Programming Fundamentals

Lecture 5 – Python Loops 2

Iresh Bandara

ROBERT GORDON UNIVERSITY ABERDEEN

TEF Gold

INFORMATICS INSTITUTE OF TECHNOLOGY

Learning Outcomes

- This lecture addresses LO1,LO2 and LO4 for the module
- On completion of this lecture, students are expected to explain and apply
  - FOR loop
  - variations of range()
  - While vs FOR
  - Break and Continue
  - FOR and ELSE with Break and Continue
- Analyse program flows based on FOR loops + Break + Continue

For Loop - Introduction

- Functional wise FOR loop is similar to the WHILE
- A **for loop** repeats part of a program for a stated number of times.
- A **for loop** repeats its loop body for each element when it was given a container (list, array)
- When it reaches the end of the sequence loop terminates
- The structure of the python for loop is different when compare with other languages

For Loop - Structure

```
for [iterating variable] in [sequence]:
    #body
```

- Iterating variable store the value of the current iteration
- Sequence can be either range() or a container like list, array.
- Developer do not need to manual set a counter and increase it like WHILE loops
- No specific condition needs to be set like a WHILE loop
- Indentation needs to be followed.

For Loop – range()

- range() is a predefined function
- range() iterates through a number sequence.

```
for i in range(100):
    print(i) # print 0 to 99
```

- Always start from the 0 when the starting point was not given and last number was discarded.
- Simply the range() iterates the given number of times.

For Loop – range() contd

- range() with a starting point

```
for i in range(50,100):
    print(i) # print 50 to 99
```

- range() with starting point + increment

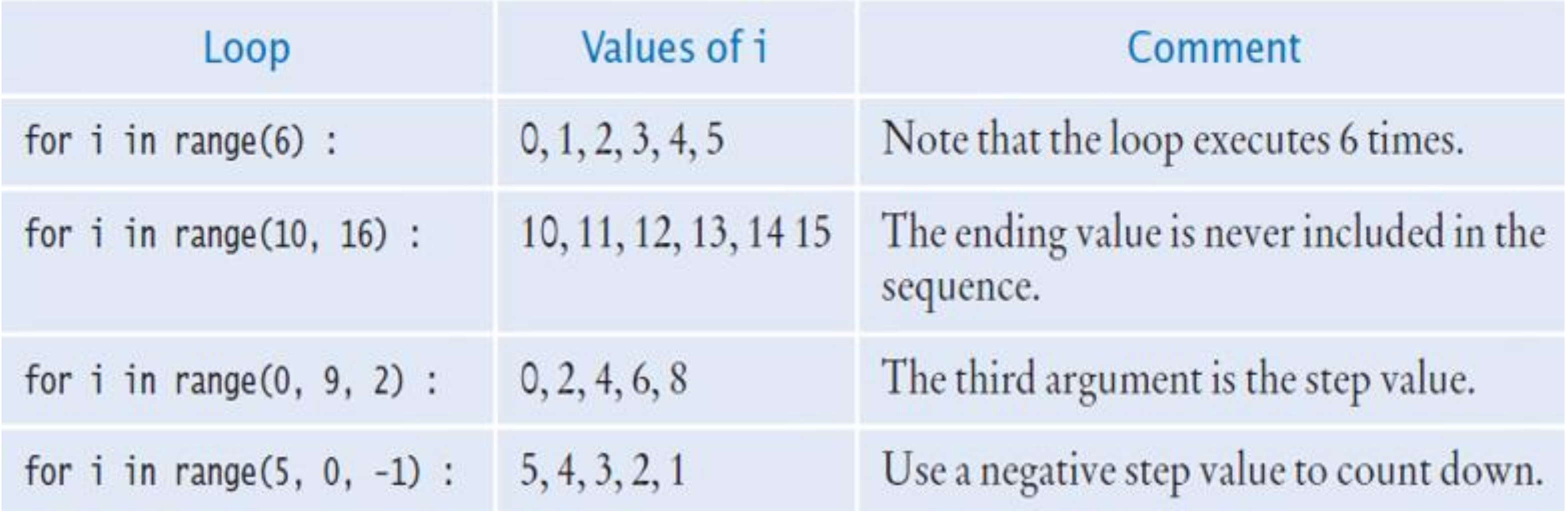
```
for i in range (50,100,5):
    print(i) #print 50,55,60…..95
```

For Loop – range() contd

- range() also support backward looping / decrement

```
for i in range (100, 50, -5):
    print(i) #print 100,95,90…55
```

- Table summarizes all the variations of range()



Loop	Values of i	Comment
`for i in range(6) :`	0, 1, 2, 3, 4, 5	Note that the loop executes 6 times.
`for i in range(10, 16) :`	10, 11, 12, 13, 14 15	The ending value is never included in the sequence.
`for i in range(0, 9, 2) :`	0, 2, 4, 6, 8	The third argument is the step value.
`for i in range(5, 0, -1) :`	5, 4, 3, 2, 1	Use a negative step value to count down.

WHILE vs FOR

- Which one is the best?

```
for i in range(2,100):
    print(i) # print 2 to 99

 counter = 2
 while counter < 100:
    print(i) #print 50,55,60…..95
    counter += 1
```

Break and Continue

- **Break** and **Continue** inside a FOR loop work the same way as WHILE
- What happens when for / while has a **ELSE**

```
start =2
end = 50
for i in range(start,end):
      if i==25:
            continue
       print(i)
else:
      print("EXIT")

#prints 2,3…..49, EXIT
#excepts 25
```

```
start =2
end = 50
for i in range(start,end):
      if i==25:
            break
       print(i)
else:
      print("EXIT")

#print 2,3,4….24
#no EXIT
```

Summary

- FOR loop is very similar to WHILE except no manual increment of the counter and the complex loop condition
- range(end) , range(start, end) ,range(start,end, increment)
- Break and continue behavior is similar to the WHILE
- FOR – ELSE with Break and FOR – ELSE with continue different results
- Above point is valid for WHILE loops too.