

PYTHON

From Simple to Complex With Examples

AYESHA NOREEN

Bachelor's in Software Engineering,

Master's in Computer Science

from COMSATS University, Islamabad

NOTE!!!

In these notes Screenshots of practice examples and coding are added. The code files are also available in code folder that contain .ipynb files that are created on Jupyter notebook.

Chapter8

Lists in python

List is a data structure contains sequence of data within square brackets. Lists are mutable, ordered and allow duplication.

```
names=["Ayesha","Rimsha","Sana"]
print(names)      #prints whole names list
print(names[0])   #prints names at index 0 which is Ayesha
mixed=[1,2,3,"four",5,'six',7.0,"none"] #mean we can store any type of data in list
print(mixed[2])   #prints 3
print(mixed[3])   #prints four
print(mixed[6])   #prints 7.0
print(mixed[-1])  #print none mean nothing not zero
mixed[3]="4"      #it replaces four with 4
print(mixed)
mixed[2]="two"
print(mixed)
```

- **Add data in a List**

We can add or concat list by following ways.

- **1-append()method**

Append method takes only one parameter and insert it at the last of list.

- **2-insert()method**

Insert method takes two parameters in 1st parameter we give index number(location) at which we want to insert data and at 2nd parameter we give data which we have to insert.

- **3-extend()method**

Extend method is used for concatenation of two strings

- Add data in list

```
names=[] #list can also be empty we can fill items in it by using append method
print(names)
names.append("ayesha") #append method always takes only 1 argument and store it at last of list
print(names)
names.append("noreen")
names.append("rimsha")
print(names)
names.insert(0,"sana") #if we want to insert data in list at specific place than use insert method
print(names)
names2=["arfan","adnan","ali"]
names+=names2 #we can add two strings as save result in names list
print(names) #it prints all data of names and names2
names3=names+names2 #or concatenation of lists can also be done through this method concatenation
print(names3)
names.extend(names2)
print(names) #prints [names,names2] makes names,names2 a whole list
names.append(names2)
print(names) #prints [names,[names]] mean give list inside list.
```

- **Delete data in list**

We can delete data of list through following ways

- **1-pop()method**

Remove last element of list if we pass specific index number than it pops out from that index but we cannot pass value in it. It takes only index in its parameter.

- **2-del keyword**

Del from specific location

- **3-remove()method**

Take value in its parameter not index.

- delete data in list

```
names=['ayesha','noreen','sana','rimsha','noreen','arfan','adnan','ali']
names.pop()           #it pops out last item of list
print(names)          #prints ['ayesha','noreen','sana','rimsha','noreen','arfan','adnan']
#names.pop('sana')    #if we want to delete some specific name it give error
names.pop(2)          #if we want to delete from some specific locaion
print(names)          #it prints ['ayesha','noreen','rimsha','noreen','arfan','adnan']
#del names[]
#print(names)         #error del method parameter is not empty it should have some name
del names[2]
print(names)
#del names['arfan']
#print(names)         #gives error bcz cannot take string
#names.remove()
#names.print()        #give error bcz removes take exactly one parameter
names.remove("noreen")
print(names)          #in string two noreen are present it removes 1st noreen
#names.remove(2)      #it gives error bcz it does not take number as parameter
#print(names)

['ayesha', 'noreen', 'sana', 'rimsha', 'noreen', 'arfan', 'adnan']
['ayesha', 'noreen', 'rimsha', 'noreen', 'arfan', 'adnan']
['ayesha', 'noreen', 'noreen', 'arfan', 'adnan']
['ayesha', 'noreen', 'arfan', 'adnan']
```

- **In keyword in list**

In keyword is used to find specific element in list.

```
▼ #in keyword in list
names=["name","ayesha","noreen"]
▼ if "name" in names:
    print("name is present in names list")
▼ else:
    print("name is not present in names list")
#it prints name is present in names list

name is present in names list
```


- Some more functions of list

```
#Some more functions of list
name=["name","ayesha","noreen","sana","noreen"]
print(name.count("noreen"))      #noreen is 2 times in list so it print 2
number=[3,4,5,6,1,3,5,]
number.sort()                    #it sort original number list in ascending order
print(number)
name.sort()                      #it sort names list alphabatically
print(name)
numbers=[5,2,4,1,3]
print(sorted(numbers))           #it prints sorted numbers list
print(numbers)                  #it prints original list without sorting
numbers.clear()                 #remove all items of numbers
print(numbers)                  #it prints numbers as []
copy_name=name.copy()           #name list is copied in copy_name
print(copy_name)                #it prints name list which is copied in copy_name
```

- Is VS == keyword

== is used to check lists are equal or not. it returns true if equal and false if not same.

Is keyword is used to check two list have same memory space or not

```
▼ #is vs == keyword
name1=['ayesha','noreen','sana']
name2=['ayesha','noreen','sana']
name3=['ayesha','noreen','sana','rehman']
print(name1==name2)      #prints true
print(name2==name3)      #print false
print(name1 is name2)    #print false
print(name2 is name3)    #print false
print(name3 is name1)    #print false

True
False
False
False
False
```

- ## Split and join method

- split method is used to convert string into list
- join method converts list into string

```
▼ #split and join methods
user_info="ayesha 22".split()      #split from spaces
print(user_info)                  #
employee_info="sana,13".split(",") #split from commas
print(employee_info)              #prints ['sana', '13']
name,age="ayesha,22".split(",")   #we can store name and age in two seprate variables
print(name)
print(age)
user_info=['ayesha', '22']
print(",".join(user_info))        #prints ayesha,22

['ayesha', '22']
['sana', '13']
ayesha
22
ayesha,22
```

- Looping in lists

```
▼ #For Loop and while Loop:  
names=['ayesha','noreen','sana']  
▼ for i in names:  
    print(i, end=" ") #prints ayesha,noreen,sana  
print('\n')  
i=0  
▼ while i<len(names):  
    print(names[i], end=" ")  
    i+=1
```

ayesha noreen sana

ayesha noreen sana

- ## String VS lists

String is immutable while list is mutable. we cannot add data in string through any function while we can add data in list through append, extend and insert methods. As,

```
▼ # string vs list
string="Ayesha"
string.upper()
print(string)    #prints Ayesha not AYESHA bcz strings are immutable mean not change original string
upper=string.upper()
print(upper)     #prints AYESHA bcz it is not original string
list_items=['ayesha','noreen','sana']
del list_items[2]
print(list_items) #prints ['ayesha','noreen']mean change in original string
```

Ayesha

AYESHA

['ayesha', 'noreen']

- **Array VS lists**

In array that we use in c, c++, java we can store only one data type at a time e.g string or int or float type .but in list we can store any type of data mean list contain mixed data type. In python we can use array module that we have to import for use its functionalities and having faster speed and better performance but in python we use only list because list is flexible.

- **List inside list**

If there is list inside list than also called 2d list and if there is list inside list and list inside list than called 3d array.

```
▼ #list inside list
two_dimension_list=[[1,2,3],[4,5,6],[7,8,9]] #this list contains 3 items each item contains 3 subitems
▼ for i in two_dimension_list:
    print(i) #prints[1,2,3]
           # [4,5,6]
           # [7,8,9]

# if we want to print 2nd item of 2nd list
print(two_dimension_list[1][1]) #if we want to print whole items of list including subitems
#than apply two loops for loop inside for
▼ for i in two_dimension_list:
▼     for j in i:
        print(j)
```

- **TODO Task**

A program that returns square of each element of list

```
#A program that returns sequare of each element of list  
def square(l):  
    sq=[]  
    for i in l:  
        i*=i  
        sq.append(i)  
    return sq  
numbers=[1,2,3,4,5]  
print(f"original List is:{numbers}")  
print(f"Squared list is:{square(numbers)}")
```

```
original List is:[1, 2, 3, 4, 5]  
Squared list is:[1, 4, 9, 16, 25]
```


- **TODO Task**

Take a list than print reverse of list using [::-1] and also append and pop method

```
numbers=[1,2,3,4,5,6,7,8,9]
def reverse(List):
    i=List[::-1]
    return i
print(f"Original list is:{numbers}")
print(f"Reverse list is:{reverse(numbers)}")
words=['word1','word2','word3']
def reverse(List):
    i=List[::-1]
    return i
print(f"Original list is:{words}")
print(f"Reverse list is:{reverse(words)}")
```

Original list is:[1, 2, 3, 4, 5, 6, 7, 8, 9]
Reverse list is:[9, 8, 7, 6, 5, 4, 3, 2, 1]
Original list is:['word1', 'word2', 'word3']
Reverse list is:['word3', 'word2', 'word1']

- **TODO Task**

Take a program that separates even and odd numbers of list

```
names=['ayesha','noreen','sana']
def reverse_elements(l):
    emp=[]
    for i in l:
        j=i[::-1]
        emp.append(j)
    return emp
print(f"Original list is:{names}")
print(f"Reverse of elements of string are:{reverse_elements(names)}")
numbers=['100','200','300']
def reverse_elements(l):
    emp=[]
    for i in l:
        j=i[::-1]
        emp.append(j)
    return emp
print(f"Original list is:{numbers}")
print(f"Reverse of elements of list are:{reverse_elements(numbers)}")
```

Original list is:['ayesha', 'noreen', 'sana']
Reverse of elements of string are:['ahseya', 'neeron', 'anas']
Original list is:['100', '200', '300']
Reverse of elements of list are:['001', '002', '003']

- **TODO Task**

Take a program that find out common elements of two lists

```
#take to lists than findout common from these lists.  
def common_finder(l1,l2):  
    output=[]  
    for i in l1:  
        if i in l2:  
            output.append(i)  
    return output  
print(common_finder([1,2,3,4,5],[1,2,3]))  
#it prints [1,2,3]
```

```
[1, 2, 3]
```

- **Min and max functions**

```
numbers=[1,2,3,4,5,6,7,8,9]
print(f"Original List is:{numbers}")
print(f"Maximum number from list is:{max(numbers)}")
print(f"Minimum number from list is:{min(numbers)}")
def max_diff(l):
    return max(l)-min(l)
print(f"Greatest difference of list is:{max_diff(numbers)}")
```

```
Original List is:[1, 2, 3, 4, 5, 6, 7, 8, 9]
Maximum number from list is:9
Minimum number from list is:1
Greatest difference of list is:8
```

- **TODO Task**

```
▼ #take a list containg many sublists define a function that counts number of sublists  
numbers=[1,2,3,[4,5],[6],[],[7,8,9,10]]  
▼ def sublists_counter(l):  
    count=0  
    ▼ for i in l:  
        ▼ if type(i)==type(l):  
            count=count+1  
    return count  
print(sublists_counter(numbers))
```