

PYTHON LIBRARIES

VIVA QUESTIONS

1. Why do we organize our code in the form of functions, modules and libraries?

To easily access it and save ourselves from rewriting the same code again and again. It also makes our code compact and easy to debug.

2. What is a relationship between module, package and library?

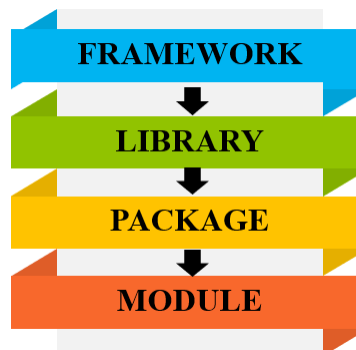
Module is a file which contains python functions, global variables etc. It is nothing but .py file which has python executable code / statement.

Package is a collection of modules. Each package in Python is a directory which **MUST** contain a special file called `__init__.py`. This file can be empty, and it indicates that the directory it contains is a Python package, so it can be imported the same way a module can be imported.

Library is a collection of packages. There is no difference between package and python library conceptually. Some examples of library in Python are:

- Python standard library containing math module, cmath module, random module, statistics module etc.
- NumPy library
- Matplotlib library
- Tkinter library

Framework is a collection of libraries. This is the architecture of the program.



3. How do we create modules in Python?

Modules in Python are simply Python files with a .py extension. The name of the module will be the name of the file. A Python module can have a set of functions, classes or variables defined and implemented.

4. What is the difference between docstrings and comments?

Docstrings are similar to commenting, but they are enhanced, more logical, and useful version of commenting. Docstrings act as documentation for the class, module, and packages.

On the other hand, Comments are mainly used to explain non-obvious portions of the code and can be useful for comments on Fixing bugs and tasks that are needed to be done.

Docstrings are represented with opening and closing quotes while comments start with a # at the beginning.

The comments cannot be accessed with the **help** function while docstring can be accessed with the **help** function.

5. What are different ways of importing modules in Python?

import <module_name> - This import the entire module and all functions inside that module can be used in our program.

import <module_name> as <alias_name> - An alias name can also be given to a module. This alias name is used to invoke the functions stored inside that module.

from <module_name> import <object_name> - This import only the specified functions/objects. No other function/object besides the imported one can be used in the program.

6. What is the importance of `__init__.py` file?

Each package in Python is a directory which MUST contain a special file called `__init__.py`. This file can be empty, and it indicates that the directory it contains is a Python package, so it can be imported the same way a module can be imported.