FUNCTIONS: Functions provide organized, reusable and modular code to perform a set of specific actions. They simplify the coding process, prevent redundant logic and make the code easier to flow. This describe the declaration and utilization of functio in python.

Python has many built-in functions like print(), input(), len().

Besides built-ins you can also create your own functions to do more specific jobs & these are called user-defined functions.

'def' is most common way to define a function. This is also called as single clause compound statement.

SYNTAX : def function_name(parameters):
statement(s)

*I/p - def my_function():
Print("Hello")
my_function()

O/p - Hello.

CALLING A FUNCTION:

To call a function, use the function name followed by paranthesis.

*I/p - print(myfunction.__doc__)

O/p - None.

*I/p - def myFirstfunction():
"""This is my first function. It will Print Hello world always."""
Print('Hello world')
myFirstfunction()

document string [doc-str]

O/p - Hello world.

*I/P - print(myFirstfunction.__doc__)

O/P - this is my first function. It will prin
Hello world always.

*I/P - print(divmod(56,32))

O/P - (1,24)

*I/P - print(divmod.__doc__)

O/P - Return the tuple (x//y, x%.y).
Invariant : div*y+mod == x.

x//y - int quotie
x%.y - remainde

*I/P - def square(x):
Print(x**2)
square(5)

O/P - 25

*I/P - help(square)

O/P - Help on function square in module__
main__:
square(x)

*I/P - help(divmod)

O/P - Help on built-in function divmod in module builtins:

divmod(x, y, /)

Return the tuple (x//y, x%y).

Invariant : div*y + mod == x.

FUNCTIONS WITH ARGUMENTS / PARAMETERS: (args)

→ A parameter is the variable listed inside the paranthesis in the function defination.

→ An argument is the value that is sent to the function when it is called.

Both of these (parameter & argument) can be used for same thing : information that are passed into a function.

*I/P - def sayHello(name):

'''This function wishes Hello.'''

Print('Hello', name)

sayHello(PYTHON)

O/P - Hello PYTHON

*I/p - def square(num):

```
    '''
    this function prints square'''
    Print('the square of {} is {}'.
                format(num, nu
```

square(10)

square(5)

O/p - the square of 10 is 100

the square of 5 is 25.

RETURN STATEMENT:

```
def dm(a, b):
    c = a // b
    d = a % b
    return c,d
```

→ SYNTAX

*I/p - def square(num):

```
    '''
    this function returns square'''
    return num* num

sq = square(6)
Print(sq)
```

O/p - 36.

*I/p - def fun():
    Print("Hello")
    Print(fun())

O/p - Hello
None.

*I/p - def add_sub(a,b):
    add = a+b
    sub = a-b
    return add, sub

addition = add_sub(10, 8)
Print(type(addition))

O/p - <class 'tuple'>

MODULE: A file containing a set of functions you want to include in your application.

SYNTAX: module_name.function_name.

Modules can also contain variables of all types like arrays, dictionaries, objects etc.

* I/p - import math
  Print (math.pi) O/p - 3.14159....
  Print (math.e) 2.71828....

↳* dir(math) - Defines all the functions and variables names in math.

* I/p - from math import factorial
  Print (factorial(5))

  O/p - 120

* I/p - from math import factorial as f
  Print (f(5))

  O/p - 120

* I/p - from math import *
  Print(pi)
  Print (factorial(5))

  O/p - 3.14159
  120

*I/P -
```
num = 7
if num > 1:
    for i in range(2,num):
        if (num % i) == 0:
            Print("True")
            Print(i, "times", num//i, "is",
                  num)
            break
    else:
        Print("True")
else:
    Print("False")
```

O/P - True.

*I/P -
```
num = 27
if num > 1:
    for i in range(2,num):
        if(num % i)==0:
            Print("True")
            Print(i,"times",num//i,"is",num)
            break
    else:
        Print("True")
else:
    Print("False")
```

O/P - True
3 times 9 is 27

*i/p - num = 5

```
if num>1:
    for i in range(2,num):
        if (num%i)==0:
            Print(num,'is not a prime nu
            Print(i,"times",num//i,"is",num)
            break
        else:
            Print(num,'is a prime numbe
    else:
        Print(num,"is not a prime number")
```

o/p - 5 is a prime number.