Data Science & AI & NIC - Param

Python-For Data Science

Functions

Lecture No.- 02

Recap of Previous Lecture

Topic Functions Part-01

$$\left\{ \begin{array}{l} \text{define} \\ \text{built - In} \\ \text{user defined} \end{array} \right\}$$

Topics to be Covered

Topic Functions Part-02

✓ Arguments of function in Python

Topic : Functions

Q] Can we return more than 1 value by a function.

```
def f(a, b):

    return a+b

f(10,20)  # call

f(10)  X   KeyError
f( )  X
```

Slide 4

docstring $\Longrightarrow$

```
def f(a,b):
'''This function returns sum and difference'''
    return a+b, a-b

          30, -10
x, y = f(10,20)
print(x,y)
```

f.__doc__

Call

$$f(\underbrace{10, 20}_{\text{Positional argument}}, \underbrace{c=30, d=40}_{\text{Keyword argument}})$$

Keyword argument → Order matter नहीं करता

Call

$$f(10, 20, \underbrace{d=40, c=30})$$

default argument

```
def sum(a, b):
    return a+b
```

sum(10,20) ✓

sum(10,20,30) ✗ Error

$a, b = \underbrace{10, 20, 30}$

variable no. of arguments

f(1, 2, 3, 4, 5, 6, 7)

f(1, 2, 3, 4, 5)

f( )

f(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

$\boxed{\text{seq, collection}}$

$\overrightarrow{}$

f (1, 2, 3, 4)

```
def f(*n) :            → var. no. of argument

    sum = 0

    for ele in n :

        sum = sum + ele

    return sum
```

def (*n, x) :

f ("A", 10, 20, 12.34)

X

f ("A", 10, 20, x=12.34)

must be keyword argument

var. no. of arguments

print(x, y, sep='@', end='#')

print(x, y, z, end='#', sep='1')

var length Keyword argument $\leftarrow$

```
def f(**kwargs):

    for k,v in kwargs:

        print('key is',k,'value is',v)
```

#call

$\underbrace{f(a=10, c=30, d='A', b='Pankaj')}_{\text{Keyword argument}}$

kwargs {a : 10, c : 30, d : 'A', b : 'Pankaj'} t.me/PWpankajsirP

Positional arg

Keyword arg // call

default arg (a, b, c = 0)

var. length argument *arg

var. length keyword arg **kwarg

Day 18 different types of arguments in functions

In [8]:
```
#can we return multiple values from a function in python
def f(a,b):
    ''' This function returns sum and difference'''
    return a+b,a-b
```

In [9]:
```
x,y=f(10,20)
```

In [10]:
```
x
```

Out[10]:
```
30
```

In [11]:
```
y
```

Out[11]:
```
-10
```

In [12]:
```
type(f(1,2))
```

Out[12]:
```
tuple
```

In [13]:
```
f.__doc__  #object.__doc__
```

Out[13]:
```
' This function returns sum and difference'
```

In [14]:
```
print(f.__doc__)
```

```
 This function returns sum and difference
```

In [15]:
```
#positinoal arguments
def f(a,b,c):
    return a+b-c
#whenever we call this function exactly 3 arguments are required
#and position also matters
f(10,20,30)
```

Out[15]:
```
0
```

In [16]:
```
f(20,10,30)
```

Out[16]:
```
0
```

In [17]:
```
f(30,10,20)
```

Out[17]:
```
20
```

In [18]:
```
#The no. of arguments and position must be matched.
#if no. of arguments is not same ===>Error
f(12)
```

Loading [MathJax]/extensions/Safe.js

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[18], line 3
      1 #The no. of arguments and position must be matched.
      2 #if no. of arguments is not same ===>Error
----> 3 f(12)

TypeError: f() missing 2 required positional arguments: 'b' and 'c'
```

In [19]:
```
#keyword arguments
#f(name,age,dob,marital_status,phone_number)

#call it===> any order is possible   now with keyword as argument name
#f(age=34,name="pankaj",phone_number=9627901***,marital_status='single')
```

In [20]:
```
def f(a,b,c,d):
    return a*b/c-d
f(10,20,30,40)
```

Out[20]:
```
-33.333333333333336
```

In [21]:
```
f(30,20,10,40)
```

Out[21]:
```
20.0
```

In [22]:
```
f(d=40,c=30,a=10,b=20)#keyword arguments
```

Out[22]:
```
-33.333333333333336
```

In [23]:
```
f(c=30,a=10,d=40,b=20)#keyword arguments concept
```

Out[23]:
```
-33.333333333333336
```

In [24]:
```
print("pankaj","sharma",end='#',sep='@')
```

```
pankaj@sharma#
```

In [25]:
```
print("pankaj","sharma",end='@',sep='#')
```

```
pankaj#sharma@
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [26]:
```
f(a=10,b=20,30,40)
```

```
  Cell In[26], line 1
    f(a=10,b=20,30,40)
                     ^
SyntaxError: positional argument follows keyword argument
```

Loading [MathJax]/extensions/Safe.js

In [27]:
```
f(10,30,b=20,d=40)#keyword arguments
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[27], line 1
----> 1 f(10,30,b=20,d=40)

TypeError: f() got multiple values for argument 'b'
```

In [28]:
```
f(10,30,c=20,d=40)
```

Out[28]:
```
-25.0
```

In [29]:
```
f(10,20,c=30,d=40)
```

Out[29]:
```
-33.333333333333336
```

In [30]:
```
f(10,20,d=40,c=30)# c and d main order matter ni but positional main==>yes
#first 2 arguments are positional
```

Out[30]:
```
-33.333333333333336
```

In [31]:
```
def sum(a,b):
    return a+b
sum(10,20)#will work fine
```

Out[31]:
```
30
```

In [32]:
```
sum(10,20,30)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[32], line 1
----> 1 sum(10,20,30)

TypeError: sum() takes 2 positional arguments but 3 were given
```

In [33]:
```
def sum(a,b,c):
    return a+b+c
sum(10,20,30)
```

Out[33]:
```
60
```

In [34]:
```
sum(10,20)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[34], line 1
----> 1 sum(10,20)

TypeError: sum() missing 1 required positional argument: 'c'
```

In [35]:
```
#default arguments
#c is taking a default value as 0
#if we dont pass c then only 0 is taken
Loading [MathJax]/extensions/Safe.js )):
    return a+b+c
sum(10,20)#a==>10,b==>20 we didnt passed c value ====>default c=0 10+20+0
```

Out[35]:
```
30
```

In [36]:
```
sum(10,20,30)# a==>10,b==>20,c==>30 default value is not needed now
```

Out[36]:
```
60
```

In [37]:
```
def f(a,b,c=0,d=0):
    return a+b+c+d
f(10,20)#default value of c and d are taken 10+20+0+0
```

Out[37]:
```
30
```

In [38]:
```
f(10,20,30)#default value of c is not needed but default value of d is taken
# 10+20+30+0
```

Out[38]:
```
60
```

In [40]:
```
f(10,20,30,40)#default value of c and d are not needed
```

Out[40]:
```
100
```

In [41]:
```
def f(a=0,b=0,c,d):
    return a+b+c+d
f(10,20,30) #default must be after positional arguments
```

```
  Cell In[41], line 1
    def f(a=0,b=0,c,d):
                  ^
SyntaxError: non-default argument follows default argument
```

In [43]:
```
#function with variable no. of arguments
def f(*n):
    sum=0
    for ele in n:
        sum=sum+ele
    return sum
```

In [44]:
```
f()
```

Out[44]:
```
0
```

In [45]:
```
f(10)
```

Out[45]:
```
10
```

In [46]:
```
f(10,20)
```

Out[46]:
```
30
```

In [47]:
```
f(10,20,30)
```

Loading [MathJax]/extensions/Safe.js

In [48]:
```
f(10,20,30,40)
```

Out[48]:
```
100
```

In [49]:
```
def f1(*n):
    for ele in n:
        print(ele)
f1()
```

In [50]:
```
f1(1,12,34,'Pankaj',45)
```

```
1
12
34
Pankaj
45
```

In [51]:
```
help(print)
```

```
Help on built-in function print in module builtins:

print(*args, sep=' ', end='\n', file=None, flush=False)
    Prints the values to a stream, or to sys.stdout by default.

    sep
      string inserted between values, default a space.
    end
      string appended after the last value, default a newline.
    file
      a file-like object (stream); defaults to the current sys.stdout.
    flush
      whether to forcibly flush the stream.
```

In [60]:
```
def f(**kwargs):
    for k,v in kwargs.items():
        print("key is",k,"value is",v)
#dict.items() ==>pairs key-value
#dict.keys()
#dict.values()
```

In [61]:
```
f(a=1,b='pankaj',c=12.34,d=[1,2,3,4])
```

```
key is a value is 1
key is b value is pankaj
key is c value is 12.34
key is d value is [1, 2, 3, 4]
```

In [62]:
```
f(b='pankaj',a='hello')
```

```
key is b value is pankaj
key is a value is hello
```

In [63]:
```
f({1:2,a:'pankaj'})
```

Loading [MathJax]/extensions/Safe.js

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[63], line 1
----> 1 f({1:2,a:'pankaj'})

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

```
In [ ]:
```

Loading [MathJax]/extensions/Safe.js

THANK - YOU