**Q1. Is it permissible to use several import statements to import the same module? What would the goal be? Can you think of a situation where it would be beneficial?**

Yes, you can import module as many times as you want in one Python program, no matter what module it is. Every subsequent import after the first accesses the cached module instead of re-evaluating it. It is used in case when we have to import multiple functions from same module.

**Q2. What are some of a module's characteristics? (Name at least one.)**

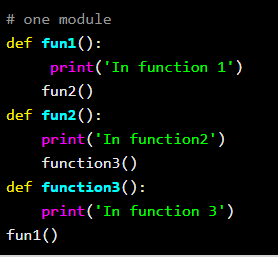
The following are some of a module's characteristics:

* **\_\_name\_\_** : It returns the name of the module
* **\_\_doc\_\_** : It denotes the documentation string line written in a module code.
* **\_\_file\_\_** : It holds the name and path of the module file from which it is loaded
* **\_\_dict\_\_** : It return a dictionary object of module attributes, functions and other definitions and their respective values

**Q3. Circular importing, such as when two modules import each other, can lead to dependencies and bugs that aren't visible. How can you go about creating a program that avoids mutual importing?**

Avoid importing objects or functions from a module that can cause Circular Imports. It is good to **import the whole module to avoid the Circular Import**.

When one module depends on another model and that module depends on first then it is good practice to **Merge both the modules** to avoid Circular Imports.



**Q4. Why is \_ \_all\_ \_ in Python?**

**\_\_all\_\_ it is the list of modules names** that should be imported when from package import \* is encountered.

### What is \_\_all\_\_

It's a list of public objects of that module, as interpreted by import \*. It overrides the default of hiding everything that begins with an underscore.

### When \_\_all\_\_ is used

It is a list of strings defining what symbols in a module will be exported when from <module> import \* is used on the module.

**Q5. In what situation is it useful to refer to the \_ \_name\_ \_ attribute or the string '\_ \_main\_ \_'?**

**“A \_\_name\_\_ is a built-in variable that returns us the name of the module being used.”**

In simple words, by **using \_\_name\_\_, we can check whether our module is being imported or run directly.**

Following are the advantages of using if \_\_name\_\_ == “\_\_main\_\_” statement:

* Using the main in our file, we can restrict some data from exporting to other files when imported.
* We can restrict the unnecessary data, thus making the output cleaner and more readable.
* We can choose what others may import or what they may not while using our module.

The value of the \_\_name\_\_ attribute **is set to \_\_main\_\_ when the module is run in the main program.** Otherwise, the value of \_\_name\_\_ is set to the name of the module if it is imported from another program.

