

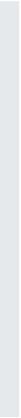


Lets learn Python

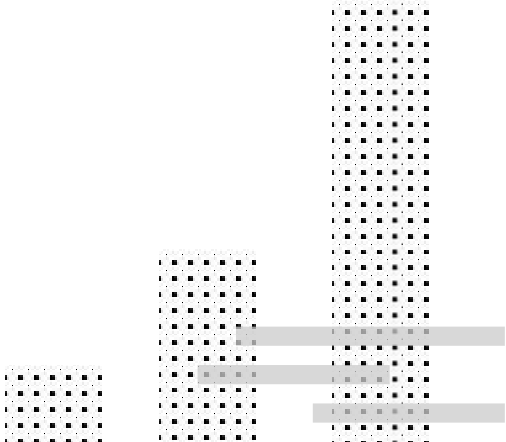
ekakshya.com
09 December 2021



Lesson 2



- Comments
- Lists
- Strings
- Tuples
- While loop
- For loop
- Slices



Comments

- Comments enable you to write meaningful messages for other developers
- Comments are not visible while program execution
- 3 types of comment are available in python

Single line comment using hash symbol

```
"""
```

Multi line comment using double quotes “

```
"""
```

```
”
```

Another multi line comment using single quote ‘

```
’
```

Lists

- Lists is a type of collection data type. It allows you to store more than one type of data in a contiguous memory location
- Easy retrieval and updation of data in list
- One of the most frequently used storage patterns in Python
- Allowed operations on a list:
 - 1) `append()`
 - 2) `len()`
 - 3) `pop()`
 - 4) `count()`
 - 5) `index()`
 - 6) `remove()`
 - 7) `in`
 - 8) `extend()`

Lists

```
num_list = [1,2,3,"4",None,[6,58,19]]
num_list.append(100)
print(num_list) #[1,2,3,"4",None,[6,58,19],100]
print(len(num_list)) #7
print(num_list.pop()) #100
print(num_list.count(3)) #1
print(num_list.index(3)) #2
num_list.remove(1)
print(num_list)#[2,3,"4",None,[6,58,19]]
print(2 in num_list) #True
print(19 in num_list) #False
num_list.extend([5,9,8,0]) #[2,3,"4",None,[6,58,19],5,9,8,0]
print(num_list + [7,22,0]) #[2,3,"4",None,[6,58,19],5,9,8,0,7,22,0]
print(num_list) #[2,3,"4",None,[6,58,19],5,9,8,0]
```

Strings

- String is used to store any data surrounded by single quote 'Hello@' or double quote "World5"
- String is a type of collection data type. As, such methods applicable to collection can be used with String. For eg, len, in etc
- "hello" in "hello world" #true

Strings

```
greeting = "heLLo woRld"
another_greeting = 'This world is beautiful'
length_greeting = len(greeting)
print(length_greeting) #11
print(greeting.count("o")) #2
print(greeting.find("e")) #1
print(greeting.find("g")) #-1
print(greeting.upper()) #HELLO WORLD
print(greeting.lower()) #hello world
print(greeting.capitalize()) #HeLLo WoRld
num_str = "15"
print(num_str.isdigit()) #True
num_str = "15.7"
print(num_str.isdigit(15.7)) #False
print(greeting.isdigit()) #False
print(greeting.split(" ")) #["heLLo", "woRld"]
print(greeting.replace(" ", "@")) #"heLLo@woRld"
print("Hello" * 3) # HelloHelloHello
print("Todays\'s Menu") #Today's Menu
multiline_string = """ Hello, This is a
Multiline string """
print("|".join(["c", "a", "t"])) #c|a|t
print("*".join(["c", "a", "t"])) #c*a*t
print(f'{greeting} and {another_greeting} also {greeting}') #f-strings to easily manipulate data or expressions in strings
```

Tuples

- Tuple is used to store any data surrounded by paranthesis ()
- Tuple is a type of immutable collection data type. As, such methods applicable to collection can be used with Tuple. For eg, len, in etc
- "hello" in ("hello",world) #true

Tuples

```
first_tuple = (1,2,"Hello",100,[101,105])
#first_tuple[1] = 3 #error, not allowed
new_tuple = first_tuple[1],97,first_tuple[4]
print(new_tuple) #(2, 97, [101, 105])
second_tuple = (first_tuple[1],97,first_tuple[4])
print(second_tuple) #(2, 97, [101, 105])
print(first_tuple[4][0]) #101
print(len(first_tuple)) #5
print(first_tuple.count(100)) #1
print(100 in first_tuple) #True
print(105 in first_tuple) #False
print(first_tuple.index(2)) #1
print(first_tuple * 5) #repeats the tuple 5 times
```

While loop

```
while True:
```

```
    a_number = input("Enter a number: ")
```

```
    if a_number.isdigit():
```

```
        print(a_number)
```

```
        break
```

```
i = 3
```

```
while i < 10:
```

```
    if i == 3:
```

```
        print("Found 3")
```

```
        break #required for while else or else part gets executed by default
```

```
    i = i + 1
```

```
else:
```

```
    print("3 not found")
```

```
    """ else part gets executed after while loop execution is complete  
        so we add break to get out of loop """
```

PS: While loop is used when the number of times a loop has to be executed is unknown

For loop: list

```
for i in range(10):
```

```
    print(i)
```

```
for i in range(5,23,2):
```

```
    print(i)
```

```
for i in range(0,-20,-2):
```

```
    print(i)
```

```
lst = [1,2,49,27,True,False]
```

```
for idx in range(len(lst)):
```

```
    print(lst[idx])
```

```
for elem in lst:
```

```
    print(elem)
```

```
for idx,elem in enumerate(lst):
```

```
    print(idx,elem)
```

For loop: tuple

```
a_tuple = (1,2,49,27,True,False)
for idx in range(len(a_tuple)):
    print(a_tuple[idx])
for elem in a_tuple:
    print(elem)
for idx,elem in enumerate(a_tuple):
    print(idx,elem)
```

For loop: string

```
a_string = "Hello World"
for idx in range(len(a_string)):
    print(a_string[idx])
for elem in a_string:
    print(elem)
for idx,elem in enumerate(a_string):
    print(idx,elem)
```

For loop

continue, break, pass

```
a_string = "Hello World"
for idx in range(len(a_string)):
    if a_string[idx] == "e":
        continue
    elif a_string[idx] == "W":
        break
    else:
        print(a_string[idx])

for i in range(10):
    pass
```

For loop

nested loops

```
a_list = [[1,1],[2,9],[7,2],[4,5]]  
for idx in range(len(a_list)):  
    internal_list = a_list[idx]  
    for elem in internal_list:  
        print(elem)
```

```
a_string = "hello world"  
for i,char in enumerate(a_string):  
    if char=='w':  
        break  
  
print(i)
```

For else loop

```
keywords = ("a","python","tutorial","for", "everyone")  
for word in keywords:  
    if word == "tutorial":  
        print("Found tutorial")  
        break #required for for-else loop  
else:  
    print("tutorial not found") #executes after for loop
```


Slices

```
a_list = [4,27,19,34,35,100]
new_list = a_list[0:5:2] # [4,19,35]
print(new_list)
```

Slices return a subset of collection type data which includes list, string and tuple etc

Slices follow the format:

1. Starting_point (included in the result)
2. Ending_point (excluded from the result)
3. Steps (number of steps taken)

PS: All the above mentioned points can be negative. Slice always returns a brand new object; it doesn't modify the existing collection

The End

Thank You

