In [ ]:

```
#Agenda of Today:
                  1. Strings in Python
                  2. Introduction to Data Structures
                  3. Problem Solving
```

In [ ]:

```
#Strings:
 Def: String is Array Sequence of Characters.

        #How to create string??
         We create strings by enclosing characters inside a single or double quotes.

#Note: There is no Char Data type in Python
```

In [11]:

```
#How to create strings?
s = " "   #empty string
print(type(s))
s = 100
print(type(str(s)))
s1 = str()
print(s1)
s = "Apssdc"
print(s)
s = 'python'
print(s)
s = """This is Online session About Python
           Programming By APSSDC
                  to all Over AP"""
s = "A"
print(s)
type(s)
```

```
<class 'str'>
<class 'str'>

Apssdc
python
A
```

Out[11]:

```
str
```

In [25]:

```
#How to access characters in strings?
s = "Today is Saturday"
#indexing
print(s[0])
print(s[12])   #Postive indexing
print(s[-1])   #negative indexing
#Slicing
print(s[0:5])  #Positive Slicing by ":"
print(s[-5:-1])  #Negative Slicing
print(s[::-1])   #Reverse a string
print(s[5:-2])
print(len(s))
print(s[16])
```

```
T
u
y
Today
urda
yadrutaS si yadoT
 is Saturd
17
y
```

In [ ]:

```
#How to change or delete s strings?
  1. We cant change the characters in strings becuase strings are immutable.
  2. But we can replace the new  string with old string
  3. we also delete a entire string by using del keyword
```

In [31]:

```
s = "Python Programming"
print(s)
s = "Java Programming"
print(s)
#delete a string
del s
#print(s)
```

```
Python Programming
Java Programming
```

In [39]:

```
#String Operations:
#string concatenation:
s1 = "Hello Dear"
s2 = " Python Students"
print(s1+s2)
s3 = ("Hello"
          " Apssdc"
                 " Python"
                       " Session"
                            " Students")
print(s3)
#String Multiplication
print(s1 * 10)
print(s2*3+s1+s2*5)
```

```
Hello Dear Python Students
Hello Apssdc Python Session Students
Hello DearHello DearHello DearHello DearHello DearHello DearHello DearHello DearHello DearHello Dear
 Python Students Python Students Python StudentsHello Dear Python Students Python Students Python Students Py
thon Students Python Students
```

In [52]:

```
#How to iterate through a string:
def countofcharacters():
    s = "Sunday is Holiday there is no session for All"
    count = 0
    for ch in s:
        if ch == "s":
            count = count+1
            print(ch,end=" ")
print("\n")
print(count)
countofcharacters()
```

```

5
s s s s s
```

In [59]:

```
#String Membership Test:
s = "python is very easy"
print("v" in s)
print("w" not in s)
print("Z" in s)
```

```
True
True
False
```

In [64]:

```
#builtin functions
#len()
#enumerate()
s = "Data Science"
print(s)
listenumerate = list(enumerate(s))
print(listenumerate,end="")
```

```
Data Science
[(0, 'D'), (1, 'a'), (2, 't'), (3, 'a'), (4, ' '), (5, 'S'), (6, 'c'), (7, 'i'), (8, 'e'), (9, 'n'), (10,
'c'), (11, 'e')]
```

In [70]:

```
#How to Format the strings?
#using triple quotes
print("""Hai...Mr.Python How are u "What's Up?""")

#escaping single quotes

print('Hai...Mr.Python How are u "What\'s Up?')

#escaping double quotes

print("Hai...Mr.Python How are u \"What's Up?\"")
```

```
Hai...Mr.Python How are u "What's Up?
Hai...Mr.Python How are u "What's Up?
Hai...Mr.Python How are u "What's Up?"
```

In [81]:

```
#format(): its can be done using {} brackets
A = "Machine Learning"
B = "Django"
C = "Data Science"
normal= "{}, {} and {}".format(A,B,C)
print(normal)
positional = "{2},{0} and {1}".format(A,B,C)
print(positional)
keyword = "{X},{Z} and {Y}".format(X="java",Y="Python",Z="C++")
print(keyword)
```

```
Machine Learning, Django and Data Science
Data Science,Machine Learning and Django
java,C++ and Python
```

In [84]:

```
a = 50.5894839333
print("value of a is %50.6f" %a)
```

```
value of a is                                          50.589484
```

In [86]:

```
#String Methods:
 #To know the all methods of string or any sequence we use which builtin function?
    #dir()
print(dir(str),end=" ")
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__', '__format__', '__ge_
_', '__getattribute__', '__getitem__', '__getnewargs__', '__gt__', '__hash__', '__init__', '__init_subclass_
_', '__iter__', '__le__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__', '__reduce__', '__re
duce_ex__', '__repr__', '__rmod__', '__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__',
'capitalize', 'casefold', 'center', 'count', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'format_ma
p', 'index', 'isalnum', 'isalpha', 'isascii', 'isdecimal', 'isdigit', 'isidentifier', 'islower', 'isnumeric',
'isprintable', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'maketrans', 'partition',
'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith',
'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']
```

In [92]:

```
#lower()
#upper()
#join()
#split()
#find()
#replace()
#count()
s = "Web Development Using Django"
s1= "web development using django"
print(s.lower())   #its change the entire string to lower case
print(s.upper())   #its change the entire string to upper case
print(s.capitalize())  #it can upper the string character only
print(s1.title())    #it can capitalize the first character of every word in a string
```

```
web development using django
WEB DEVELOPMENT USING DJANGO
Web development using django
Web Development Using Django
```

In [97]:

```
#count() and find()
s = "Web Development Using Django"
print(s.count("e"))
print(s.find("i"))
print(s.find("o"))
```

```
4
18
9
```

In [124]:

```
#split() and join():
s = "Web Development Using Django"
print(s.split(" "))
print(":".join(s))
s1 = "123456789"
print(" ".join(reversed(s1)))
s2 = "Web Development"
print(s2.split('e'))
print(s2.split("e",1))
```

```
['Web', 'Development', 'Using', 'Django']
W:e:b: :D:e:v:e:l:o:p:m:e:n:t: :U:s:i:n:g: :D:j:a:n:g:o
9 8 7 6 5 4 3 2 1
['W', 'b D', 'v', 'lopm', 'nt']
['W', 'b Development']
```

In [131]:

```
#Replace():
s = "Web Development Using Django"
s.replace("Development","Designing")
#index()
print(s.index("U",1,20))     #index(string,start,end)
```

```
16
```

In [136]:

```
#center():  It returns a string which is padded with the specified character.
s = "Python is Awesome"
print(len(s))
s1=s.center(24,"$")
s2 =s.center(28,"*")
print(len(s1))
print(s1)
print(s2)
```

```
17
24
$$$Python is Awesome$$$$
*****Python is Awesome******
```

In [ ]:

```
#String boolean functions:
#islower()
#isupper()
#isalpha()
#isalnum()
#isdecimal()
#isdigit()
#isnumeric()
#isspace()
#isprintable()
#istitle()
```

In [151]:

```
s = "apssdc"
print(s.islower())
s1 = "Apssdc"
print(s1.islower())
s2 ="DAD"
print(s2.isupper())
s3 = "12345"
s4 = "String"
print(s3.isnumeric())
print(s3.isdigit())
print(s4.isalpha())
s5 = "123str"       #alpha numerical string
print(s5.isdigit())
print(s5.isnumeric())
print(s5.isalpha())
print(s4.isalnum())
```

```
True
False
True
True
True
True
False
False
False
True
```

In [161]:

```
s = " "
print(s.isspace())
s = "100"
print(s.isprintable())
ss = "Abc Daadkdlksdl"
print(ss.istitle())
```

```
True
True
True
```

In [178]:

```
#strip(): it can remove the white spaces at both left and right sides
s = "  Code is Enjoying "
print(s.strip())
print(s.lstrip())  #removes only at left side
print(s.rstrip())  #removes only at right side
```

```
Code is Enjoying
Code is Enjoying 
  Code is Enjoying
  Code is Enjoying
```

In [186]:

```
#startswith and endswith
s = "python iS easy to Debug"
print(s.startswith("python"))
print(s.endswith("python"))
print(s.startswith("p"))
print(s.swapcase())
```

```
True
False
True
PYTHON Is EASY TO dEBUG
```

In [197]:

```
#zfill(): it returns a copy of the string with "0" characters padded to the left.
s = "coding is fun"
print(len(s))
s1=s.zfill(5)
print(s1)
```

```
13
coding is fun
```

In [195]:

```
s= "python is !\n \\ Easy #637484959??"
s.isprintable()
```

Out[195]:

```
False
```

In [ ]:

```
9.#input: s = "A P S S D C P Y T H O N"
#output: s =n O H T Y P C D S S P A
```

In [ ]:

```
#Day- 6 (Today Tasks)

1. find the given number is palindrome or not.
2. check the given number is prime or not.
3. Print the given year is leap or not.
4. print the leap years in given range of years.
5. print the math table as up to given number
6. To check the given number is positive or not.
7. Print the swaping of given actual numbers
8. Program to do the basic calculator operations.
9. Program to print the given string as like output string
   #input: s = "A P S S D C P Y T H O N"
   #output: s =n O H T Y P C D S S P A
```

In [196]:

```
s = "python"
s.zfill(1)
```

Out[196]:

```
'python'
```

In [198]:

```
s="APPSSDCPYTHON"

print(" ".join(reversed(s)))
```

```
N O H T Y P C D S S P P A
```

In [ ]: