In [ ]:
```
#Agenda of Today :
                1. Function arguments in Python
                2. Strings in Python
                3. Data Structures in Python
```

In [ ]:
```
    what is argument?
        what is parameter?
          Difference b/w them ?
            type of arguements?
             What is the use of arguements?
```

In [ ]:
```
#Parameter?
A parameter is a named variable passed into a functions.
Parameter Variables are used to import arguments into functions.
#Argument?
 Its nothing but passed variable to function which having actual values.
```

In [ ]:
```
#Difference Notes:
 1. Parameters are names listed in the functions defintion.
 2. Arguments are the real values passed to the function.
```

In [1]:
```
#Example on Arguments?
def sumof2(a,b):  #Function declation   #formal parameters
    c = a+b
    print(c)
x = 20
y = 80
sumof2(x,y)  #function calling   #x,y are actual parameters
```

```
100
```

In [ ]:
```
#Type of Arguments:
1. Positional
2. Keyword
3. Default
4. Variable-Length or Arbitary
```

In [15]:
```
#Ex: Positonal (Its deal with order of passing arguments)
def Student(Name,Id):
    print("Student Name:",Name)
    print("Student ID:",Id+100)
Student("Surya",501)
```

```
Student Name: Surya
Student ID: 601
```

In [17]:
```
#Keyword Arguments: (we attached values to specific keywords at function calling)
def Student(Name,Id):
    print("Student Name:",Name)
    print("Student ID:",Id+100)
Student(Id=501,Name="Surya")
```

```
Student Name: Surya
Student ID: 601
```

In [43]:
```
#Default Arguements: (We fixed the values of arguments defaultly at function calling)
def Student(Name,Id=501):
    print("Student Name:",Name)
    print("Student ID:",Id+100)
Student("Aditya")
```

```
Student Name: Aditya
Student ID: 601
```

In [40]:
```
#Variable-Length Arguments or arbitary:  (We when deal with Single Varaible with multiple Values)
def DoSum(a, *B):
    print(a)
    print(B)
    c = a
    for i in B:
        c = c+i
    print(c)
DoSum(10,20,30,40,50,60,70,80,90)
```

```
10
(20, 30, 40, 50, 60, 70, 80, 90)
450
```

In [39]:
```
def DoSum(*B):  #Arbitary arguement
    c = 0
    for i in B:
        c = c+i
    print(c)
DoSum(10,20,30,40,50,60,70,80,90)
```

```
450
```

In [41]:
```
#Ex: arbitary Arguments:
def IndianTeam(*Players):
    print(Players)

IndianTeam("Dhoni","Bumrah","Kohli","Rohit","Dhawan","Pant")
```

```
('Dhoni', 'Bumrah', 'Kohli', 'Rohit', 'Dhawan', 'Pant')
```

In [44]:
```
#Strings in Python:

A string is a Sequence of Unicode characters.
```

In [ ]:
```
#Encoding - Conversion of character to number (0,1)
#Decoding - Conversion of number(0,1) to character
   In this process they follow some conversion formats:
         1. Ascii
         2. Unicode
```

In [56]:
```
#How to create a string in python?
#- Strings are created by enclosing any characters inside a single or double-quotes
#- Even we may use triple quotes also they called as doctstring or muliple string

s = "Today"
s1 = 'Tomorrow'
s2 = """Hello everyone this program about demo function
            in python programming
              from APSSDC """
print(s)
print(s1)
print(s2)
print(str)
```

```
Today
Tomorrow
Hello everyone this program about demo function
            in python programming
              from APSSDC
python
```

In [52]:
```
def demo():
    """Hello everyone this program about demo function
            in python programming
              from APSSDC """              #Doc string or Multiline string
    print("Hello IIIT Students")
demo()
```

```
Hello IIIT Students
```

In [60]:
```
#Empty string
s3 = " "
s3
```

Out[60]:
```
' '
```

In [ ]:
```
#How to Access characters in string:
(By using Indexing and Slicing and Loops)
```

In [81]:
```
s1 = "programming"  #index range (0 to n-1)
print(s1[0])   #Positive (left to right)
print(s1[-1])  #Negative (Right to left)
print(s1[-2])
```

```
p
g
n
```

In [78]:
```
print(s1[5])
print(s1[14])
```

```
a
```

In [79]:
```
len(s1)
```

Out[79]:
```
15
```

In [92]:
```
#Slicing:
s1 = "programming"
#print(s1[0:5])  #Positive Slicing
#print(s1[3:11])
#print(s1[-5:-1])
#print(s1[-11:-3])  #Negative Slicing
for ch in s1:
    print(ch,end="*")
```

```
p*r*o*g*r*a*m*m*i*n*g*
```

In [95]:
```
s1 = "programming"
print(s1)
print(s1[::-1])   #reverse the string
```

```
programming
gnimmargorp
```

In [102]:
```
s = "APSSDC Workshops"
#slcing : to get the range or group of required character in a string
print(len(s))
print(s[0:6])
print(s[7:16])
```

```
16
APSSDC
Workshops
```

In [115]:
```
#How to change or delete a string ?
#Note: Strings are Immutable(Not changable) so we cant change or resign to string characters.
s1 = "Coding is fun"
len(s1)
s1[0]="E"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-115-a86e2bd0b22d> in <module>
      3 s1 = "Coding is fun"
      4 len(s1)
----> 5 s1[0]="E"

TypeError: 'str' object does not support item assignment
```

In [117]:
```
del s1
```

In [118]:
```
s1
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-118-d0257f733e5e> in <module>
----> 1 s1

NameError: name 's1' is not defined
```

In [125]:
```
#Python String Operations:
#(Concatenation and Multiplication..etc)
s = "123456*@+_)&"
s1 = 12345
print(s)
print(type(s))
print(type(s1))
```

```
123456*@+_)&
<class 'str'>
<class 'int'>
```

In [137]:
```
#Concatenation and Muliplication( +,*)
str1 = "IIIT "
str2 = " CSE STUDENTS"
print("str1 + str2 = ",str1+str2)
print(str1*5 + "100"+" "+ str2*5)
```

```
str1 + str2 =  IIIT  CSE STUDENTS
IIIT IIIT IIIT IIIT IIIT 100  CSE STUDENTS CSE STUDENTS CSE STUDENTS CSE STUDENTS CSE STUDENTS
```

In [144]:
```
#Iterating through a string:
count = 0
s1 = "Hello IIIT"
for letter in s1:
    if letter == "I":
        count = count+1    #count +=1
print(count)
```

```
3
```

In [149]:
```
#Membership Test : (in, not in)
s1 = "Hello IIIT"
"H" in s1
"Y" in s1
"Hello" not in s1
"Hello"  in s1
"A" in "APSSDC"
```

Out[149]:
```
True
```

In [152]:
```
#String Methods:
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 [ ]:
```

```