

LEARNING TO CODE



pythonTM

Coordinator: Konstantinos Emmanouilidis

IEEE SB OF THRACE

2017-2018

- LISTS
- TUPLES
- LOOPS



python



LISTS

- Lists are Python's generalized arrays
- Sequence of not necessary homogenous objects/variables
- Index starts from 0
- You can add, remove or search for items in the list

LISTS

➤ **DECLARATION:**

```
mylist=[]
```

➤ **Add elements:**

```
mylist=[1,2,3]
```

Or

```
mylist=[]
```

```
mylist.append(1)
```

```
mylist.append(2)
```

```
mylist.append(3)
```

➤ **Accessing an index:**

➤

```
mylist[2]
```

 accesses to the 3rd element of the list

OPERATIONS ON LISTS

| | |
|-------------------|--|
| x in A | Check whether an item in the list. Returns True or False |
| x not in A | The same as not(x in A) |
| min(A) | The smallest element of list |
| max(A) | The largest element in the list |
| A.index(x) | The index of the first occurrence of element x in the list; in its absence generates an exception ValueError |
| A.count(x) | The number of occurrences of element x in the list |

TUPLES

- Tuples are used to hold together multiple objects.
- One major feature of tuples is that they are *immutable* like strings i.e. you cannot modify tuples.
- Tuples are defined by specifying items separated by commas within an optional pair of parentheses.

EXAMPLE

```
zoo = ('python', 'elephant', 'penguin')
print('Number of animals in the zoo is', len(zoo))

new_zoo = 'monkey', 'camel', zoo    # parentheses not required but are a good idea
print('Number of cages in the new zoo is', len(new_zoo))
print('All animals in new zoo are', new_zoo)
print('Animals brought from old zoo are', new_zoo[2])
print('Last animal brought from old zoo is', new_zoo[2][2])
print('Number of animals in the new zoo is',
      len(new_zoo)-1+len(new_zoo[2]))
```

Result:

```
Number of animals in the zoo is 3
Number of cages in the new zoo is 3
All animals in new zoo are ('monkey', 'camel', ('python', 'elephant', 'penguin'))
Animals brought from old zoo are ('python', 'elephant', 'penguin')
Last animal brought from old zoo is penguin
Number of animals in the new zoo is 5
```

LOOPS

THE **loop**

THE “FOR” LOOP

- For loops iterate over a given sequence.
- For a sequence of numbers use range() function.
- range() function returns a list numbers of that specified range.
- General form of range() function: range(start,end,step).
- It is zero based and step by default is 1.

THE “FOR” LOOP

➤ FORM:

for var ***in*** sequence:

statement 1

statement 2

statement 3

.

.

.

➤ FORM FOR SEQUENCE OF NUMBERS:

for var ***in*** range(start,end,step):

statement 1

statement 2

statement 3

.

.

.

EXAMPLES

script.py

```
1 # Prints out the numbers 0,1,2,3,4
2 ▾ for x in range(5):
3     print(x)
4
5 # Prints out 3,4,5
6 ▾ for x in range(3, 6):
7     print(x)
8
9 # Prints out 3,5,7
10 ▾ for x in range(3, 8, 2):
11     print(x)
```

Result:

IPython Shell

```
0
1
2
3
4
3
4
5
3
5
7
```

script.py

```
1 primes = [2, 3, 5, 7]
2 ▾ for prime in primes:
3     print(prime)
```

Result:

```
2
3
5
7
```

THE “WHILE” LOOP

➤ FORM:

while condition:
statement 1
statement 2
statement 3

•
•
•

script.py

```
1  # Prints out 0,1,2,3,4
2
3  count = 0
4  while count < 5:
5      print(count)
6      count += 1  # This is the same as count = count + 1
```

Result:

IPython Shell

```
0
1
2
3
4
```


"BREAK" AND "CONTINUE" STATEMENTS

- **break** is used to exit a loop.
- **continue** skips the current block, and returns to the "for" or "while" statement.

script.py

```
1  # Prints out 0,1,2,3,4
2
3  count = 0
4  while True:
5      print(count)
6      count += 1
7      if count >= 5:
8          break
```

```
# Prints out only odd numbers - 1,3,5,7,9
for x in range(10):
    # Check if x is even
    if x % 2 == 0:
        continue
    print(x)
```

IPython Shell

```
0
1
2
3
4
```

```
1
3
5
7
9
```

"ELSE" CLAUSE FOR LOOPS

When the loop condition is false, the code in the “else” block is executed.

script.py

```
1  # Prints out 0,1,2,3,4 and then it prints "count value
   reached 5"
2
3  count=0
4  while(count<5):
5      print(count)
6      count +=1
7  else:
8      print("count value reached %d" %(count))
9  -
```

IPython Shell

```
0
1
2
3
4
```

EXAMPLES

```
# Prints out 1,2,3,4
for i in range(1, 10):
    if(i%5==0):
        break
    print(i)
else:
    print("this is not printed because for loop is
    terminated because of break but not due to fail in
    condition")
```

Result:

```
1
2
3
4
```

DO-WHILE LOOP IMPLEMENTATIONS

➤ First Implementation

statements

while condition:

statements

➤ Second Implementation

while True:

statements

if fail_condition:

break