,4,5

x = range(3,6) -> 3,4,5

x = range(3, 20, 2) -> 3, 5, 7, 9, 11, 13, 15, 17, 19

Python For Loop Example

```
#printing hello and CS310
for i in range(1,6,1):
    print("Hello.")
for i in range(5) :
    print(“CS310”)
print("END")
```

The Output

```
Hello.
Hello.
Hello.
Hello.
Hello.
CS310.
CS310.
CS310.
CS310.
CS310.
END
```

Algorithm to calculate the average of first n natural numbers

1. Allows a user to enter the number (n) he wishes to calculate the sum and average. The program accepts user input using the **input function**.
2. Next, run loop till the entered number using the for loop and **range() function**.
3. Next, calculate the sum using a **sum = sum + current number** formula.
4. At last, after the loop ends, calculate the average using **average = sum / n**.
   n is a number entered by the user.

Python program to calculate the average of first n natural numbers

```
n = int(input("Enter ending number : "))
sum = 0
for num in range(0, n+1, 1):
    sum = sum+num
print("SUM of first ", n, "numbers is: ", sum )
```

03

Python

List/Tuple

Python Collection Data Types

Python LIST

โครงสร้างที่ใช้จัดเก็บข้อมูลแบบลำดับ (Sequence) โดยชนิดข้อมูลเหล่านี้สามารถเก็บข้อมูลต่างประเภทกันได้ ทั้งชนิดข้อมูลจำนวนเต็ม หรือชนิดข้อมูลสตริง นอกจากนี้ยังสามารถที่จะเก็บชนิดข้อมูลลิสต์ซ้อนลิสต์ หรือลิสต์ซ้อนทูเพิลก็ได้ ข้อมูลลิสต์สามารถทำการเพิ่ม ลบ แก้ไข ข้อมูลได้

Python Tuple

โครงสร้างที่ใช้จัดเก็บข้อมูลแบบลำดับ (Sequence) โดยชนิดข้อมูลเหล่านี้สามารถเก็บข้อมูลต่างประเภทกันได้ ทั้งชนิดข้อมูลจำนวนเต็ม หรือชนิดข้อมูลสตริง นอกจากนี้ยังสามารถที่จะเก็บชนิดข้อมูลทูเพิลซ้อนทูเพิล หรือทูเพิลซ้อนลิสต์ก็ได้ ส่วนชนิดทูเพิลไม่สามารถทำการเปลี่ยนแปลงข้อมูลได้

Python List Characteristics

- Lists Are Ordered
- List Elements Can Be Accessed by Index
- Lists Are Dynamic
- Lists Are Mutable
- Lists Can Be Nested

Creating Python List

ชนิดข้อมูลลิสต์คล้ายกับข้อมูลอาร์เรย์ในภาษาอื่นๆ การสร้างตัวแปรขึ้นมาเก็บชนิดข้อมูลลิสต์ก็ไม่แตกต่างกับการสร้างตัวแปรชนิดข้อมูลอื่นๆ เพียงแต่ข้อมูลที่เก็บในลิสต์ต้องอยู่ในเครื่องหมาย [...] และใช้เครื่องหมาย comma (,) คั่นระหว่างข้อมูลในลิสต์

ตัวอย่างการสร้าง List

```
numbers = [-1, 2, 5, 8, 10, 13]
names = ['Mateo', 'Danny', 'James', 'Thomas', 'Luke']
mixed_type = [-2, 5, 84.2, "Mountain", "Python"]
print("List number =", numbers)
print("List names = ",names)
print("Mixed type = ",mixed_type)
```

ผลลัพธ์

```
List number = [-1, 2, 5, 8, 10, 13]
List names =  ['Mateo', 'Danny', 'James', 'Thomas', 'Luke']
Mixed type =  [-2, 5, 84.2, 'Mountain', 'Python']
```

Creating Python List

```
# empty list
mylist1 = []

# list of integers
mylist2 = [1, 2, 3]

# list with mixed data types
mylist3 = [1, "Hello", 3.4]
```

Accessing Elements on List

There are various ways in which we can access the elements of a tuple

1. List Index
2. Negative Index
3. Slice List

Accessing : List Index

- You can use the **index operator** **[ ]** to access an item in a list. In Python, indices start at 0. So, a list having 5 elements will have an index from 0 to 4.
- Trying to access indexes other than these will raise an **IndexError**. The index must be an integer. We can't use float or other types, this will result in **TypeError**.

```
mylist = ['p', 'r', 'o', 'b', 'e’]
# Output: p
print(mylist[0])
# Output: o
print(mylist[2])
# Output: e
print(mylist[4])
```

Accessing : Negative Index

- Python allows negative indexing for its sequences. The index of -1 refers to the last item, -2 to the second last item and so on.

```
# Negative indexing in lists
mylist = ['p','r','o','b', 'e']

print(mylist[-1])
#output e
print(mylist[-5])
#output p
```

Negative index	-5	-4	-3	-2	-1
mytuple	'p'	'r'	'o'	'b'	'e'
index	0	1	2	3	4

Accessing : Slice List

- We can access a range of items in a list by using the slicing **operator : (colon)**

```
variable[start : stop : step]
```

```
names = ['Mateo', 'Danny', 'James', 'Thomas', 'Luke']
```

Negative index	-5	-4	-3	-2	-1
names	Mateo	Danny	James	Thomas	Luke
index	0	1	2	3	4

Slicing List Example

Negative index	-5	-4	-3	-2	-1
names	Mateo	Danny	James	Thomas	Luke
index	0	1	2	3	4

```
names =  ['Mateo', 'Danny', 'James', 'Thomas', 'Luke']
print("index  -3 in names list >> ",names[-3])
print("index  3 in names list >> ",names[3])
print("index  0-3 in names list >> ",names[:4])
print("index  1-3 in names list >> ",names[1:4])
print("index  1-3 in names list 2 step >> ",names[1:4:2])
print("names list 2 step >> ",names[::2])
```

```
index  -3 in names list >>  James
index 3 in names list >> Thomas
index  0-3 in names list >>  ['Mateo', 'Danny', 'James', 'Thomas']
index  1-3 in names list >>  ['Danny', 'James', 'Thomas']]
index  1-4 in names list 2 step >>  ['Danny', 'Thomas']
names list 2 step >>  ['Mateo', 'James', 'Luke']
```

Operation on List

There are various ways in which we can access the elements of a list

1. Add a new element on list
2. Update an element on list
3. Remove an element on list

1 Add a new element on list

- We can add one item to a list using the **append()** method or add several items using **extend()** method.

Syntax	*list*.**append** (new_*element*)

```
# Appending and Extending lists in Python
mylist = [1, 3, 5]

mylist.append(7)
print(mylist)
```

→ [1, 3, 5, 7]

```
mylist.extend([9, 11, 13])
print(mylist)
```

→ [1, 3, 5, 7, 9, 11, 13]

https://www.programiz.com/python-programming/list

1 Insert a new element on list

- we can insert one item at a desired location by using the method **insert()** or insert multiple items by squeezing it into an empty slice of a list.

Syntax	*list*.**insert** (index,new_*element*)

```
# Appending and Extending lists in Python
prime_numbers = [2, 3, 5, 7]

# insert 11 at index 4
prime_numbers.insert(4, 11)

prime_numbers.insert(5, 13)

print('List:', prime_numbers)
```

List: [2, 3, 5, 7, 11, 13]

https://www.programiz.com/python-programming/list

2 Update an element on list

- we can update one item at a desired location by using assignment statement.

Syntax	listname[index] = new value

```
list = ['physics', 'chemistry', 1997, 2000]
print ("Value at index 2 : ",end="")
print (list[2])
```

Value at index 2 : 1997

```
list[2] = 2001
print ("New value at index 2 : " ,end="")
print (list[2])
```

New value at index 2 : 2001

https://www.programiz.com/python-programming/list

3 Remove an element on list

- The **remove(value)** method removes the first occurrence of the given value from the list. There must be one occurrence of the provided value, otherwise the Python raises ValueError.

Syntax	*list*.**remove** (value)

```
list = ['physics', 'chemistry', 1997, 2000]
list.remove(1997)
print(list)
list.remove("2000")
print(list)
```

→ ['physics', 'chemistry', 2000]

→ ValueError: list.remove(x): x not in list

https://www.programiz.com/python-programming/list

3 Pop an element on list

- The **pop()** method removes an element on list at position of index.

Syntax	*list*.**pop** (index)

```
list = ['physics', 'chemistry', 1997, 2000]
list.pop(2)
print(list)
list.pop(5)
print(list)
```

→ ['physics', 'chemistry', 2000]

→ IndexError: pop index out of range

https://www.programiz.com/python-programming/list

Clear all of elements on list

- We can also use the **clear()** method to empty a list.

Syntax	*list*.**clear**()

```
list = ['physics', 'chemistry', 1997, 2000]
mylist = ['p','r','o','b','l','e','m']
list.clear()
print(list)
mylist.clear()
print(list)
```

→ []

→ []

https://www.programiz.com/python-programming/list

Built-in List Functions & Methods

There are various ways in which we can access the elements of a list

1. List length method
2. List index method
3. List count method
4. List sort method
5. List sum,max,min method

https://www.tutorialspoint.com/python/python_lists.htm

1 List Length Method

- Python list method **len()** returns the number of elements in the *list*.

Syntax	len(list)

```
list1 = [123, 'xyz', 'zara']
list2 = [456, 'abc']

print ("List 1 length : ", len(list1))
print ("List 2 length : ", len(list2))
```

List 1 length : 3

List 2 length : 2

https://www.programiz.com/python-programming/list

2 List Index Method

- The **index()** method returns the index of the specified element in the list.

Syntax	list.index(value)

```
list1 = [123, 'xyz', 'zara']
list2 = [456, 'abc']

print (" zara : ", list1.index("zara"))
print (" 456 : ", list2.index(456))
```

zara : 2

456 : 0

https://www.programiz.com/python-programming/list

3 List Count Method

- Python list method **count()** returns count of how many times *element* occurs in list.

Syntax	*list*.**count** (element)

```
list = [123, “xyz”, “zara”, “abc”, 123]

cnt = list.count(123)
print ("Count for 123 : ",cnt)

print ("Count for zara : ", list.count('zara'))
```

Count for 123 : 2

Count for zara : 1

https://www.programiz.com/python-programming/list

4 List Sort Method

- Python list method **sort()** sorts objects of list.

Syntax	*list*.**sort** ()

```
mylist1 = ['123', 'xyz', 'zara', 'abc', 'xyz']
mylist1.sort()
print("List : ", mylist1)
```

List : ['123', 'abc', 'xyz', 'xyz', 'zara']

```
mylist2 = [44, 99, 33, 88, 11]
mylist2.sort()
print("List : ", mylist2)
```

List : [11, 33, 44, 88, 99]

```
mylist2 = [44, 99, 33, 88, 11]
mylist2.sort(reverse=True)
print("List : ", mylist2)
```

List : [99, 88, 44, 33, 11]

https://www.programiz.com/python-programming/list

5 List sum,max,min method

Syntax	
	max(list) : finding maximum elements
	min(list) : finding minimum elements
	sum(list) : finding summation all of elements

```
mylist = [23,44,56,78,99,12,11]

print("Maximum = ",max(mylist))
print("Minimum = ",min(mylist))
print("Summation = ",sum(mylist))
```

→ Maximum = 99

→ Minimum = 11

→ Summation = 323

https://www.programiz.com/python-programming/list

Python Tuple

Python Tuple Characteristics

- Tuples Are Ordered
- Tuples Are Immutable
- Tuples Can Be Nested
- Tuples Elements Can Be Accessed by Index
- Tuples are Static.

Creating a Tuple

```
# empty tuple
tuple1 = ()

# tuple of integers
tuple2 = (1, 2, 3)

# tuple with mixed data types
tuple3 = (1, "Hello", 3.4)
```

Creating a Tuple

- ชนิดข้อมูลทูเบิลคล้ายกับข้อมูลอาร์เรย์ในภาษาอื่นๆ การสร้างตัวแปรขึ้นมาเก็บชนิดข้อมูลทูเบิลก็ไม่แตกต่างกับการสร้างตัวแปรชนิดข้อมูลอื่นๆ เพียงแต่ข้อมูลที่เก็บในทูเบิลต้องอยู่ในเครื่องหมาย (...) และใช้เครื่องหมาย comma (,) คั่นระหว่างข้อมูลในทูเบิล

ตัวอย่างการสร้าง Tuple

```
empty_tuple = ()
any_tuple = ("Name", "Address", [1,2,3,4], ("ID", 3452))
fruit_tuple = ("apple", "banana", "cherry")
print("empty tuple = ", empty_tuple)
print("any tuple = ", any_tuple)
print("fruits tuple = ", fruit_tuple)
```

ผลลัพธ์

```
empty tuple =  ()
any tuple =  ('Name', 'Address', [1, 2, 3, 4], ('ID', 3452))
fruits tuple =  ('apple', 'banana', 'cherry')
```

Accessing Elements on Tuple

There are various ways in which we can access the elements of a tuple

1. Tuple Index
2. Negative Index
3. Slice Tuple

1 Tuple Index

- You can use the **index operator** [ ] to access an item in a tuple. In Python, indices start at 0. So, a list having 5 elements will have an index from 0 to 4.
- Trying to access indexes other than these will raise an IndexError. The index must be an integer. We can't use float or other types; this will result in TypeError.

```
mytupple = ('p', 'r', 'o', 'b', 'e')
# Output: p
print(mytupple[0])
# Output: o
print(mytupple[2])
# Output: e
print(mytupple[4])
```

https://www.programiz.com/python-programming/list

2 Negative Index

- Python allows negative indexing for its sequences. The index of -1 refers to the last item, -2 to the second last item and so on.

```
# Negative indexing in lists
mytuple = ('p','r','o','b', 'e')

print(mytuple[-1])
#output e
print(mytuple[-5])
#output p
```

Negative index	-5	-4	-3	-2	-1
mytuple	'p'	'r'	'o'	'b'	'e'
index	0	1	2	3	4

https://www.programiz.com/python-programming/list

3 Slicing on a Tuple

We can access a range of items in a list by using the slicing operator :(colon)

```
variable[start : stop : step]
```

```
names = ('Mateo', 'Danny', 'James', 'Thomas', 'Luke')
```

Negative index	-5	-4	-3	-2	-1
names	Mateo	Danny	James	Thomas	Luke
index	0	1	2	3	4

3 Slicing List Example

Negative index	-5	-4	-3	-2	-1
names	Mateo	Danny	James	Thomas	Luke
index	0	1	2	3	4

```
names =  ('Mateo', 'Danny', 'James', 'Thomas', 'Luke')
print("index  -3 in names tuple >> ",names[-3])
print("index  3 in names tuple >> ",names[3])
print("index  0-3 in names tuple >> ",names[:4])
print("index  1-3 in names tuple >> ",names[1:4])
print("index  1-4 in names tuple 2 step >> ",names[1:4:2])
print("names list 2 tuple >> ",names[::2])
```

```
index  -3 in names list >>  James
index 3 in names list >> Thomas
index  0-3 in names list >>  ['Mateo', 'Danny', 'James', 'Thomas']
index  1-3 in names list >>  ['Mateo', 'Danny', 'James']
index  1-4 in names list 2 step >>  ['Danny', 'Thomas']
names list 2 step >>  ['Mateo', 'James', 'Luke']
```

Built-in List Functions & Methods

There are various ways in which we can access the elements of a list

1. Tuple length method
2. Tuple count method
3. Tuple sum,max,min method

https://www.tutorialspoint.com/python/python_lists.htm

1 **Tuple** Length Method

- Python list method **len()** returns the number of elements in the *tuple*.

Syntax	len(tuple)

```
tuple1 = (123, 'xyz', 'zara')
tuple2 = (456, 'abc')

print ("Tuple1 1 length : ", len(tuple1))
print ("Tuple 2 length : ", len(tuple2))
```

Tuple 1 length : 3

Tuple 2 length : 2

https://www.programiz.com/python-programming/list

2 List Count Method

- Python list method **count()** returns count of how many times *element* occurs in tuple.

Syntax	*tuple*.**count** (element)

```
tuple = (123, “xyz”, “zara”, “abc”, 123)

cnt = tuple.count(123)
print ("Count for 123 : ",cnt)

print ("Count for zara : ", tuple.count('zara'))
```

Count for 123 : 2

Count for zara : 1

https://www.programiz.com/python-programming/list

3 List sum,max,min method

Syntax	
	max(tuple) : finding maximum elements
	min(tuple) : finding minimum elements
	sum(tuple) : finding summation all of elements

```
tuple1 = (23,44,56,78,99,12,11)

print("Maximum = ",max(tuple1))
print("Minimum = ",min(tuple1))
print("Summation = ",sum(tuple1))
```

→ Maximum = 99

→ Minimum = 11

→ Summation = 323

https://www.programiz.com/python-programming/list

Thank you

Any Question?

You can contact me sirinthorn.c@bu.ac.th

Or MS Team Chat