Functions in Python:
A function is a block of organized, reusable code that is used to perform a single, related action. Functions provide better modularity for your application and a high degree of code reusing.

As you already know, Python gives you many built-in functions like print(), etc. but you can also create your own functions. These functions are called user-defined functions.

Defining a function:
You can define functions to provide the required functionality. Here are simple rules to define a function in Python. Function blocks begin with the keyword def followed by the function name and parentheses (( )).

Any input parameters or arguments should be placed within these parentheses. You can also define parameters inside these parentheses.

The first statement of a function can be an optional statement - the documentation string of the function or docstring.The code block within every function starts with a colon (:) and is indented.The statement return [expression] exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as return None.

Calling a function:
Defining a function only gives it a name, specifies the parameters that are to be included in the function and structures the blocks of code.Once the basic structure of a function is finalized, you can execute it by calling it from another function or directly from the Python prompt.

Returning a value from a function:
The statement return [expression] exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as return None.All the above examples are not returning any value.

You can return a value from a function as follows −

```
 # Function definition is here
def sum( arg1, arg2 ):
   # Add both the parameters and return them."
   total = arg1 + arg2
   print "Inside the function : ", total
   return total;

# Now you can call sum function
total = sum( 10, 20 );
print "Outside the function : ", total
```

Passing functional arguments to functions:

We can also pass function as an argument to other function example:

```
def add(a,b):
    return a + b
def square(c):
    return c * c

square(add(2,3))
```

Modules in Python:
A module allows you to logically organize your Python code. Grouping related code into a module
makes the code easier to understand and use. A module is a Python object with arbitrarily named attributes that you can bind and reference.Simply, a module is a file consisting of Python code. A module can define functions, classes and variables. A module can also include runnable code. You can use any Python source file as a module by executing an import statement in some other Python source file.

example:
```
import module_name
```
Here module_name is the name of the module which contains the code which you want to use.