Code example: https://github.com/geez0219/import\_trap

[Type of import]

* import <package/module> [as <variable>]
* from <package/module> import <subpackage/module/object> [as <variable>]

[relative import]

It has to be conducted by the form of:

>> from <several\_dot> <package/module> import <subpackage/module/object> [as <variable>]

* import a module in same directory

>> from . import <module>

* import a module in parent directory

>> from .. import <module>

(the grandparent directory will be …, and so on)

* import some module in subpackage directory

>> from .<subpackage> import <module>

>> from . import <subpackage>.<module>

(logic: input the subpackage that is in the same directory first, and then input the module inside it)

1. It need to have at less a ‘.’ before a module/package name, otherwise interpreter will take the import path as absolute import path
2. The way to execute python is strongly recommended using module way  
   (python -m <top-package>.<module>), Because only in this way, the envirment will set the <top-package> as top-package and gurantee the relative import work.  
     
   either [python -m main] and [python main.py] will not work because the current directory will not be treated as package

[absolute import]

>> import <top\_package>.<sub\_package>….<module>

1. Import everything from the top\_package level
2. The <top\_package> does not have dot prefix
3. The <top\_package> should be accessble from current directory

[frequent error]

* >>(X) import <package>.<subpackage>.<module>.<function/class>

The last thing of import stream cannot be the object in the module

>> (O) import <package>.<subpackage>.<module>

>> (O) from <package>.<subpackage>.<module> import <function/class>

* >>(X) import .<module>

The relative import can only be called by “from … import … ” syntax

>> (O) from . import <module>