Python functions

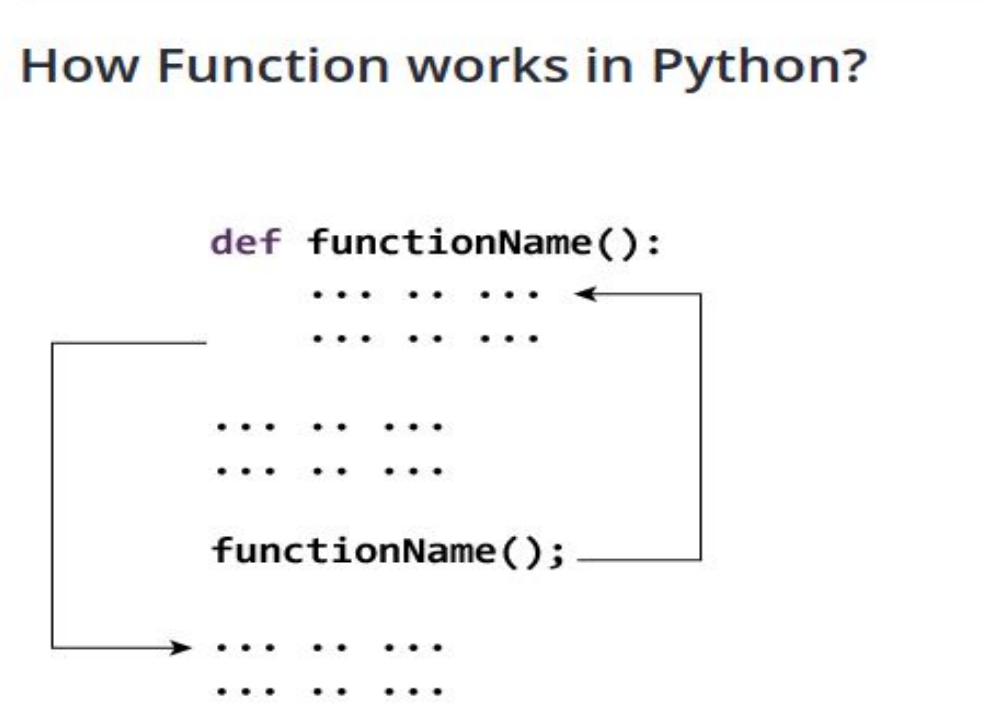
Besant Technologies

Functions:

- Group of statements to perform some task.
- For dividing program into small blocks.
- Easy to debug
- Reusability
- More organized and manageable.
- **Syntax of Function**

```
def function_name(parameters):
        """docstring"""
        statement(s)
```

Working:


How Function works in Python?
def functionName():
... .
... .
... .
... .
functionName();
... .
... .

Components:

- Function:
  - def keyword
  - Comments - doc string.
  - Statements
  - Arguments - optional
  - Return statement - optional

* Function should be called from main program to use that block.

* Scope of variables

Types:

- Based on argument and return:
  - Without arguments and return statement
  - Without arguments but with return statement
  - With arguments and without return statement
  - With both arguments and return statement

* Lambda/Anonymous functions

* Recursive functions

Argument types:

- Formal arguments
- Default arguments
- Keyword arguments
- Variable length arguments
- Variable length keyword arguments

Recursive functions:

- Function that calls itself.
- Ex: sum of n natural numbers, factorial etc.,

Lambda functions:

- Syntax: lambda arguments: expression
- Can be assigned to a variable.
- Simple programs with single line executions.

Special functions:

- Filter, map, reduce.

Modules:

- Module means a separate file.
- How to call functions from different files.
- Import <file>
- From <file> import <function>
- From <file> import *
- Import <file> as <different name>
- dir() - getting the functions in a file.
- __init__.py file
- __name__==__main__