***User Defined Functions***

***Block of statements***

A block is one or more lines of code, grouped together so that they are treated as one big sequence of statements while executing. In Python, statements in a block are written with indentation*.*

**the Syntax of function is:**

***def*** NAME ([PARAMETER1, PARAMETER2, …..]):

The first line of function definition, i.e., Line No. 1 is called **heade**r and the a=radius\*\*2

Return a

The last statement of the function, i.e. return statement returns a value from the function. Return statement may contain a constant/literal, variable, expression or function, if return is used without anything, it will return **none**.

Those function does not contain a return statement, such functions are called **void functions**. Void functions might display something on the screen or have some other effect, but they don’t have a return value. if we are return the value, then we get a special value called none.

***Void function***

The function that perform some action or do some work but donot return any computed value or final values to the caller are called **void function.**

**Example** def check (num):

if (num%2==0):

Print “True”

else:

Print “False”

**Docstring Conventions:**

1. The first line of a docstring starts with capital letter and ends with a period (.)
2. Second line is left blank (it visually separates summary from other description).
3. Other details of docstring start from 3rd line.

**Parameters and Arguments**

**Parameters** are the value(s) provided in the parenthesis when we write function header. These are the values required by function to work.

If there is more than one value required by the function to work on, then, all of them will be listed in parameter list separated by comma.

**Argument** is values being passed through a function-call statement are **called argument** (or **actual parameter** or **actual argument**).

**Exmple:**

Of argument in function call

>>> Area (5)

5 is an argument. An argument can be constant, variable, or expression.

**Default argument**: A parameter having default value in function header is known as a default parameter.

**Key ward argument** the named argument with assigned values being passed in the function call statement.

rest, i.e. Line No. 2 in our example, is known as **body.**

**Function Header**.

It begins with the keyword def and ends with colon and contains the function identification details**.**

**DocSrintg**

DocString is an important tool to document the program better, and makes it easier to understand.

**EXAMPLE**= def area (radius):

**Fruit function**: Function returning values are also known as fruit function.

***Scope of Variables***

Scope of variable refers to the part of the program, where it is visible, i.e., area where you can refer (use) it.

***Global scope:***

*A variable, with global scope can be used anywhere in the program. It can be created by defining a variable outside the scope of any function/block.*

*Global scope variable is a variable defined in “main” program .such variable are said to have global scope*

***Example:***

*x=50*

*def test ( ):*

*Print “Inside test x is”, x*

*Print “Value of x is”, x*

*On execution the above code will produce*

*Inside test x is 50*

*Value of x is 50*

**Local Scope**

*A variable with local scope can be accessed only within the function/block that it is created in.*

*A local variable is a variable defined within a function .such variable are said to have local scope.*

***Example :***

*X=50*

*def test ( ):*

*y = 20*

*print „Value of x is ‟, X, „; y is ‟ , y*

*print „Value of x is ‟, X, „ y is „ , y*

*On executing the code we will get*

*Value of x is 50; y is 20*

***The function greet () is used to print a message (string) given number of times. If the second argument value, is not specified, then parameter times work with the default value provided to it.***

*Example:*

*def greet (message, times=1):*

*print message \* times*

*>>> greet („Welcome‟)*

*>>> greet („Hello‟, 2)*

*Will result in:*

*Welcome*

*HelloHello*

**keyword arguments:-**

keyward argument are the named argument with assigned values being passed in the function

call statement.

**Advantages of writing functions with keyword arguments are:-**

**1-** Using the function is easier as we do not need to remember about the order of the arguments.

**2-** We can specify values of only those parameters to which we want to, as - other parameters have default argument values

***Note:***

1. Python does not allow you to call a function before the function is declared.
2. When you write the name of a function without parenthesis, it is interpreted as the reference, when you write the function name with parenthesis, the interpreter invoke the function (object).