4 - Looping

Data Science Practicum 2021/22, Lesson 4

Marko Tkalčič

Univerza na Primorskem

Table of Contents

Collections summary

Collection	Creation	Duplicates	Ordered	Mutable	Scenario
List	[]	Yes	Yes	Yes (add, remove elements)	when you want to store similar elements `groceries=['bread','butter','cheese']`
Tuple	()	Yes	Yes	No (the tuple items can not be deleted by using the del keyword, you can delete the whole tuple)	Use a tuple when you know what information goes in the container that it is. For example, when you want to store a person's credentials for your website. `person=('ABC','admin','12345')`
Set	{}	No	No	Yes	Is the element X in the collection?
Dictionary	{:}	Keys no, values yes	No (Yes since Python 3.7)	key immutable, value mutable	address-book search by key

Table of Contents

if

- Logical conditions in Python:
  - Equals: a == b
  - Not Equals: a != b
  - Less than: a < b
  - Less than or equal to: a <= b
  - Greater than: a > b
  - Greater than or equal to: a >= b

if

- Logical conditions in Python:
  - Equals: a == b
  - Not Equals: a != b
  - Less than: a < b
  - Less than or equal to: a <= b
  - Greater than: a > b
  - Greater than or equal to: a >= b

```
a = 33
b = 200
if b > a:
  print("b is greater than a")
```

if

- Logical conditions in Python:
  - Equals: a == b
  - Not Equals: a != b
  - Less than: a < b
  - Less than or equal to: a <= b
  - Greater than: a > b
  - Greater than or equal to: a >= b

```
a = 33
b = 200
if b > a:
  print("b is greater than a")
```

- short hand if

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

elif

```
a = 33
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
```

else

```
a = 200
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
else:
  print("a is greater than b")
```

- short hand else

```
print("A") if a > b else print("B")
```

or, and

```
a = 200
b = 33
c = 500
if a > b and c > a:
  print("Both conditions are True")
```

or, and

```
a = 200
b = 33
c = 500
if a > b and c > a:
  print("Both conditions are True")
```

```
a = 200
b = 33
c = 500
if a > b or a > c:
  print("At least one of the conditions is True")
```

nested if

```
x = 41

if x > 10:
  print("Above ten,")
  if x > 20:
    print("and also above 20!")
  else:
    print("but not above 20.")
```

Table of Contents

Looping

- Looping
  - for: go through all the elements of a collection
  - while: repeat while a condition is true (does not necessarily go through each and all the elements)
  - nested loops

For

```
for iterating_var in sequence:
   statements(s)
```

For

```
for iterating_var in sequence:
   statements(s)
```

```
for letter in 'Python':     # First Example
   print 'Current Letter :', letter

fruits = ['banana', 'apple',  'mango']
for fruit in fruits:        # Second Example
   print 'Current fruit :', fruit

print "Good bye!"
```

For

```
for iterating_var in sequence:
   statements(s)
```

```
for letter in 'Python':     # First Example
   print 'Current Letter :', letter

fruits = ['banana', 'apple',  'mango']
for fruit in fruits:        # Second Example
   print 'Current fruit :', fruit

print "Good bye!"
```

```
Current Letter : P
Current Letter : y
Current Letter : t
Current Letter : h
Current Letter : o
Current Letter : n
Current fruit : banana
Current fruit : apple
Current fruit : mango
Good bye!
```

range()

- for loop in other languages?

```
for(x=0;x<10;x++){
    y = y+x;
}
```

```
for x=1:3:15,
    echo(x)
```

range()

- for loop in other languages?

```
for(x=0;x<10;x++){
    y = y+x;
}
```

```
for x=1:3:15,
    echo(x)
```

- range() function: returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.

```
for x in range(6):
  print(x)
```

range()

- for loop in other languages?

```
for(x=0;x<10;x++){
    y = y+x;
}
```

```
for x=1:3:15,
    echo(x)
```

- range() function: returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.

```
for x in range(6):
  print(x)
```

```
0
1
2
3
4
5
```

range()

- for loop in other languages?

```
for(x=0;x<10;x++){
    y = y+x;
}
```

```
for x=1:3:15,
    echo(x)
```

- range() function: returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.

```
for x in range(6):
  print(x)
```

```
0
1
2
3
4
5
```

```
for x in range(2, 6):
  print(x)
```

range()

- for loop in other languages?

```
for(x=0;x<10;x++){
    y = y+x;
}
```

```
for x=1:3:15,
    echo(x)
```

- range() function: returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and ends at a specified number.

```
for x in range(6):
  print(x)
```

```
0
1
2
3
4
5
```

```
for x in range(2, 6):
  print(x)
```

```
2
3
4
5
```

range()

```
for x in range(2, 30, 3):
  print(x)
```

range()

```
for x in range(2, 30, 3):
  print(x)
```

```
2
5
8
11
14
17
20
23
26
29
```

For using index

```
fruits = ['banana', 'apple',  'mango']
for index in range(len(fruits)):
   print 'Current fruit :', fruits[index]

print "Good bye!"
```

For using index

```
fruits = ['banana', 'apple',  'mango']
for index in range(len(fruits)):
   print 'Current fruit :', fruits[index]

print "Good bye!"
```

```
Current fruit : banana
Current fruit : apple
Current fruit : mango
Good bye!
```

for else

- If the else statement is used with a for loop, the else statement is executed when the loop has exhausted iterating the list

```
for x in range(6):
  print(x)
else:
  print("Finally finished!")
```

```
0
1
2
3
4
5
Finally finished!
```

While

```
while expression:
   statement(s)
```

While

```
while expression:
   statement(s)
```

```
count = 0
while (count < 9):
   print 'The count is:', count
   count = count + 1

print "Good bye!"
```

```
The count is: 0
The count is: 1
The count is: 2
The count is: 3
The count is: 4
The count is: 5
The count is: 6
The count is: 7
The count is: 8
Good bye!
```

break, continue

- With the `break` statement we can stop the loop even if the while condition is true
- With the `continue` statement we can stop the current iteration, and continue with the next

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

```
1
2
3
```

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

```
1
2
4
5
6
```

while else

```
i = 1
while i < 6:
  print(i)
  i += 1
else:
  print("i is no longer less than 6")
```

```
1
2
3
4
5
i is no longer less than 6
```

Nested loops

- loop in a loop (. . . in a loop. . .)

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

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

Nested loops

- loop in a loop (. . . in a loop. . .)

```
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
```

Nested loops

- loop in a loop (. . . in a loop. . .)

```
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
```

- Cyclomatic complexity: a software metric used to indicate the complexity of a program:
  - nested loops are more complex than simple
- Computational complexity: $O(n^2)$

Table of Contents

Exercise for

- create a list with 15 names
- sort the list
- print out every third name starting with the fourth using the for loop

Exercise for

- create a list with 15 names
- sort the list
- print out every third name starting with the fourth using the for loop

```
names = ["John", "Jim", "Anna", "George", "Paul", "Richard", "Robert", "Ella", "Billie", \
         "Amy", "Dorothy", "Virginia", "Jonatan", "Markus", "Peter"]
print(len(names))
names.sort()
print(names)
for i in range(3,15,3):
    print(names[i])
```

```
15
['Amy', 'Anna', 'Billie', 'Dorothy', 'Ella', 'George', 'Jim', 'John', 'Jonatan', 'Markus', 'Paul', 'Peter', 'Richard', 'Robert', '
Dorothy
Jim
Markus
Richard
```

Exercise while

- make a list with 15 names
- sort them
- print only those whose first letter is J using the while loop

Exercise while

- make a list with 15 names
- sort them
- print only those whose first letter is J using the while loop

```
names = ["John", "Jim", "Anna", "George", "Paul", "Richard", "Robert", "Ella", "Billie", \
         "Amy", "Dorothy", "Virginia", "Jonatan", "Markus", "Peter"]
print(len(names))
names.sort()
i=0
while i<len(names):
    if names[i][0] == "J":
        i+=1
        continue
    print(names[i])
    i+=1
```

```
15
Amy
Anna
Billie
Dorothy
Ella
George
Markus
Paul
Peter
Richard
Robert
Virginia
```

Exercise nested

- Print the following output

```
1
3 3 3
5 5 5 5 5
7 7 7 7 7 7 7
9 9 9 9 9 9 9 9 9
```

Exercise nested

- Print the following output

```
1
3 3 3
5 5 5 5 5
7 7 7 7 7 7 7
9 9 9 9 9 9 9 9 9
```

```
for x in range(1,10,2):
    for y in range(x):
        print(x, end = " ")
    print()
```

Exercise

- Accept a number from the user and calculate the sum of all numbers between 1 and the user given number

```
sum1 = 0
n = int(input("Please enter number "))
for i in range(1, n + 1, 1):
    sum1 += i
print("\n")
print("Sum is: ", sum1)
```

Exercise

- Given the list below iterate it and display all the numbers that are divisible by 5 and if you find number greater than 150 stop the loop

```
list1 = [12, 15, 32, 42, 55, 75, 122, 132, 150, 180, 200]
```

Exercise

- Given the list below iterate it and display all the numbers that are divisible by 5 and if you find number greater than 150 stop the loop

```
list1 = [12, 15, 32, 42, 55, 75, 122, 132, 150, 180, 200]
```

```
list1 = [12, 15, 32, 42, 55, 75, 122, 132, 150, 180, 200]
for item in list1:
    if (item > 150):
        break
    if(item % 5 == 0):
        print(item)
```

Exercise

- Given a number count the total number of digits in a number

Exercise

- Given a number count the total number of digits in a number

```
num = 75869
count = 0
while num != 0:
    num //= 10
    count+= 1
print("Total digits are: ", count)
```

Exercise

- Reverse print the following list using the for loop

```
list1 = [10, 20, 30, 40, 50]
```

Exercise

- Reverse print the following list using the for loop

```
list1 = [10, 20, 30, 40, 50]
```

```
list1 = [10, 20, 30, 40, 50]
start = len(list1) - 1
stop = -1
step = -1
for i in range(start, stop, step) :
    print(list1[i])
```

Table of Contents

Assignment

Use loops, if/else etc. to write a program that prints the letter Z as follows

```
*******
      *
     *
    *
   *
  *
 *
*******
```

References

Part of the material has been taken from the following sources. The usage of the referenced copyrighted work is in line with fair use since it is for nonprofit educational purposes.

- https://www.tutorialspoint.com/python/python_for_loop.htm
- https://www.w3schools.com/python/python_for_loops.asp