GLS UNIVERSITY

Faculty of Computer Applications & Information Technology

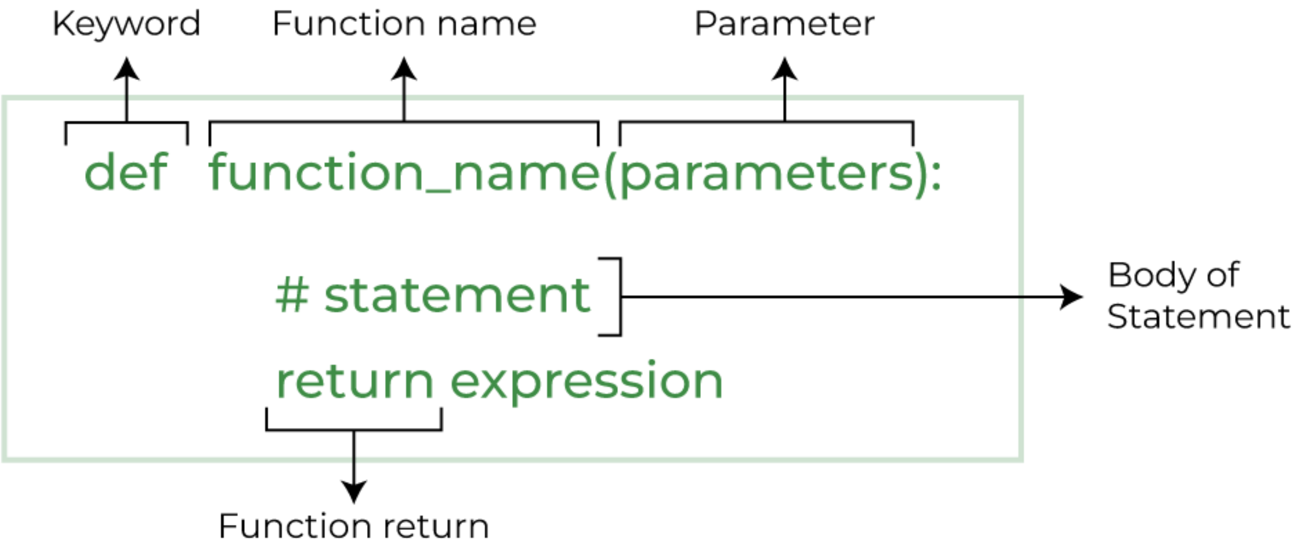
Integrated MCA Programme

Semester V

Unit 3: Python Functions, Modules and Packages

Functions

- A function is a block of code which only runs when it is called.
- You can pass data, known as parameters, into a function.
- A function can return data as a result.



Types of Functions in Python

- Default Argument
- Keyword Arguments (named arguments)
- Positional Arguments
- Arbitrary Arguments (variable-length arguments *args and **kwargs)
- Anonymous Functions (lambda)
- Recursive Functions
- Pass by Reference and Pass by Value

Python Modules

- A Python module is a file containing Python definitions and statements.
- A module can define functions, classes, and variables.
- A module can also include runnable code.
- Grouping related code into a module makes the code easier to understand and use. It also makes the code logically organized.
- To create a Python module, write the desired code and save that in a file with .py extension.
- The functions, and classes defined in a module to another module using the **import** statement in some other Python source file.
- When the interpreter encounters an import statement, it imports the module if the module is present in the search path.

Locating Python Modules

- Whenever a module is imported in Python the interpreter looks for several locations.
- First, it will check for the built-in module, if not found then it looks for a list of directories defined in the sys.path.
- Python interpreter searches for the module in the following manner –
  - First, it searches for the module in the current directory.
  - If the module isn’t found in the current directory, Python then searches each directory in the shell variable PYTHONPATH.
  - The PYTHONPATH is an environment variable, consisting of a list of directories.
- If that also fails python checks the installation-dependent list of directories configured at the time Python is installed.

Locating Python Modules

- Whenever a module is imported in Python the interpreter looks for several locations.
- First, it will check for the built-in module, if not found then it looks for a list of directories defined in the sys.path.
- Python interpreter searches for the module in the following manner –
  - First, it searches for the module in the current directory.
  - If the module isn't found in the current directory, Python then searches each directory in the shell variable PYTHONPATH.
  - The PYTHONPATH is an environment variable, consisting of a list of directories.
- If that also fails python checks the installation-dependent list of directories configured at the time Python is installed.

Packages in Python

- Python Packages are a way to organize and structure your Python code into reusable components.
- Package is like a folder that contains related Python files (modules) that work together to provide certain functionality.
- Packages help keep your code organized, make it easier to manage and maintain, and allow you to share your code with others.

Creating Packages in Python

- **Create a Directory:** Start by creating a directory (folder) for your package. This directory will serve as the root of your package structure.
- **Add Modules:** Within the package directory, you can add Python files (modules) containing your code. Each module should represent a distinct functionality or component of your package.
- **Init File:** Include an __init__.py file in the package directory. This file can be empty or can contain an initialization code for your package. It signals to Python that the directory should be treated as a package.
- **Subpackages:** You can create sub-packages within your package by adding additional directories containing modules, along with their own __init__.py files.

Creating Packages in Python

- **Importing:** To use modules from your package, import them into your Python scripts using dot notation. For example, if you have a module named module1.py inside a package named mypackage, you would import its function like this: from mypackage.module1 import greet.
- **Distribution**: If you want to distribute your package for others to use, you can create a setup.py file using Python's setuptools library. This file defines metadata about your package and specifies how it should be installed