REC

Import()

The syntax for using a function defined in a module is,

module_name.function_name()

The module name and function name are separated by a dot.

Here we list some of the functions supported by *math* module.

```
>>> import math
>>> print(math.ceil(5.4))
6
>>> print(math.sqrt(4))
2.0
>>> print(math.pi)
3.141592653589793
```

REC

dir()

- The built-in function *dir()* returns a sorted list of comma separated strings containing the names of functions, classes and variables as defined in the module.

1. >>> dir(math)

['__doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos', 'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos', 'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp','expm1', 'fabs', 'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf', 'isclose', 'isfinite', 'isinf', 'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'log2', 'modf', 'nan', 'pi', 'pow', 'radians', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'tau', 'trunc']

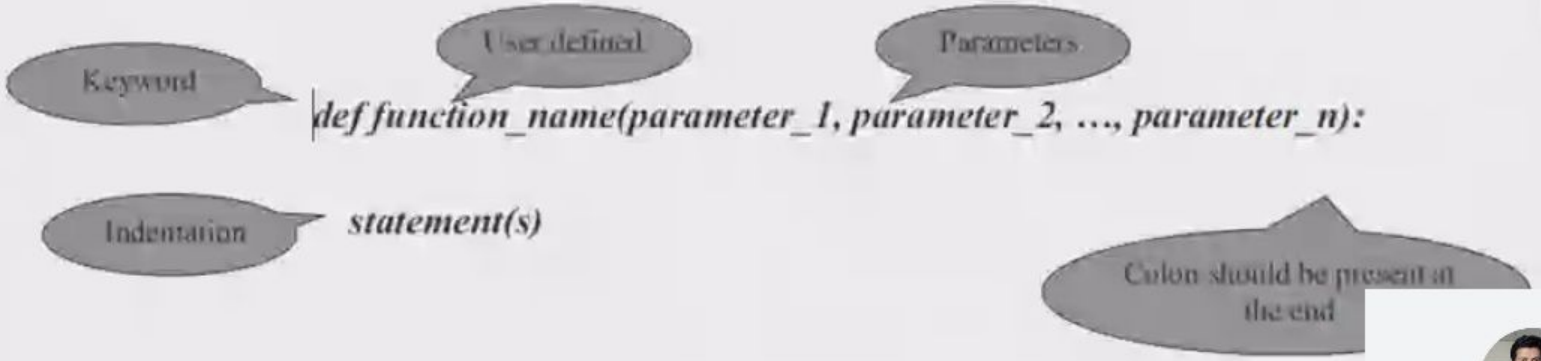
>>> print(math.sqrt(4))

You

REC

Function Definition and Calling the Function

- You can create your own functions and use them as and where it is needed. User-defined functions are reusable code blocks created by users to perform some specific task in the program.
- The syntax for function definition is,



You

REC

docstring

- The *def* keyword introduces a function definition. The term parameter or formal parameter is often used to refer to the variables as found in the function definition.
- The first statement among the block of statements within the function definition can optionally be a documentation string or docstring. There are tools which use docstrings to produce online documents or printed documentation automatically. Triple quotes are used to represent docstrings. For example,

""" This is single line docstring """

OR

""" This is

multiline

docstring """

You

REC

Calling a function

- Calling the function actually performs the specified actions with the indicated parameters.

The syntax for function call or calling function is,

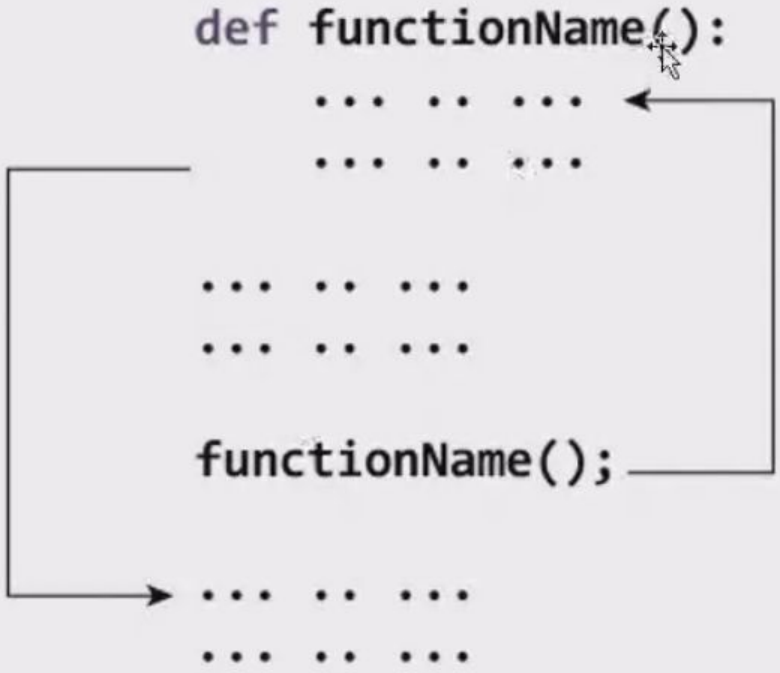
function_name(argument_1, argument_2,…,argument_n)

- Arguments are the actual value that is passed into the calling function. There must be a one to one correspondence between the formal parameters in the function definition and the actual arguments of the calling function.

You

REC

Click to



You

REC

Program to Find the Area of Trapezium Using the Formula Area = (1/2) * (a + b) * h Where a and b Are the 2 Bases of Trapezium and h Is the Height

```
def area_trapezium(a, b, h):
        area = 0.5 * (a + b) * h
        print(f"Area of a Trapezium is {area}")
def main():
        area_trapezium(10, 15, 20)
```

You

REC

program

```
colg_name = input('your college name ' )

def NameOfCollege(college_name):
    print(f"The college name is {colg_name}")

NameOfCollege(colg_name)
```

You

REC

Functions

Function Arguments

- You can call a function by using the following types of formal arguments –

- positional arguments
- Keyword arguments
- Default arguments
- Variable-length arguments

You

```
    print("Name and section=",name,sec)
    print("Info=",data)

    #for i,j in data.items():
       # print(i,j)

information('DSCE',"6B",PIN=560078, city="Benaluru", Phone=1234567890)
```

```
a= 6 b= 4 c= 2 d= 1 sum= 13 sub= 1 Mul= 24 div= 1.0
a= 5 b= 3 c= 9 d= 2 sum= 19 sub= 7 Mul= 15 div= 1.0
a= 2 b= 9 c= 5 d= 6 sum= 22 sub= -1 Mul= 18 div= 1.0
sum of var lenth= 6 21
Name and section= DSCE 6B
Info= {'PIN': 560078, 'city': 'Benaluru', 'Phone': 1234567890}
```

In [ ]:

```
def calculation(a,b,c,d):
    summ= a+b+c+d
    subb= c-d
    mult= a*b
    divi= a/a
    print("a=",a,"b=",b,"c=",c,"d=",d,"sum=",summ, "sub=",subb,"Mul=",mult,"div=",

#Positional arguments
calculation(6,4,2,1)
```

In [ ]:

```

```

You

REC

```
a= 6 b= 4 c= 2 d= 1 sum= 13 sub= 1 Mul= 24 div= 1.0
a= 5 b= 3 c= 9 d= 2 sum= 19 sub= 7 Mul= 15 div= 1.0
a= 2 b= 9 c= 5 d= 6 sum= 22 sub= -1 Mul= 18 div= 1.0
sum of var lenth= 6 21
Name and section= DSCE 6B
Info= {'PIN': 560078, 'city': 'Benaluru', 'Phone': 1234567890}
```

In [14]:

```
def calculation(a,b,c,d):
    summ= a+b+c+d
    subb= c-d
    mult= a*b
    divi= a/a
    return summ, subb, mult, divi
#Positional arguments
addition, subtraction, multiplication, division=calculation(6,4,2,1)
print( "addtion= ", addition, "sub=",subtraction,"Mul=",multiplication,"div=",division)
```

```
addtion=  13 sub= 1 Mul= 24 div= 1.0
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

You

REC

Python Jupyter/ | Functions - Jupyter Notebook

localhost:8888/notebooks/Python%20Jupyter/Functions.ipynb

jupyter Functions Last Checkpoint: Last Sunday at 11:55 PM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run Code

In [1]:

```
# Function-Arguments or parameters
def calculation(a,b,c=5,d=6):
    summ= a+b+c+d
    subb= c-d
    mult= a*b
    divi= a/a
    print("a=",a,"b=",b,"c=",c,"d=",d,"sum=",summ, "sub=",subb,"Mul=",mult,"div=",divi)

#Positional arguments
calculation(6,4,2,1)

#Keyword arguments
calculation(c=9,a=5,b=3,d=1)

#default arguments
calculation(2,9)

#*variable lenth arguments
def sumadd(*z):
    sum_var= 0
    for i in z:
        sum_var+=i
    print("sum of var lenth=",i,sum_var)

sumadd(1,2,3,4,5,6)

#**kwargs-Multiple keyword arguments
def information(name,sec,**data):
    print("Name and section=",name,sec)
    print("Info=",data)

    #for i,j in data.items():
      # print(i,j)

information('OSCE',"6B",PIN=560078, city="Benaluru", Phone=1234567890)
```

You

23-09-2020

Python Jupyter/ | Functions - Jupyter Notebook

localhost:8888/notebooks/Python%20Jupyter/Functions.ipynb

Brahma Kumaris | https://onlinecours... | Communications Sy... | Modeling the Prop... | YouTube | ecds_2017_21 - Go... | Knimbus - Your eLi... | International Journ... | Sci-Hub: removing...

jupyter **Functions** Last Checkpoint: Last Sunday at 11:55 PM (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run | Code

```
sum of var length= 6 21
Name and section= DSCE 6B
Info= {'PIN': 560078, 'city': 'Benaluru', 'Phone': 1234567890}
```

In [17]:

```
def calculation(a,b,c=6,d=9):
    summ= a+b+c+d
    subb= c-d
    mult= a*b
    divi= a/a
    return summ, subb, mult, divi

addition, subtraction, multiplication, division=calculation(2,4)
print( "addtion= ", addition, "sub=",subtraction,"Mul=",multiplication,"div=",division)
```

```
addtion=  21 sub= -3 Mul= 8 div= 1.0
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [6]:

```
def calculation(a,b,c=5,d=6):
    summ= a+b+c+d
    subb= c-d
    mult= a*b
    divi= a/a
    return summ, subb, mult, divi
    #print("a=",a,"b=",b,"c=",c,"d=",d,"sum=",summ, "sub=",subb,"Mul=",mult,"div=",divi)

#Positional arguments
```

You

23-09-2020

```
a= 6 b= 4 c= 2 d= 1 sum= 13 sub= 1 Mul= 24 div= 1.0
a= 5 b= 3 c= 9 d= 2 sum= 19 sub= 7 Mul= 15 div= 1.0
a= 2 b= 9 c= 5 d= 6 sum= 22 sub= -1 Mul= 18 div= 1.0
sum of var lenth= 6 21
Name and section= DSCE 6B
Info= {'PIN': 560078, 'city': 'Benaluru', 'Phone': 1234567890}
```

```python
def information(name,sec,**data):
    print("Name and section=",name,sec)
    print("Info=",data)

    #for i,j in data.items():
       # print(i,j)

information('DSCE',"5B",PIN=560078, city="Benaluru", Phone=1234567890)
```

```
Name and section= DSCE 5B
Info= {'PIN': 560078, 'city': 'Benaluru', 'Phone': 1234567890}
```

```python
#*variable lenth arguments
def sumadd(*z):
    sum_var= 0
    for i in z:
        sum_var+=i
    print("sum of var lenth=",i,sum_var)

sumadd(1,2,3,4,5,6,7,8,9)
```

```
sum of var lenth= 9 45
```

You