Day-4

String functions & operators

A Python string is a sequence of characters. There is a built-in class ‘str’ for handling Python string.

Creating strings is easy as you only need to enclose the characters either in single or double-quotes.

```
# Python string examples - all assignments are identical. var = 'Floxus'
var = "Floxus" var = """Floxus""" # with Triple quotes Strings can extend
to multiple lines String_var = """ This is a multi-line string!!! """
```

Access Individual Character of a String

```
var1= 'Floxus' print (var1[0]) # return 1st character # output: F print
(var1[-1]) # return last character # output: s print (var1[-2]) # return
last second character # output: u
```

Slice a String in Python

To retrieve a range of characters in a String, we use the ‘slicing operator,’ the colon ‘:’ sign. With the slicing operator, we define the range as [a:b]. It will print all the characters of the String starting from index ‘a’ up to char at index ‘b-1’. So the char at index ‘b’ is not a part of the output.

```
var1 = 'Floxus' print (var1[2:5]) #return a range of character # oxu prin
t (sample_str[3:]) # return all characters from index 3 # xus print (samp
le_str[:3]) # return all characters before index 3 # Flo print (sample_st
r[0:-3]) # Flo
```

Deleting a String in Python

Python Strings are by design immutable. It means that once a String binds to a variable, it can't be modified.

```
var = 'Floxus' var[2] = 'a' # TypeError: 'var' object does not support it
em assignment del var print (var) # NameError: name 'var' is not defined
```

String Operators in Python

```
var1 = 'Flo' var2 = 'xus' print (var1+var2) #Concatenation (+) #Output: F
loxus var1 = 'Floxus' print (var1*3) #Repetition #Output: FloxusFloxusFlo
xus var1 = 'Floxus' print ('o' in var1) #Membership (in) #Output: True va
r1 = 'Floxus' print ('O' not in var1) #Membership (not in) #Output: True
for i in var1: print (var, end ="") #Output: Floxus print ("Python is a
"high" used language") # SyntaxError: invalid syntax # After escaping wit
h double-quotes print ("Python is a \\"high\\" used language") # Output:
Python is a "high" used language
```

Built-in String Functions in Python

```
var = 'FLOXUS' print (var.capitalize()) #Returns the string with the firs
t character capitalized and rest of the characters in lower case. #Outpu
t: Floxus var = 'Floxus' print (var.lower()) #Converts all the characters
of the String to lowercase # Output: floxus var = 'Floxus' print (var.upp
er()) #Output: FLOXUS var = 'FloXus' print (var.swapcase()) #Output: fLOx
US var='Programmers' str='m' print (var.count(str)) #Output: 2 var='This
is a good example' print (len(var)) #Output: 22
```

Lists 👩‍💻

In this Data Structure, we can store data of different data types in a sequence and are changeable.

Syntax -

```
my_list = [1, 2, 3, 'four'] print(my_list) #Output [1, 2, 3, 'four']
```

→ How to **access** an element in List?🤔

Syntax -

```
my_list = [1, 2, 3, 'four'] print(my_list[1])#indexing starts from 0 #Out
put 2 print(my_list[0:2]#it will print values from 0th index to 2nd #Outp
ut [1,2,3] print(my_list[-1])#it will print last value of the list #Outpu
t ['four']
```

→How to **add** any element in List?🤔

```
append( ),extend( ),and insert( ) these functions are used to add any ele
ment in the list.
```

Syntax -

```
my_list = [1, 2, 3, 'four'] my_list.append(['five',6])#it will add as sin
gle element **print(my_list)#Output [1, 2, 3, 'four',['five',6]] my_list.
extend('five') print(my_list)#Output [1, 2, 3, 'four',['five',6],'five']
my_list.insert('zero') print(my_list)#Output ['zero',1, 2, 3,'four',['fiv
e',6],'five']
```

→How to delete any element in List?🤔

For deleting any element from list we can use del( ),remove( ), and clear( ) functions

Syntax -

```
my_list = [1, 2, 3, 'four'] del my_list[1];#for deletion by taking the re
ferance of index print(my_list)# Output [1,3,'four] my_list.remove(3);#di
rectly passing the value that is to be deleted print(my_list)#Output
[1,'four'] my_list.clear();#it will empty the list print(my_list)#Output
[]
```

Now you have a basic idea to work with lists😇

There are few functions that can help you a lot in coding💁‍♀️

- sort( ) :- This function can arrange your data in increasing order and it modifies the original list.
- sorted( ):- This function returns the sorted list but does not change the original list.
- count( ):- This function will count how many times any specific value is occurring in a list.
- len( ):- This function will return the number of elements in a list.

```
my_list = [1, 2,20,3,6, 10] print(sorted(my_list))#It will print the sort
ed list but no change in original list #Output [1,2,3,4,10,20] my_list.so
rt(reverse=False);#It will change the original list to sorted list print
(my_list) #Output [1,2,3,4,10,20] print(my_list.count(2)) #Output 1 print
(len(my_list)) #Output 6
```

That's all for lists

😇😇😇😇😇😇😇😇😇😇😇😇😇😇😇😇😇😇😇😇