Builtin Functions of Strings

Count() Function

In [ ]:
```
count() Function --> It is basically used to return the count of the given substring.

Syntax:
    string_name.count(Sub_String , Start_index , End_index)

Note --> Start_Index and End_Index both are optional if you are not giving start_index and end_index
         then it will take start_index as 0 and end_index as length of the string.

    -->If the given substring is not present then you will get answer as 0
```

Examples of Count Function

In [2]:
```
s="abcabcbabcabcbac"
print(s.count("a"))
```
```
5
```

In [3]:
```
s="abcabcbabcabcbac"
print(s.count("ab",2,12))
```
```
3
```

In [4]:
```
s="abcabcbabcabcbac"
print(s.count("z"))
```
```
0
```

In [5]:
```
s="abcabcbabcabcbac"
print(s.count("ba",5,16))
```
```
2
```

Replace() Function

In [ ]:
```
Replace Function --> The replace() in Python returns a copy of the string where all occurrences of a
                     substring are replaced with another substring.
Syatax :
        string.replace(old_string,new_string)

Note: If the old string is not present in the given string then nothing will happen you will get
      original string without any error
```

Example of Replace Function

In [9]:
```
s="learning java is easy"
s1=s.replace("123","difficult")
print(s1)
```
```
learning java is easy
```

In [8]:
```
s="learning java is easy and learning programming is easy"
s1=s.replace("easy","hard")
print(s1)
```
```
learning java is hard and learning programming is hard
```

In [10]:
```
s="learning java is easy"
s1=s.replace("easy","difficult")
print(s1)
```
```
learning java is difficult
```

Split() Function

In [ ]:
```
Split Function  --> Split function is basically used to convert a given string into a list.

Syntax:
    string_name.split()
```

Example of Split Function

In [16]:
```
x="learning java is easy"
x.split()
```
Out[16]:
```
['learning', 'java', 'is', 'easy']
```

In [1]:
```
x="Hello World"
x
```
Out[1]:
```
'Hello World'
```

Join() Function

In [ ]:
```
Join Function  --> Join function is basically used to convert a given list into a string.

Syntax:
    empty_string.join(list_name)
```

Example of Join Function

In [4]:
```
x=['learning', 'java', 'is', 'easy']
y=" ".join(x)
y
```
Out[4]:
```
'learning java is easy'
```

In [5]:
```
x=['hello', 'world']
y=" ".join(x)
y
```
Out[5]:
```
'hello world'
```

Input Function

In [ ]:
```
input() --> It is used to take the input from the user  in  the form of string.
```

Example of Input Function

In [6]:
```
x=input("Enter a number")
print(type(x))
```
```
Enter a number10
<class 'str'>
```

In [7]:
```
x=int(input("Enter a number"))
print(type(x))
```
```
Enter a number10
<class 'int'>
```

ASCII/UNICODE VALUES

In [ ]:
```
--> According the to American standard coding information interchange each and every alphabet is a
    asscociated with its unique value of integer and that unique value is known as Ascii Value.

--> Each and every alphabet is associated with its integer unqiue value is known as ascii value.

Examples:

    Character                                          Ascii value/Unicode Values

       A                                                   65
       B                                                   66
       C                                                   67
       ...                                                 ...
       ...                                                 ...
       Z                                                   90

       a                                                   97
       b                                                   98
       c                                                   99
       d                                                   100
       ...                                                 ...
       ...                                                 ...
       z                                                   122
```

ord () Function

In [ ]:
```
--> for getting ascii value for a character In python we are having ord function. and This
    ord function is used to get the ascii value of a character.

Syntax:
    ord(Character_Name)

Note --> Ord function is only expecting a single character if you are trying to give more than one
         character than you will get an error.
```

Example of Ord Function

In [8]:
```
print(ord("L"))
print(ord("#"))
print(ord("%"))
print(ord("A"))
#print(ord("ABC"))           #Error
```
```
76
35
37
65
```

chr Function

In [ ]:
```
--> Based on Ascii Value if you want to find the character then we should use this chr Function.

Syntax:
    chr(Ascii_Value)
```

Example of chr Function

In [10]:
```
print(chr(49))
print(chr(99))
print(chr(199))
print(chr(2999))
print(chr(5999))
```
```
1
c
Ç
ஷ
```

Practice Programs based on Ascii Values

Python Program to find the Character(UpperCase Alphabet) based on Ascii Values.

In [11]:
```
for i in range(65,92):
    print("Ascii/Unicode value of "+str(chr(i))+" is "+str(i))
```
```
Ascii/Unicode value of A is 65
Ascii/Unicode value of B is 66
Ascii/Unicode value of C is 67
Ascii/Unicode value of D is 68
Ascii/Unicode value of E is 69
Ascii/Unicode value of F is 70
Ascii/Unicode value of G is 71
Ascii/Unicode value of H is 72
Ascii/Unicode value of I is 73
Ascii/Unicode value of J is 74
Ascii/Unicode value of K is 75
Ascii/Unicode value of L is 76
Ascii/Unicode value of M is 77
Ascii/Unicode value of N is 78
Ascii/Unicode value of O is 79
Ascii/Unicode value of P is 80
Ascii/Unicode value of Q is 81
Ascii/Unicode value of R is 82
Ascii/Unicode value of S is 83
Ascii/Unicode value of T is 84
Ascii/Unicode value of U is 85
Ascii/Unicode value of V is 86
Ascii/Unicode value of W is 87
Ascii/Unicode value of X is 88
Ascii/Unicode value of Y is 89
Ascii/Unicode value of Z is 90
Ascii/Unicode value of [ is 91
```

Python Program to find the Character(Decimal numbers) based on Ascii Values

In [12]:
```
for i in range(48,59):
    print("Ascii/Unicode value of "+str(chr(i))+" is "+str(i))
```
```
Ascii/Unicode value of 0 is 48
Ascii/Unicode value of 1 is 49
Ascii/Unicode value of 2 is 50
Ascii/Unicode value of 3 is 51
Ascii/Unicode value of 4 is 52
Ascii/Unicode value of 5 is 53
Ascii/Unicode value of 6 is 54
Ascii/Unicode value of 7 is 55
Ascii/Unicode value of 8 is 56
Ascii/Unicode value of 9 is 57
Ascii/Unicode value of : is 58
```

Python Program to find the Character(LowerCase Alphabet) based on Ascii Values

In [13]:
```
for i in range(97,123):
    print("Ascii/Unicode value of "+str(chr(i))+" is "+str(i))
```
```
Ascii/Unicode value of a is 97
Ascii/Unicode value of b is 98
Ascii/Unicode value of c is 99
Ascii/Unicode value of d is 100
Ascii/Unicode value of e is 101
Ascii/Unicode value of f is 102
Ascii/Unicode value of g is 103
Ascii/Unicode value of h is 104
Ascii/Unicode value of i is 105
Ascii/Unicode value of j is 106
Ascii/Unicode value of k is 107
Ascii/Unicode value of l is 108
Ascii/Unicode value of m is 109
Ascii/Unicode value of n is 110
Ascii/Unicode value of o is 111
Ascii/Unicode value of p is 112
Ascii/Unicode value of q is 113
Ascii/Unicode value of r is 114
Ascii/Unicode value of s is 115
Ascii/Unicode value of t is 116
Ascii/Unicode value of u is 117
Ascii/Unicode value of v is 118
Ascii/Unicode value of w is 119
Ascii/Unicode value of x is 120
Ascii/Unicode value of y is 121
Ascii/Unicode value of z is 122
```

Python Program to find the Ascii Value of Each Character of a String

In [15]:
```
x=input("Enter a String")
for i in x:
    print(ord(i),end=" ")
```
```
Enter a StringPython
80 121 116 104 111 110
```

Python Program to Solve the Below Problem:

In [ ]:
```
if A=70, B=71, C=72, D=73
    please find the sum of APCD?
```

In [16]:
```
x=input("Enter a String")
sum=0
for i in x:
    y = ord(i)
    sum=sum+y
print(sum)
```
```
Enter a StringAPCD
280
```

IsAlpha() Function

In [ ]:
```
isalpha --> return True if the given string is containing only alphabets
```

Examples of isalpha Function

In [17]:
```
x="helloWorld"
print(x.isalpha())
```
```
True
```

In [18]:
```
x="hello World"
x.isalpha()
```
Out[18]:
```
False
```

Sorted Function

In [ ]:
```
sorted function --> Sorted Function is Used to Sort a Given String.
                --> Sorted Function always return a list.
```

Examples of Sorted Function

In [21]:
```
x="123AratyQui"
sorted(x)
```
Out[21]:
```
['1', '2', '3', 'A', 'Q', 'a', 'i', 'r', 't', 'u', 'y']
```

In [22]:
```
x="HelloWorld"
sorted(x)
```
Out[22]:
```
['H', 'W', 'd', 'e', 'l', 'l', 'l', 'o', 'o', 'r']
```

Practice Problems Based on Strings

Python program to Check weather a Given String is Palindrome or not

In [ ]:
```
Palindtome String or Not:
ABCDCBA --> ORGINAL STRING
ABCDCBA --> REVERSE STRING
ORGINAL == REVERSE STRING
PALINDROME STRING
MADAM #PEEP #MALYALAM #NOON
```

In [23]:
```
String_1=input("Enter Original String: ").lower()
Reverse_string_1 = String_1[::-1]
if String_1==Reverse_string_1:
    print("Palindrome String")
else:
    print("Not Palindrome")
```
```
Enter Original String: Madam
Palindrome String
```

Python Program to Check weather a given string is Anangram or Not

In [24]:
```
#Anagram or Not   exmaple--> Listen --> siLent , RACEs-->CAREs
string_1=input().upper()
string_2=input().upper()
if sorted(string_1)==sorted(string_2):
    print("It is an anagram string")
else:
    print("It is not an anaagram string")
```
```
Listen
Silent
It is an anagram string
```

Introduction to List

In [ ]:
```
List --> if we want to store a group of dissimilar elements as a single entity then we should
         go for list.
     --> Indexing is very important incase of list. Because we will differentiate elements based
         on index value.
     --> Duplicates are allowed in list.
     -->List is dynamic in nature that means we can add or delete elements from the list(Mutable)
```

Creation of List Object

In [ ]:
```
Creation of list object:
    l = []  #square bracket
    l = [10,20,30,40] #if you know the element
    l = list()
```

In [60]:
```
l=[]
print(type(l))
```
```
<class 'list'>
```

In [61]:
```
l=[10,20]
print(type(l))
```
```
<class 'list'>
```

In [62]:
```
l=list()
print(type(l))
```
```
<class 'list'>
```

How we can access the elements of a list?

In [ ]:
```
1.Using indexing
2.Using slicing
```

Practice Questions of Indexing and Slicing

In [26]:
```
#indexing syntax --> list_name[index]
#slicing syntax --> list_name[begin_index : end_index : step]
x=[10,20,30,40,50]
print(x[1])   #20
#print(x[5])   #error
#print(x[80])   #error
print(x[::])   #whole list
print(x[::-1])  #reverse of list
print(x[::-3])  #[50,30,10]
print(x[10:9:])  #
print(x[2:9])
print(x[-5:-1:-2])
print(x[-5:-1:2])
print(x[1:-1:2])
print(x[5:0:3])
print(x[4:0:3])
print(x[5:0:-2])
print(x[-1:-9:1])
print(x[2:3:9])
```
```
20
[10, 20, 30, 40, 50]
[50, 40, 30, 20, 10]
[50, 20]
[]
[30, 40, 50]
[]
[10, 30]
[20, 40]
[]
[]
[50, 30]
[]
[30]
```

In [ ]:
```
Note: step --> if step is 1 you need not to skip anything
               if step is 2 then you need to skip one element.
               if srep is 3 then you need to skip two elements.
```

In [ ]: