

Python Advance Assignment_24

Name : Bhavik Modi

E-mail : bmodi700@gmail.com

Website : <https://keenbm.github.io/>

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, it is permissible to use several import statements to import the same module. 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.)

Some Modules Characteristics are as follows

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?

Circular importing means importing the two modules in each other. If suppose we are working in MOD1.py file and it is importing some function say F2() from some other module say MOD2.PY file or we can do vice-versa. What will happen is: This will give an import error.

This is because when we import F2() function from module MOD2.py, then this will execute MOD2.py file. And in MOD2.py file there is another statement of importing MOD1.py module.

This will result in endless loop. To avoid this error just do one thing We can use if **name == 'main'**

In the function, you can't directly refer to the function in the program. The addition of this sentence avoids the endless loop of the program .

Q4. Why is `_all_` in Python?

It provides list of all modules present in a library.

Q5. In what situation is it useful to refer to the `_name attribute or the string 'main_'`?

During the time of execution of the code if we want to refer the module in which we are working on then we use name attribute. In that case it will return the module in which we are working on. Suppose if that module is being imported from some other module then name will have the name of that module from where the current module has been imported. The current module in which we are working is refer to the string **main** .

Q6. What are some of the benefits of attaching a program counter to the RPN interpreter application, which interprets an RPN script line by line?

Reverse Polish notation (also known as postfix notation) is one of multiple notations for representing mathematical expressions. Most of us are familiar with infix notation, but there are also the less popular prefix and postfix notations. The most obvious difference among these lie in the position of operators in the expression. Infix: operators are "in" between the operands Prefix: operators are before (pre) the operands Postfix: operators are after (post) the operands Other differences include the significance of space as a separator (prefix and postfix are space sensitive while infix is not) and the way order of operations is expressed. The order of operation is inherent in

the expression structure in prefix and postfix notations while the infix notation requires the use of parentheses to make the order of operations explicit.