#05th-JUNE-22

8th Class - "Python"

"FUNCTION"

In [1]:
```
#Code No 1 :-

# Let's start creating a fucntion by writing (def is nothing but, defination of the function ) then I can try
#to give a function name is test() ( open & close bracket ) which I have given (:) column.

# If we miss to put (:) column then, it will give us an Error.

# Now, "Def test()" - We have created, inside that, whatever we are going to write, whatever logic,
#we are going to write, "Yes". It will try to execute itself as it is.

# Wherever we have to call, i will end up calling a test simple.

# Now lets create a functionis going to input as a (a,b) this is what I am trying to say over here.

# def test (a,b):

# Whenever we are going to call, just we have to make sure that we are suppose to pass a "Value".

# Now whatever operation we are going to internally that depend, but yeah, pass the value over here.
#- (return a+b)

# (return a+b), If this is a function - test(a,b), which I have created, it is trying to return ( a+ b ).
```

In [2]:
```
def test(a,b) :
    return a+b
```

In [3]:
```
# #Code No 2 :-

# if lets suppose there is a function called as "test()"

# If we call test() & if we execute it, it will give me an Error. The region is in the signature of
#function Test(a,b).I am trying to pass a(a,b) when, I am trying to called fucntion.

# Test() here it is not finding any (a,b) that is the region it is giving me an "Error".

# The region is that, in the singnature of the fucntion test(a,b) here, I am trying to pass (a,b) &
#when I am trying To call a fucntion test(), it is not able to find out (a,b) that is the region,
#because of it, it is giving me an Error.
```

In [4]:
```
test()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[4], line 1
----> 1 test()

TypeError: test() missing 2 required positional arguments: 'a' and 'b'
```

In [ ]:
```
# Code No 3 :

# Lets suppose here, I am going to pass test(4,6) = 10 outcome.Yes (4+6) = 10.
```

In [ ]:
```
test(4,6)
```

In [ ]:
```
# Code No 4 :-

# Let's suppose, I am going to pass like "vijay" & may be I can pass another string "gurung".
# It is going to perform in
# the same way.
```

In [ ]:
```
test("vijay","gurung")
```

In [ ]:
```
# Code NO 5 :

# It is also possible that, I can call this fucntion, with the assignment operator.

# test(a=9, b=6)  or test(b = 45, a =3)  If I willchange the order test(b = 45, a =3), even in this case it is
#going to work, order does not matter

# Even in these case, it is going to work order does not matter but. If we are not giving order with respect
# to a particular variable assignment in that case. it is going to take a order that we have assign in a
# original function.

# Like suppose, I Have define test(a,b) & test(4,6).
# I am trying to pass parameter over here it is consider
# Test(a,b)

#(a=4, b =6). It will consider a = 4 & b = 6 bydefault.
```

In [ ]:
```
test(a=9, b=6)
```

In [5]:
```
test(b = 45, a =3)
```

Out[5]:
```
48
```

In [6]:
```
                  # Just for my understanding & knowledge.- ( Topic - "Docsting ")

# Print :- Let's suppose I am going to call Print() function & when I do (shift+Tab) it will basically gives
# us a Docstring : with respect to any function that we have created in Python.

# test(b=45, a=3) :- Let's suppose, I am going to perform same thing with my function. 'YES' it is giving me
#docstring,but it is saying me that < No docstring > at all. Docstring is not available.

# It means that, I have just created the fucntion, but I have not mention why I have created the function
#what is the nature of the function.

# How & what kind of parameter I will be able to pass, if such kind of information or such kind of instruction

# if I have to provide to the user. In that case even we can create our own docstring. JUst like inbuild
# function.


# Every inbuild function having the Docstin, basically its an instrument or readable instruction for an user.

# Where with the help of ("shift + Tab") any user will be ablt to see, that what this function is.

# what kind of input this function is going to take & what kind of outcome function is going to give.

# What is the permutation combination of this function to do that.
# what we have to do is. ( "This is my fucntion for concatination or addition")
```

In [7]:
```
# Explanation below code No 6 :-

# what kind of input this function is going to take & what kind of outcome function is going to give.

# what is the permutation combination of this fucntion to do that. what we have to do is, we have to create a
# ("""  """) Triple coat. Interm of Comas we can use both double coat or Triple Coat both is fine,
# it will work.

# test(4,6) :-
#10

# If now I am going to see ("shift + TAB"), we will be able to find out our own Docsting has been created.

# We can create any kind of a "docstring " or any kind of a information that we would like to showcase
# to the users.we can try to pass all of these things over here.

# This is how we can create our own "Docstring".
```

In [8]:
```
# Code NO 6 :

def test(a, b) :
    "this is my fucntion for concationation or addition"
    return a + b
```

In [9]:
```
test(4,6)
```

Out[9]:
```
10
```

In [10]:
```
# Explaination Below code :

# Lets Suppose, I am going to create a function.

# Def test1(a,b,c,d,e) : - These are the variables, which I have assigned, these are parameter which,
# I have assigned to this particular fucntion.

# Here how we can perform any operation, here we have 5 variable (a,b,c,d,e).

# There is a possibility that, there could be a 10 variables, there could be end number of variable,
# that we have to pass.

# As of now when we are trying to say that (a,b,c,d,e). It will be ablt to consider only (a,b,c,d,e).
# If I am going to pass 6 number of variable, it is not going to accept &
# it is going to give me an Error.

# 1."return a,b,c,d,e" -

# Lets suppose, I am going to do like a "return a,b,c,d,e". If I am going to write this one & then,
# If I am going to call
# the function - test1(3,4,5,6,7,7)

# If we execute it, as a outcome we are geeting an "Error", why because required number of
# parameter test1(a,b,c,d,e).

# And I Have here passed the 6 number of parameter test1(3,4,5,6,7,7).
# might be this is my requiredement let's assum there is a possibilty that in a run time.
# I will end up providing more number of arguments.

# There is a surety that it will end up giving an Error, beacause it was expecting just a
# 5 variable (a,b,c,d,e) & we have
# passed a 6 variable or 6 parameter (3,4,5,6,7,7) this is telling me is that.

# TypeError: test1() takes 5 positional arguments but 6 were given

# Now -> test1(3,4,5,6)

# Let's suppose If I am giving a 5 parameter (3,4,5,6,7) then it is completely fine, it is going to work.

# TypeError: test1() missing 1 required positional argument: 'e' :-

# Once again, if i am going to provide only 4 variable (3,4,5,6) then again it is giving me an Error.
```

In [11]:
```python
# Code NO 7 :

def test1(a,b,c,d,e):
    return a,b,c,d,e
```

In [12]:
```python
test1(3,4,5,6,7,7)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[12], line 1
----> 1 test1(3,4,5,6,7,7)

TypeError: test1() takes 5 positional arguments but 6 were given
```

In [13]:
```python
test1(3,4,5,6)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[13], line 1
----> 1 test1(3,4,5,6)

TypeError: test1() missing 1 required positional argument: 'e'
```

In [14]:
```python
# Explaination Below code :

# Lets suppose -I would like to create a kind of function, where I do not want this kind of limitation.
# So, whatever number of input that I would like to give. I can give it.

# If such kinf of function I have to create so how I will be able to create such kind of function.
# Lets understand that Particular part.

# Its a interesting thing to know, how I will be able to pass a multiple like a parameter at a any point
# of the time.So, how I will be able to pass  a multiple parameter at any point of the time.

# Lets suppose -Here I am going to create a function test(a) instead of writing "a". what I can do is that,
#I can write *(Asterisk) (*a) Asterisk (*) "a" simply means that. Asterisk (*) of anything, I can provide any
# number of parameter that I amlooking for, it is possible to processed. Here is a single (*) of ("a").
# I am going to return ('a').

# def test(*a):
#    return a

# Lets suppose - I am going to call a test(34,56) two parameter & execute it,it will working, because
# I have passed (34,56) Yes, it is ablt to take.

#test(34,56)
#(34, 56)

# Definition - "Agrs" :- is a not a keyword, we can try to use anything, so majority of time if we are going
# to search over the internet,we will be able to find out that people write this way.
```

In [15]:
```python
# Code NO 8 :

def test(*a):
    return a
```

In [16]:
```python
test(34,56)
```

Out[16]:
```
(34, 56)
```

In [17]:
```python
# Lets suppose - I am going to pass 4 parameter or 5 parameter (2,3,4,4) this is fine.
```

In [18]:
```python
test(2,3,4,4)
```

Out[18]:
```
(2, 3, 4, 4)
```

In [19]:
```
# I can pass any parameters of number of parameter does not matter at all. What we are going to pass.
# It will be independent of number of the argument that, we have provided.
# 4,5,6,6,6 is = 1,2,3,4,5 we start counting & [3,4,5,5,5,6] this is consider as only 1.
#so the total parameter of number is 6.I am able to provide.

# This is something called as Asterisk (*) & many function that, we are going to use, we will be able
# to find out that, it written as "Args".
```

In [20]:
```
test(4,5,6,6,6,[3,4,5,5,5,6])
```

Out[20]:
```
(4, 5, 6, 6, 6, [3, 4, 5, 5, 5, 6])
```

In [21]:
```
# Lets Suppose - If we write -  test4(*args):  this way we write, people thing that (args)
# is a keyword over here.
# "Args" is just a representation, instead of using "Args", we can use anything.

# I can use my name or anything "xyz", whatever, I want I can try to use it.
```

In [22]:
```
def test4(*args):
    return args
```

In [23]:
```
# Here I am trying to pass any number of parameter & it is going to return, that number of parameter.This way,
# I end up
# cerating a kind of function, which is going to consider or which is going to take end number of parameter
# at any point of the time.
# It would not be having any kind of constraint with respect to number of parameter that we going to pass.

# So, we understsand the "Args" & how we can try to create a function which can take end number of argument.
```

In [24]:
```
test4(34,56,6,76,7)
```

Out[24]:
```
(34, 56, 6, 76, 7)
```

In [25]:
```
# Code NO 9 :

# Explanation :-

# Lets suppose - I am going to write def test5(*a):
# Just use "a" and I can try to write "a" can take any number of parameter. So, I can try to pass any number
# of parameter, finally it will behave as a "Tuples".

# L = [] - So here I have create blank "List".

# l.append(i)
#    return l
# So, try to append whatever, I am going to write, it is going to append it or whatever I am going to pass,
# it is to return me "List"(L).

# So, Here in below code, I have created a fucntion & I have just pass a Asterisk(*) of "a" or "Args" whatever.
# I want out of these 2 its just a name not a keyword. I can write anything.
```

In [26]:
```
def test5(*a):
    l = []
    for i in a :
        l.append(i)
    return l
```

In [27]:
```
# So, I am trying to call test5(2,3,4,45,5) + [345,56,5,6] pass list something.

# (2,3,4,45,5,[345,56,5,6]) - Here we will be find out that as for the function defination,
# creating a "List" L = [].

# Going through entire input - (*a), entire input will be in the form of "Tuples" & Yes, I will be "iterate"
# Tuples for "i" in "a". & then one by one, I can append it -  L.append(i).
# So, one by one I am able to extract the data from "a", whatever value of "a" that, I have the I am returing

# a "List". so here is the "List" I am able to get - [2, 3, 4, 45, 5, [345, 56, 5, 6]].
```

In [28]:
```
test5(2,3,4,45,5,[345,56,5,6])
```

Out[28]:
```
[2, 3, 4, 45, 5, [345, 56, 5, 6]]
```

In [29]:
```
test5(3,4,5,56)
```

Out[29]:
```
[3, 4, 5, 56]
```

In [30]:
```
# Code NO 10 :-

# Here How I will be able to deal with the function, if I have Asterisk(*) or something means,
# multiple argument function + specific argument.
# Then return may be - return a,b,c,d & Asterisk (*) m.
# Now here,if I am going to call these function test6() over here.

# If I am going to pass (3,4,5,5,6,6,7,7,8,8) if we execute it.

# As a outcome we can see over here. -  (3, 4, 5, 5, (6, 6, 7, 7, 8, 8))
# As we can see

# a,b,c,d * m
# 3,4,5,5 * (6, 6, 7, 7, 8, 8)

# As we can see over here 1st 4 things or 4 variable is normal argument that we have consider
# - a,b,c,d = 3,4,5,5 & (6, 6, 7, 7, 8, 8) - This one is having multiple string(str) that we can take.
# As we know (6, 6, 7, 7, 8, 8) is (Tuples).

# (6, 6, 7, 7, 8, 8)  This one is multiple string because of we have Asterisk (*). So as a outcome everything,
# it converted into a "Tuples".
```

In [31]:
```
def test6(a,b,c,d,*m):
    return a,b,c,d,m
```

In [32]:
```
test6(3,4,5,5,6,6,7,7,8,8)
```

Out[32]:
```
(3, 4, 5, 5, (6, 6, 7, 7, 8, 8))
```

In [33]:
```
# Code NO 11 :-

# This is the function that, I have created. - test7(4,5,65,6,6,7,7,56,67,34,54,67,78)
# If we execute it why we are getting an Error this time, when I have written code, previously that Asterisk(*)
# in last.

# that time, I was able to pass by data. But this time when, I have written Asterisk (*) infront. As a outcome,
# I am geeting an Error.

# The region is that the all elements that we have - (4,5,65,6,6,7,7,56,67,34,54,67,78) conside as a part of
# Asterisk (*) (*m) multiple input variable & for these one (a,b,c,d,e)  it is not able to get anything.

# How to solve these kind of situation lets see. next line code.
```

In [34]:
```
def test7(*m,a,b,c,d,e):
    return m,a,b,c,d,e
```

In [35]:
```
test7(4,5,65,6,6,7,7,56,67,34,54,67,78)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[35], line 1
----> 1 test7(4,5,65,6,6,7,7,56,67,34,54,67,78)

TypeError: test7() missing 5 required keyword-only arguments: 'a', 'b', 'c', 'd', and 'e'
```

In [36]:
```
# Code NO 12 :-

# Let's try to undersatand that, How to solve above kind of situation lets try, below code.

# To solve this kind of situation, what we can do is, if I can try to provide or assign the named one
# in that case, it is going to work.

# In this code I am going to give the named assignment, test7(4,5,65,6,6,7,7,56,a=67,b=34,c=54,d=67,e=78)
# otherwise,
# It is going to give me an Error. Because, it consider that all the data belong to basically Asterisk(*)
# of "m".
# So these time we have written these way {return m,a,b,c,d,e } instead of these earlier one -
# {return a,b,c,d,e,m}
```

In [37]:
```
# For Understand :-

# Lets suppose, I have create a fucntion & over there, I have to pass a variable as well as its value,
# may be a key value,base data, if I have to pass.

# In that situation what I will do, interm of providing a individual parameter for that, I can try to create
# a fucntion.
# then, inside a function I can try to give Asterisk(*) of something that will consider a multiple argument,
# multiple data. I will be pass, without even declearing variable over here.
```

In [38]:
```
def test6(a,b,c,d,*m):
    return m,a,b,c,d,e
```

In [39]:
```
test7(4,5,65,6,6,7,7,56,a=67,b=34,c=54,d=67,e=78)
```

Out[39]:
```
((4, 5, 65, 6, 6, 7, 7, 56), 67, 34, 54, 67, 78)
```

In [40]:
```
# Code NO 13 :-
# Let's Suppose I have to pass a keyword kind of things, If I have to pass. In that case what I will do.
# Keyword argument, I have to pass.
# There can be situation, where I would put up 2 (**) Asterisk. - test8(**)

# So, there is a possibility that, I have to pass a dataset, may be as a dictionary, where we have key value,
# "key value" kind of situation. How I will be able to pass a such kind of data.
```

In [41]:
```
# Lets try to undersand :-

# We have created - def test8(**) double Asterisk(*) is notification or double asterisk (*) is a
# implementation for key value kind of input. End numnber of key value input if I have to give
# in that case.

# Because the thing is that we are able to create (*m) & we are able to pass the data (4,5,65,6,6,7,7,56).

# So, it is trying to consider the entire dataset as a proper collection. It is trying go through a particular

# Collection. we have to do whatever have to do, whole collection of dataset.
```

In [42]:
```
def test8(**)
```

```
  Cell In[42], line 1
    def test8(**)
              ^
SyntaxError: invalid syntax
```

In [43]:
```
# Code NO 14 :-

# Let's suppose -

# I would like to create a specific variable as a key value in a runtime.
# So, it would like to create "Z" & its value "a" & its value "b" & its value in a run time.

# When I am trying to call the fucntion not do that, what we can do is that we can use keyword argument
# over here.

# def test8(** Kwargs)

# "Kwargs" - is what we can find out everywhere in the internet.But this is not a keyword, even. If I am going
# to use my name over  here, it is going to work.

# "Kwargs - Vijay" - Instead of "kwargs" I can use "vijay" or any name i want to use over here.

# whenever we see keyword argument or "args" like a things its nothing it's a notation. Notation is basically
# calling from Asterisk (*).

# Single Asterisk(*) & double Asterisk(*) after that whatever name we are going to give does not matter at all.

# If I will write (** vijay) & then if, I will try to write ( "return Vijay").

# Then I can try to call test8(4,5,6,6). Try to pass multiple parameter. As a outcome "Error" why?

# Because, I have used double (**). I have not used single (*) over here.Incase, single (*) it is going
# to work well.

# But incase of double (**) Asterisk. what happend is, it will be expecting a key value kind of a pair or
# Dictionary kind of object, we are suppose to pass.

# If we are going to pass a dictionary kind of object, we are suppose to pass.If we are going to pass a
# dictionary kind of object then it is completely fine.

# key value kind of object, if we are going to pass then, it is completely fine. Otherwise, it is not consider.

# So, kind of object as we all know that, disctionary is basically a combination of key & values.
```

In [44]:
```
def test8(**vijay):
    return vijay
```

In [45]:
```
test8(4,5,6,6)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[45], line 1
----> 1 test8(4,5,6,6)

TypeError: test8() takes 0 positional arguments but 4 were given
```

In [46]:
```
# Code NO 15 :-

#Explanation below code part-1 :-

# test8(b=4,c=5,d=6,e=6)

# Now as a outcome we can see its working - {'b': 4, 'c': 5, 'd': 6, 'e': 6}

# Can we say that, it is trying to accumulate all the input as an dictionary. It is trying to accumulate
# everything  as a dictionary over here.
# Explanation part-2 :-

# Let's suppose, I am going to consider these dictionary "D" & I am going to call test8. If I am going to pass
# Dictionary(d) test8(d).

# As a outcome, we are getting an Error.  ["d"] is basically like a dictionary that, i have test8(d),
# if I am going to pass a dictionary, it is not going to work.

# It is going to give return as a disctionary(d), but we have to pass a value as a key & value pair with the
# name variable. Otherwise, it is not going to work.
```

In [47]:
```
test8(b=4,c=5,d=6,e=6)
```

Out[47]:
```
{'b': 4, 'c': 5, 'd': 6, 'e': 6}
```

In [48]:
```
d = {'b': 4, 'c': 5, 'd': 6, 'e': 6}
```

In [49]:
```
test8(d)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[49], line 1
----> 1 test8(d)

TypeError: test8() takes 0 positional arguments but 1 was given
```

In [50]:
```
# Code NO 16 :-

# Now, we can understand that, If we have to pass or if we have to pass variable or end number of data
# in a run time & with the name one.

# That with the name one.So, that with the variable name & its respective dataset.

# Below code we can see, I can try to pass any number of variables over here.

# As a "List", "Tuples", "Disctionary" whatever, I want I can try to pass it.

# As a outcome we can see, we just have given a name variable & then we have to pass the data inside (test8)
# then, It will work.
```

In [51]:
```
test8(b=4,c=5,d=6,e=6,n=[3,4,45,5,6,6,6])
```

Out[51]:
```
{'b': 4, 'c': 5, 'd': 6, 'e': 6, 'n': [3, 4, 45, 5, 6, 6, 6]}
```

In [52]:
```
# Code NO 17 :-

# Let's understand :-

# How we can try to merged both, now we can try to use a single Asterisk (*) & double Asterisk (*) at a time.

# Here we can call - def test9(*m, **v) or we can write this way - def test9(*args, **kwargs).

# In general peoples use "m" as a "Args" & "V" as a "kwargs".

# But its not mendatory to use - [Args & Kwargs].

# Its completely fine, we can try to use our own meaningful name & based on that, we will be able to perform
# same exact Same task.

# Note :-

# Single(*) & double(*) is basically for the "Notation purpose".

# Single Asterisk(*) simply means that, we will be able to take any number of variable & end number of variable
# system try to consider those variable as a "Tuples".

# Double Astrisk (**) means we will be able to pass the dictionary.

# As a outcome we can see that its an "Error". Because, I have not given function name over here, (test9)
# we have to give but, I have given only (test) so Error.
```

In [53]:
```
def test9(*m, **v):
    return m,v
```

In [54]:
```
test(345,45,56,5,6,56,b=4,c=6,d=7,g=9)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[54], line 1
----> 1 test(345,45,56,5,6,56,b=4,c=6,d=7,g=9)

TypeError: test() got an unexpected keyword argument 'b'
```

In [55]:
```
# Explanation below code :-
# Here are the things, it is consider for -

# Single (*'m') is - (345,45,56,5,6,56)

# Double (**"V") is - (b=4,c=6,d=7,g=9)

# This is now, I can pass the data, if I have to.

# If I have to create a fucntion where, I can try to pass a data. In these particular way in a key value
# way in a key value way & may be these way we can create.
```

In [56]:
```
test9(345,45,56,5,6,56,b=4,c=6,d=7,g=9)
```

Out[56]:
```
((345, 45, 56, 5, 6, 56), {'b': 4, 'c': 6, 'd': 7, 'g': 9})
```

In [57]:
```
# Code NO 18 :-

# Question :-

# Create a function, which will be able to take end number of input in a run time & it should be able
# to give a multiplication of each & every input.

# May be submission of each & every input or may be concatenation of each & every input.
# If these is a case, we believe we can do it.

# Because, whatever we are going to pass as a Asterisk (*) (*m), system always try to contain that as a "Tuples
# we all know "Tuples" is a collection which can hold any kind of information & we can iterated over Tuples &
# we can try to perform any kind of operation on top of that.

# My question is simple that :-

# Simple that, I have to create a function where. I can take nultiple input at the end of the day. I am looking
# for a submission of all the input.

# Let's suppose all the input are integers let's check.
```

In [58]:
```
# Below Code Explanation :-

# Let's understand :-

# def test10(*m): I have created n = 0 ( so, try to perform a submission operation or any kind of operation
# that,I am Looking for.

# If I have to do submission operation, I have defined a variable ( n = 0) for i in m : I can write.

# if type(i) == int : - so that case try to perform submission operation or any kind of operation that
# I am looking for.
```

In [59]:
```
def test10(*m):
    n = 0
    for i in m :
        if type(i) == int :
            n = n+i
    return n
```

In [60]:
```
# Now I can try to call test10 inside that, I can pass multiple data (2,3,4,5,5,6,6).
# Now as a outcome, we can see that (31).
```

In [61]:
```
test10(2,3,4,5,5,6,6)
```

Out[61]:
```
31
```

In [62]:
```
# Now i can pass 3 digit or end number of data & it will work.
```

In [63]:
```
test10(2,3,4)
```

Out[63]:
```
9
```

In [64]:
```
# Explanation Below Code :-

# because, the thing is def test10(*m) :
# (*m) this is a "Tuples" when we try to give a data as a input, it's a "Tuple".
# Now, we know how to deal with "Tuples".

# If I have to perform any short of operation how I will be able to perform any short of operation with the
# help of "Tuples"
```

In [65]:
```
# Code NO 19 :-

def test1(*m):
    n = 1
    for i in m :
        if type(i) == int :
            n = n*i
    return n
```

In [66]:

```
test1(4,5,6,6,7,87,8)
```

Out[66]:

```
3507840
```

In [67]:

```
# Just for understanding :-

# Now this code also multiple wise same. If I have to do multiplication, I have to change, we can see below.

# Test10 to (Test1)
# n = 0 to (n=1)
# n = n+i to (n=n*1)

# test1(4,5,6,6,7,87,8)
# 3507840

# If I try to pass Test1(4,5,6,6,7,87,8) inside as a outcome we can see, I am able to do a multiplication.

# So, here we are able to create function, which is good. We can try to create function, we write a logic
# inbetween & then we can try to call a return, which if going to give me an outcome.
```

In [68]:

```
# Code NO 20 :-

# Let's Undersatnd below code.

# Now here, there is a another kind of fucntion that we can try to create & that is something call as a Lambda
# fucntion or something called as Anonymous function.

# Let's Understand what is Lambda function & Anonymous function.

# There is a keyword called Lambda, that we all ablt to find out.
# Incase of Lambda, the way we provide an argument the way we pass the data incase of our last fucntions.
# Similar way we can pass an argument or we can try to pass a data even incase of lambda fucntion.
# Why we are using Lambda basically it call as Anonymous in line fucntion or Lambda function.

# So, many people say, can you create a "Anonymous fucntion" someone say can you create "Lambda fucntion",
# someone say can you create "Inline function".

# Why people have given these many names & what happens when we create "Lambda fucntion."

# "Function" :-  means I will give some input & I will get some outcome. This is fucntion.

# Below code :-

# Lets suppose, If I am going to call this fucntion, "n" is what "n" is a basically fucntion over here.
# Let's suppose n(4,5) If I am going to pass n(4,5) is 9.

# Can I say that this fucntion is equvalent to below fucntion, because same outcome is [9].

# Functionality wise both equvalent both trying to give me same additional of (a+b).Functionality wise both
# of these two function are same.
```

In [69]:

```
# Explanation :-

# n = lambda a,b:a+b

# What I am trying to do over here is that, we can try to create a fucntion even in a different different way.
# This is called as Anonymous fucntion or its called as a Lambda fucntion.

# Lambda is basically reserved keyword whenever, we are trying to craete such kind of fucntion always try
# to use Lambda keyword in a begining, because then only Python complair will understand that,
# what I am trying to do.

# Basically, I am trying to create a "Anonymous" function, because there is no name.

# We can see, I have not given any name to this fucntion. -  ( Lambda a,b:a+b )

# What I did is, I have just assign this with one of the variable is "N"
# That variable behave as a name or whatever we can say & I am trying to call it. -> n(4,5) outcome is 9.

# Here in this code, I have given a name "Test5" that is "True"

# n = lambda a,b:a+b ->

# Here I  have not given any name, so that is called as "Anonymous". A function without a name, we can try
# to call anytime.with the help of these variable name "n".

# Because, I am trying to store, these data inside a variable "n".

# n = lambda a,b:a+b
# def test5(a,b):

# Now here parameter wise or argument wise, I am able to pass (a,b) -> [Lambda a,b :]

# Left side of these (:) column, I am able to pass a parameter just like this one -> [test5(a,b):]
# [:a+b] & return a+b

# Here, I am able to write a logic, whatever, I have to written, I am able to write it down.
```

In [70]:
```
# So, Basically these two fucntion are equvalent to each other interms of functionality.
# But interms of nature, function are like a different one is fully define function.

# Difference is that :-

# Always try to avoid write a complex logic to Lambda function, we can write a complex code in Lambda fucntion,
# but try to do not do that, its make confuse you or headache.

# Try to write complex code in def test function, it will little bit easy for us.
# But its a person to person depend, in which we will going to write a complex code.
```

In [71]:
```
n = lambda a,b:a+b
```

In [72]:
```
n(4,5)
```

Out[72]:
```
9
```

In [73]:
```
def test5(a,b):
    return a+b
```

In [74]:
```
test5(4,5)
```

Out[74]:
```
9
```

In [75]:
```
n(5,6)
```

Out[75]:
```
11
```

In [76]:
```
# Code NO 21 :-

# Let's Undersatnd below code :-

# I can call it anytime that, I want "N" is variable where, I have assign (5,6) 11.
# Reuseability wise, I can make a reuseable, I can call it as number of many times.It is not going
# to complain at all.
```

In [77]:
```
n(5,6)
```

Out[77]:
```
11
```

In [78]:
```
# Below Code Explanation :-

# Let's suppose If I would like to pass a multiple paramater.To pass multiple parameter keyword called single
# (*)Asterick. Single Asterick(*) & then may be, I can try to write *Vijay & then try to return (Vijay)
# that Completely fine.

# I can store entire fucntion into a variable ["b"].

# If I am going to call --->  b(45,5,4,5,5,6,677)

# Outcome we can say that behaviour of these fucntion -> b = lambda*vijay:vijay is exact same as the fucntion
# that,we have already study.
```

In [79]:
```
b = lambda*vijay:vijay
```

In [80]:
```
b(45,5,4,5,5,6,677)
```

Out[80]:
```
(45, 5, 4, 5, 5, 6, 677)
```

In [81]:
```
# # Code NO 22 :-

# Let's Understand below code :-

# But Lambda function, let's try to understand some comprehensive operation, how it happens.

# First let's understand comprehensive operation then later back to again lambda fucntion.

# Let's understand comprehensive operation :-
```

In [82]:
```
# Question :-

# There is a "Tuples" "t" we have basically these tuples (3,4,4,5,5,6,67,7,7) & we have to convert these
# entire this into a "List".

# Yes, "List" is a fucntion I can try to use, I can go & iterate through a iterator then, I will be able to
# perform a same thing. There is a another way by which I will be able to perform same thing called as
# "Comprehensive operation".
# It can be done on a Multiple different different kind of variables or "list" or anything.
# There is a multiple things, try to perform on top of each & everything. Let's try to see.
# What kinds of operation, I will be able to perform  with the help of same thing.
```

In [83]:
```
# Below code Explanation :-

# l = [] - here may be, I can create a blank "List".

# for i in t :
#    l.append(i)

# If I have to convert everything into a "List" withour using "List" fucntion.

# L  outcome is ->   [3, 4, 4, 5, 5, 6, 67, 7, 7]

# Append what (i) [l.append(i)] then call "L".

# Now we can see as a outcome, we are able to convert these entire "Tuple" ->  t = (3,4,4,5,5,6,67,7,7) into a

# List  ->  [3, 4, 4, 5, 5, 6, 67, 7, 7] with the help of "for loop".
```

In [84]:
```
t = (3,4,4,5,5,6,67,7,7)
l = []
for i in t :
    l.append(i)
```

In [85]:
```
l
```

Out[85]:
```
[3, 4, 4, 5, 5, 6, 67, 7, 7]
```

In [86]:
```
# Continuee  code .....

# Now, What sir asked me to write a fucntion, I am able to do it.

# But there is another way to write a these entire steps into a single line code.

# [i for i in t] -> as a outcome we can see we got the same result --> [3, 4, 4, 5, 5, 6, 67, 7, 7]

# Both the code is going to give us same result. [i for i in t]. This is a comprehensive operation.

# Both operation is not different both are same. Both are giving me same outcome or result.

# [i for i in t] --> outcome ---> [3, 4, 4, 5, 5, 6, 67, 7, 7]

# Both are giving same outcome, only things we have done is that, instead of writting things in a multiple line
# I have written these entire things into a Single line. which is ->  [i for i in t]

# I was exepecting a "List", what I was try to iterate over to these (t) t = (3,4,4,5,5,6,67,7) & one by one
# I am trying return, these is something I will be able to get from these "Comprehensive Operation".
# [i for i in t].
```

In [87]:
```
[i for i in t]
```

Out[87]:
```
[3, 4, 4, 5, 5, 6, 67, 7, 7]
```

In [88]:
```
# Code No 23 :-

# Explanation below code :-

# v = "vijay"
# [i for i in v]

# Here we have converted entire string(str) into a "List". By using "List Comprehensive".

# [for i in v] return what "i".

# [i for i in v]  outcome ['v', 'i', 'j', 'a', 'y']

# For i in v return what ["i"]. It is [v= "vijay"] convert in a single line [i for i in v].

# for loop --> for i in "V"

# These is return statement --> ["i"].

# It will capture inside a "List" we are not suppose to called "Append operation".Because its a part of
# "List" itself.

# [i*i for i in range(10)] outcome is --> [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

In [89]:
```
# Code NO 23 :-

v = "vijay"
```

In [90]:
```
[i for i in v]
```

Out[90]:
```
['v', 'i', 'j', 'a', 'y']
```

In [91]:
```
# Code NO 24 :-

[i*i for i in range(10)]
```

Out[91]:
```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

In [92]:
```
# Code NO 25 :-

# Question :-

# Explanation below code :- Outcome for both Upper & below code is same. 2 different way to write code.


# Can you please, try to generate a basically square of a number starting from till 10.
# Square of each & every number. I am just looking for a "List" of square of all the number starting
# from 1 to 10.

# For "i" in range(10) then I can return (i*i). This is something I have to return.
# As a outcome, I am able to get  a square of a number of (0 to 10) basically, I have taken range of 0 to 10,
# which is always going to execlude the upper bound.

# l = []
# for i in range(10):
#    l.append(i*i)
# L = [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]

# Let's suppose,

# If I am not aware of "List Comprehensive operation", I can write the code this way. I will try to create
# a "Blank List".

# "L" If I will call L - "List".
# As a outcome, I can see same result. [0, 1, 4, 9, 16, 25, 36, 49, 64, 81].
# Both the result outcome is same there is no difference at all.
# Both way, I can able to perform all such kind of a operation.
```

In [93]:
```
l = []
```

In [94]:
```
for i in range(10):
    l.append(i*i)
```

In [95]:
```
l
```

Out[95]:
```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

In [96]:
```
# Code NO 26 :-

# Let's understand below code :

# l = Lambda, it will take some input as *Asterisk of X, means multiple input, there is chance that I would
# like to return a "List"

# "List Comprehensive" always return a "List", I can even write complete logic, that's completly fine.

# But,If we would like to use shortcut "List Comprehensive" is basically a shortcut for all of us, interms
# of doing coding

# [i for i in x] -> Convert entire things into a "List" & give it to me.

# l(4,45,5,6,6,7,8,9) -> I can try to call "L" & I can try to pass (4,45,5,6,6,7,8,9). As a outcome,
# we can see it is return me as a "List". ---> [4, 45, 5, 6, 6, 7, 8, 9].

# I have just pass a input (4,45,5,6,6,7,8,9) because, I was expecting a "List" [i for i in x] element.
# So, my outcome got as "List" --> [4, 45, 5, 6, 6, 7, 8, 9].
```

In [97]:
```
l = lambda *x : [i for i in x]
```

In [98]:
```
l(4,45,5,6,6,7,8,9)
```

Out[98]:
```
[4, 45, 5, 6, 6, 7, 8, 9]
```

In [99]:
```
# Code NO 27 :-

# Explanation below code :-

# Let's suppose,

# I am looking for a square of a each & every element.whatever, element I am going to pass I am looking for
# square of it.
#As a outcome we can see over here --> [16, 2025, 25, 36, 36, 49, 64, 81] = 4*4 = 16, 45*45 = 2025

# So, this is "Lambda" function where I am trying [i**2 for i in x] to "list Comprehensive operation"
# based on that,I am trying to return something.

# (**) double Asterisk means, square basically.
```

In [100…
```
l = lambda * x : [i**2 for i in x]
```

In [101…
```
l(4,45,5,6,6,7,8,9)
```

Out[101]:
```
[16, 2025, 25, 36, 36, 49, 64, 81]
```

In [102…
```
# Code NO 28 :-

# test14(a=7,b= "vijay", c = 345, d = "vijay", l = [2,3,4,45,"vijay"])

# let's suppose :-

# We are going to pass a multiple key value pair as a parameter, inside a function & let's suppose there is a
# requiredment to find out, how many number of string (str) we have passed as a parameter.
# What I am suppose to do.

# Let's suppose, I have a fucntion as "Test14" inside
# key value pair (a=7,b= "vijay", c = 345, d = "vijay", l = [2,3,4,45,"vijay"])

# If I have to count, how many times string(str), I have passed, if this is requiredment.

# If this type of argument, I have to pass. How, I am able to create a fucntion, what kind of parameter,
# I am suppose to give inside this one.

# Let's understand :-

# As we can see over here its a key value kind pair of input we are trying to pass.

# Now, I can try to call - def test14(** kwargs) : I can try to give what kind of argument double Asterisk(**).

# If we are going to pass Key value kind of pair in that case double Asterisk(**) is something,
# which we are suppose to pass.

# Double Asterisk(**) means keyword argument kind of things, I will be able to pass.
```

In [ ]:
```
# Question  :-

# My requiredment is that find out a number of a string(str), which I have passed.

# If we are going to pass key value kind of a pair, in that case double Asterisk (**) something that
# we are suppose to pass.

# Ok fine double Asterisk(**) keyword argument things, I will be able to pass.So, double Asterisk(**) keyword
# or any name
# I can give instead of "Kwargs" Then my requiredment is that, try to find out a  number of string(str).

# Which I have passed,

# Test14(a=7,b="vijay",c=345,d="vijay",l=[1,2,3,4,5,"vijay"])
# So, basically 3 string(str) "vijay"

# Here, I just have to count number of string(str) that it.
# So, my outcome or output would be [3 ] because, I have 3 string(str) inside my input.
```

In [112…
```
# outcome of above code.

def test14(**kwargs):
    count = 0
    for v in kwargs.values():
        if type(v) == str or type(v) == list:
            count += 1
    return count
```

In [113…
```
test14(a=7,b="vijay",c=345,d="vijay",l=[1,2,3,4,5,"vijay"])
```

Out[113]:
```
3
```

In [114…
```
# Code NO 29 :-

# What type of input, it is going to take, its a dictionary(dict) object. For verification, we can try to
# "Print(kwargs)".

# As a outcome 3 also printed but it should not be print.
```

In [115…
```
def test14(**kwargs):
    count = 0
    print(kwargs)
    for v in kwargs.values():
        if type(v) == str or type(v) == list:
            count += 1
    return count
```

In [116…
```
test14(a=7,b="vijay",c=345,d="vijay",l=[1,2,3,4,5,"vijay"])
```
```
{'a': 7, 'b': 'vijay', 'c': 345, 'd': 'vijay', 'l': [1, 2, 3, 4, 5, 'vijay']}
```
Out[116]:
```
3
```

In [ ]:
```
# Code NO 30 :-

# Let's Suppose :-

# We all know, if there is a dictionary, from dictionary if we have to count number of string(str) which is
# available inside "List or outside of it". How I will be count or pass those things.

# print(kwargs.values()) Once we execute it by adding this line of code as a outcome we can see below "kwargs.
# values" it is going to return as a outcome.
```

In [128…
```
def test14(**kwargs):
    count = 0
    print(kwargs)
    print(kwargs.values())
    for v in kwargs.values():
        if type(v) == str :
            count += 1
        if type(v) == list :
            for i in v :
                if type(i) == str :
                    count += 1
    return count
```

In [129…
```
test14(a=7,b="vijay",c=345,d="vijay",l=[1,2,3,4,5,"vijay"])
```
```
{'a': 7, 'b': 'vijay', 'c': 345, 'd': 'vijay', 'l': [1, 2, 3, 4, 5, 'vijay']}
dict_values([7, 'vijay', 345, 'vijay', [1, 2, 3, 4, 5, 'vijay']])
```
Out[129]:
```
3
```

In [130…
```
# Code NO 31 :-

# Question :-

# Can you please try to return whatever data which, I am going to pass as a keyword argument. Can you please
# try to return a "List" of all the data.

# I am just looking for a "List" of data as a return.

# Let's suppose,

# I am going to create a def test15 over here. -> (**kwargs) maybe, I can try to return List(kwargs.values())

# test15 & try to use same data inside test15 excute, As a outcome or result, we can see.

# [7, 'vijay', 345, 'vijay', [1, 2, 3, 4, 5, 'vijay']] - There is a "List" which I am able to return.
```

In [131…
```
def test15(**kwargs):
    return list (kwargs.values())
```

In [132…
```
test15(a=7,b="vijay",c=345,d="vijay",l=[1,2,3,4,5,"vijay"])
```
Out[132]:
```
[7, 'vijay', 345, 'vijay', [1, 2, 3, 4, 5, 'vijay']]
```

In [ ]:

In [ ]:

In [ ]:

Question or Task to solve for "Function Class"

In [134…
```
#                                    Questions :-

# 1.Try to Print a prime number in between 1 to 1000.

# 2.Try to write a fucntion which is equivalent to print function in "Python".

# 3.Try to write a function which is a replica of List append, extend and Pop function.

# 4.Try to write a Lambda function which can return a concatination of all the string that we will pass.

# 5.Try to write a Lambda function which can return "List" of square of all the data between 1-100.

# 6.Try to write a 10 different different example of Lambda function with a choice of your tasks.

# 7.Try to write a function which can perform a read operation from, txt file.
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js