IT Department

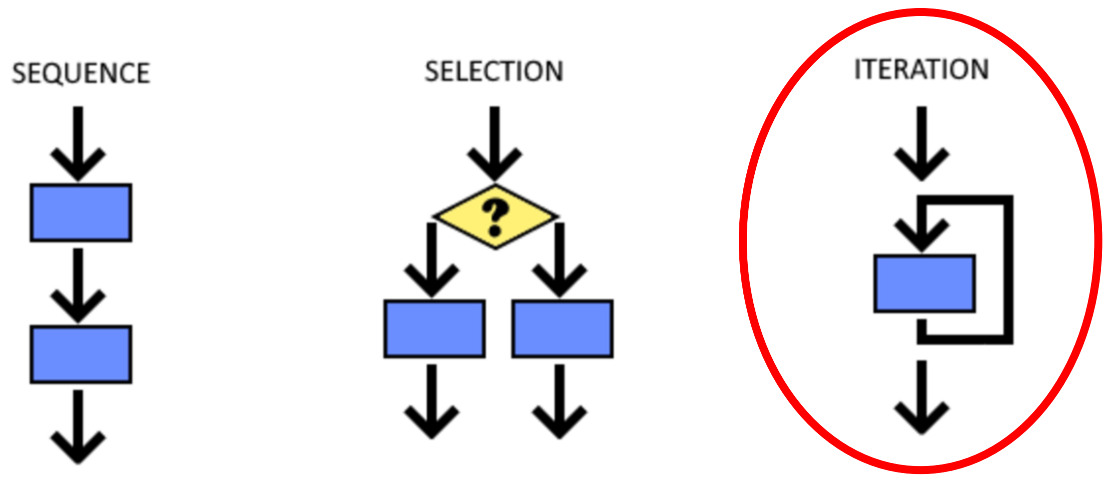
Loops and Iteration (for loop)

Soma Soleiman Zadeh
Programming I (IT 117)
Fall 2024 - 2025
Week 8
February 5, 2025

Outline

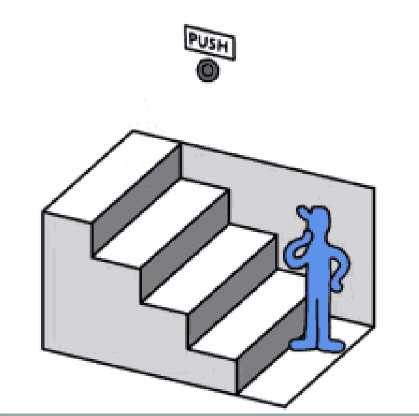
- Control Structures
- **Loops**
  - **Definite** Loop (**Counting** Loop)
  - **Indefinite** Loop (**Conditional** Loop)
- **For** Loop
- **range( )** Function

ITERATION Control Structure



What is Loop in Programming?

- **Loops** are a way to **repeat a set of actions** a specific number of times under certain conditions.
- A **loop** takes a few lines of code, and runs them again and again.



Definite Loops vs. Indefinite Loops

- There are two types of loops in Python:

  1. **Definite** (Counting) Loops → **for** loop
     - **Exact number of iterations** to do.
     - Iterates through the members of a set (set of numbers, characters, strings).
  2. **Indefinite** (Conditional) loops → **while** loop
     - Not definite number of iterations.
     - Iterates while some condition is **True**.

for Loop

- A **for-loop** is a set of instructions that is repeated, or iterated, for every value in a sequence.
  - **Body** of loop → The code that is repeated in a loop
  - **Iteration** of the loop → Each repetition of the loop body
- General syntax of **for** loop:

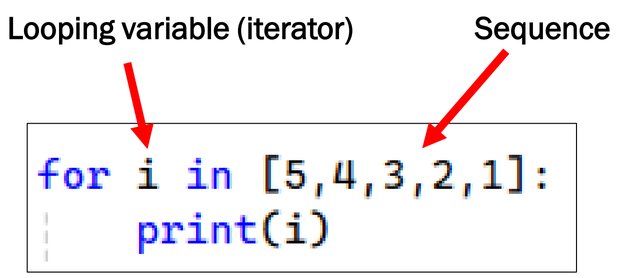
```
for  looping_variable  in  sequence :
        code block
```

What does happen in a for loop?

1. A for-loop assigns the looping variable to the first element of the sequence. It executes everything in the code block.
2. Then it assigns the looping variable to the next element of the sequence and executes the code block again.
3. It continues until there are no more elements in the sequence to assign.

```
for looping_variable in sequence :
        code block
```

A Simple Example of for Loop

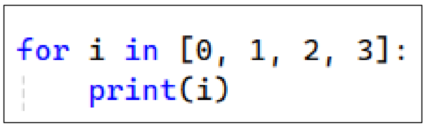


```
for i in [5,4,3,2,1]:
    print(i)
```

Output

```
5
4
3
2
1
```

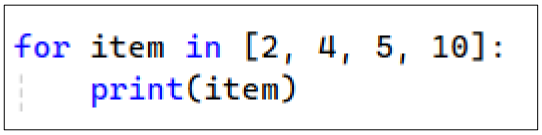
For Statement Examples



```python
for i in [0, 1, 2, 3]:
    print(i)
```

Output

```
0
1
2
3
```



```python
for item in [2, 4, 5, 10]:
    print(item)
```

Output

```
2
4
5
10
```

range( ) Function

- The **range( )** function generates a sequence of numbers, often used in loops for iteration.
- The syntax of **range( )** function:

range (**start**, **stop**, **step**)

range( ) Function

- The **range( )** function generates a sequence of numbers, often used in loops for iteration.
- The syntax of **range( )** function:

  range (**start**, **stop**, **step**)

- **start** and **step** arguments are optional, while **stop** is mandatory.

range( ) Function

range (**start**, **stop**, **step**)

- **start** → The starting number of the sequence.

  The **default value of start is 0** if not specified.

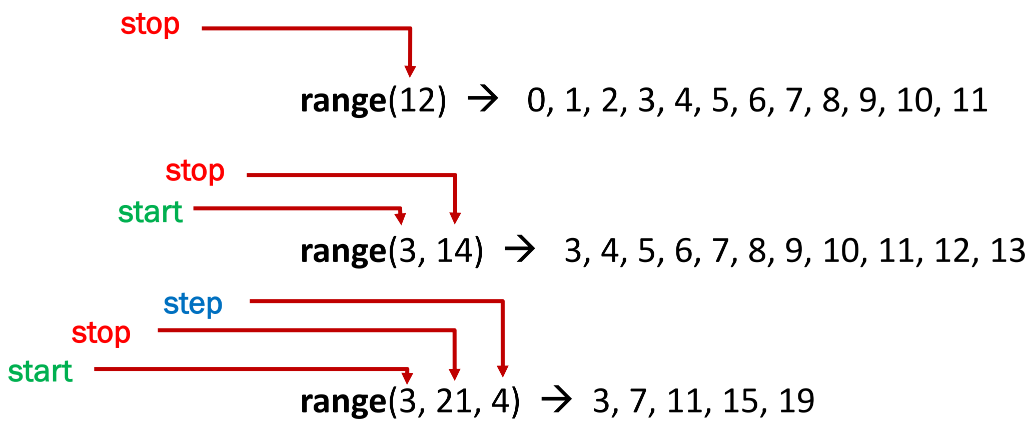
- **stop** → The sequence of numbers is generated up to this number.

  The **stop** number is **not included** in the result sequence.

- **step** → The increment value.

  The **default value of step is 1** if not specified.

range( ) Function Examples



range(12) → 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11

range(3, 14) → 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

range(3, 21, 4) → 3, 7, 11, 15, 19

What is the output of this code?

```python
for i in range(4):
    print("Review of Lecture Notes")

print ("Done!")
```

Output →

```
Review of Lecture Notes

Review of Lecture Notes

Review of Lecture Notes

Review of Lecture Notes

Done!
```

What is the output of this code?

```
for i in range(1,5):
    print("Review", i , "of Lecture Notes")

print ("Done!")
```

Output

```
Review 1 of Lecture Notes

Review 2 of Lecture Notes

Review 3 of Lecture Notes

Review 4 of Lecture Notes

Done!
```

What is the output of this code?
How many times the loop is executed?

```
for num in range(6):
    print(num*num)
```

Output

```
0

1

4

9

16

25
```

What is the code to get the following output?

```
$
$$
$$$
$$$$
$$$$$
$$$$$$
Lines of Dollars!
```

Code

```python
for i in range(1,7):
    print("$"*i)

print("Lines of Dollars!")
```

For loop with Strings

- You can use **for** loop to iterate for every character of a string.
- The following **for** loop is for printing all characters in “IT Department”, each character in a separate line.

```python
for char in "IT Department":
    print(char)
```

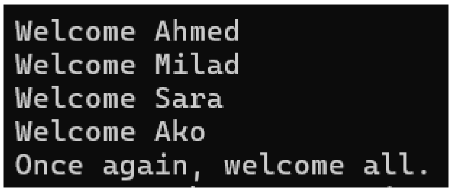
Output

```
I
T

D
e
p
a
r
t
m
e
n
t
```

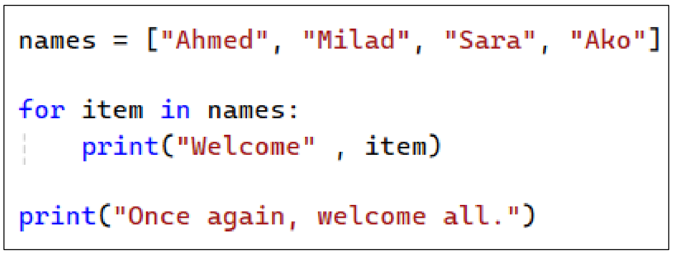
For loop with a Sequence of Strings

- We have a list of four names of students. Write a code to get the following output.

names = [“Ahmed” , “Milad” , “Sara” , “Ako”]



```
Welcome Ahmed
Welcome Milad
Welcome Sara
Welcome Ako
Once again, welcome all.
```



```python
names = ["Ahmed", "Milad", "Sara", "Ako"]

for item in names:
    print("Welcome" , item)

print("Once again, welcome all.")
```

Loop Break

- The **break** keyword in a loop <u>exits the loop immediately</u>.

- Usually, the **break** is put inside an **if** that checks for some condition.

```
nums = [12 , 1 , 6 , 13 , 6 , 0]
for num in nums :
        if (num == 6) :
                break
        print (num)
print ('All done')
```

Output

```
12
1
All done
```

Loop Continue

- The **continue** keyword directs the loop run to go back to the top of the loop immediately to start the next iteration.
- It skips the current iteration.

```
nums = [12 , 1 , 6 , 13 , 6 , 0]
for num in nums :
    if (num == 6) :
        continue
    print (num)
print ('All done')
```

Output

```
12
1
13
0
All done
```

What is the Output?

```
My_text = "I like Programming"
for  char  in  My_text:
     if  (char == "o") or (char == "a") :
          continue
     print (char)
print ("Edited Text")
```

Output

```
I

l
i
k
e

P
r
g
r
m
m
i
n
g
Edited Text
```

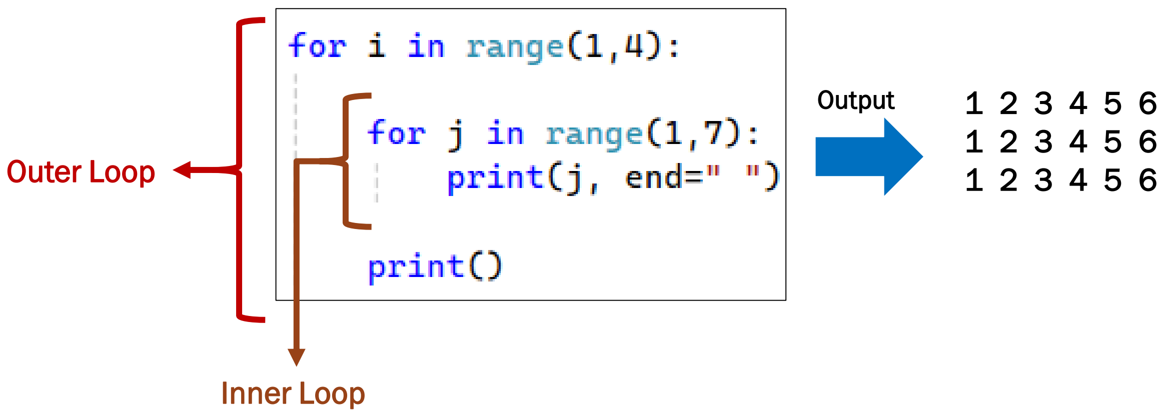
Nested Loop

- A **nested loop** is a loop inside another loop, where the outer loop determines the total number of times the inner loop will execute.

```
Outer_loop expression:
        inner_loop expression:
                Statements inside inner_loop
        statements inside outer_loop
```

Nested Loop – Example

- Consider following nested **for** loop:



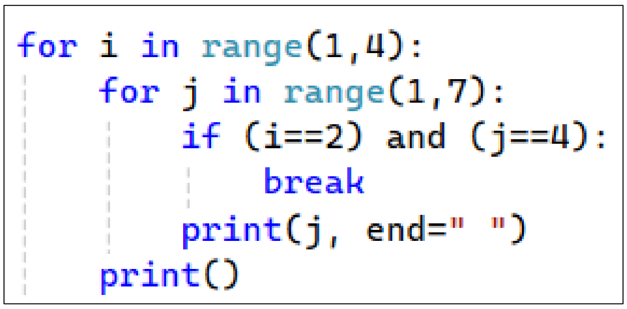
```
for i in range(1,4):

    for j in range(1,7):
        print(j, end=" ")

    print()
```

Output

```
1 2 3 4 5 6
1 2 3 4 5 6
1 2 3 4 5 6
```

Nested Loop – Example

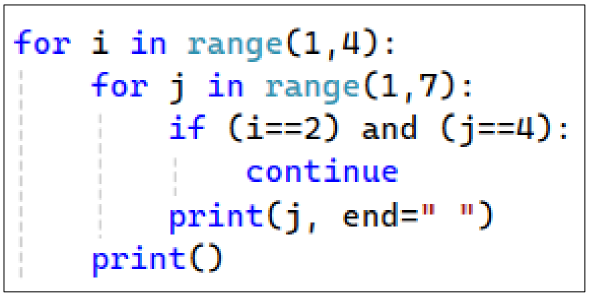
◦ What is the output of the following code?



```python
for i in range(1,4):
    for j in range(1,7):
        if (i==2) and (j==4):
            break
        print(j, end=" ")
    print()
```

Nested Loop – Example

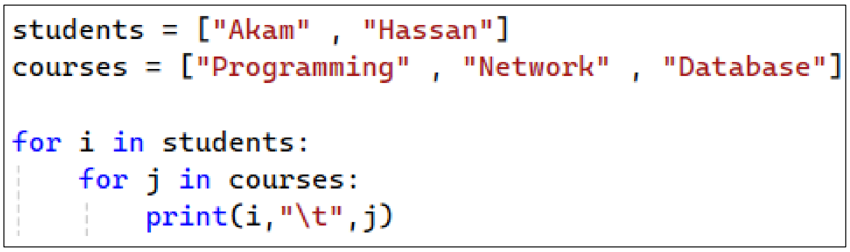
◦ What is the output of the following code?



```python
for i in range(1,4):
    for j in range(1,7):
        if (i==2) and (j==4):
            continue
        print(j, end=" ")
    print()
```

Nested Loop – Example

◦ Consider following nested **for** loop. What is the output?



```python
students = ["Akam" , "Hassan"]
courses = ["Programming" , "Network" , "Database"]

for i in students:
    for j in courses:
        print(i,"\t",j)
```

Output

```
Akam       Programming
Akam       Network
Akam       Database
Hassan        Programming
Hassan        Network
Hassan        Database
```