python for Computational Problem Solving - pCPS - Functional_Programming_Testing Lecture Slides - Class #41_#42

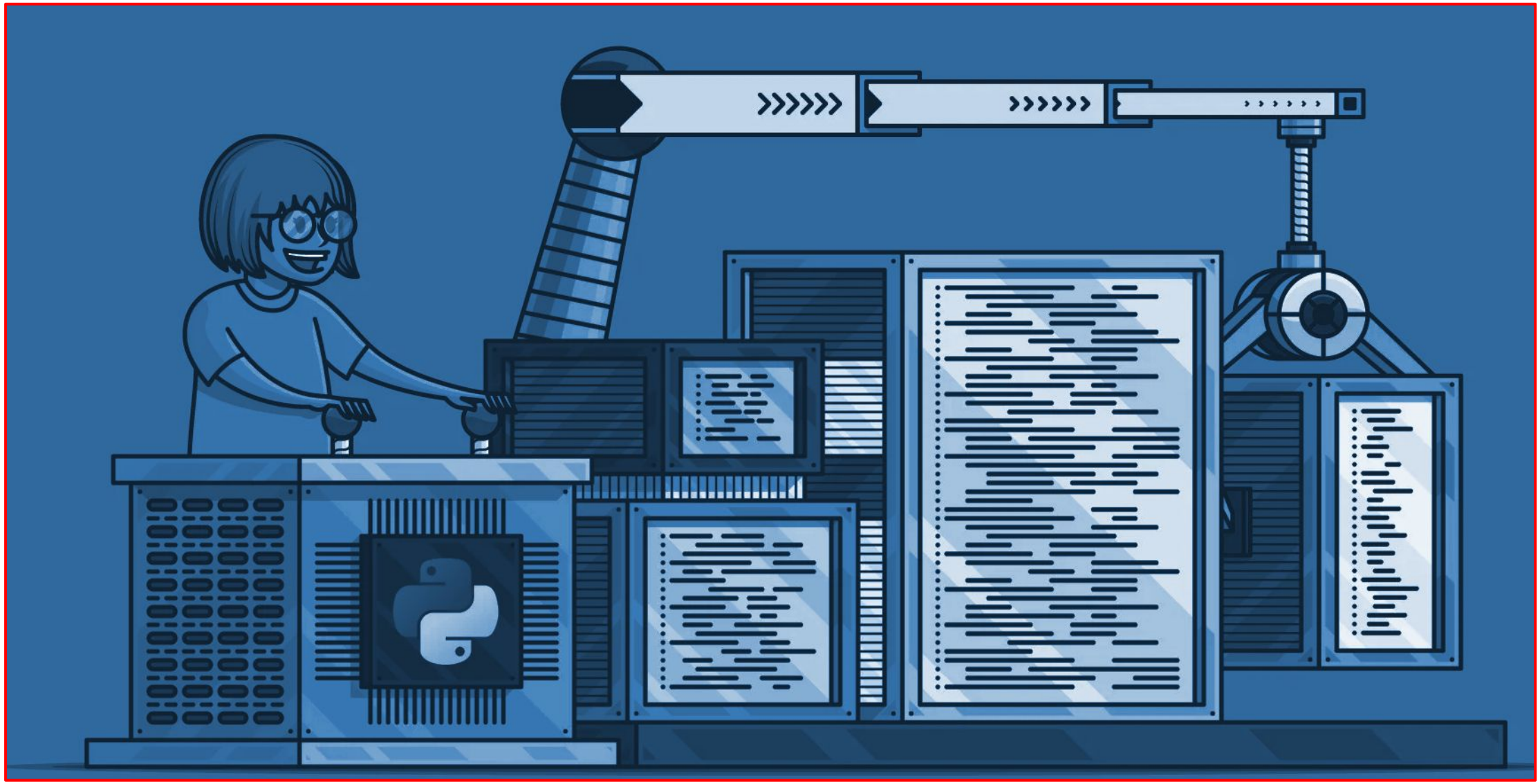
Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University

python for Computational Problem Solving Syllabus

Unit IV: Functional Programming, Modules,Testing and Debugging - 10 Hours

- Functional Programming - map, filter, reduce, max, min, lambda function
- list comprehension,
- Modules - import mechanisms
- Testing
  - Pytest , Function testing with Doctest
  - pdb debugger commands.

Functional Programming in python



Modules and Packages in python

- **Modular programming** refers to the process of breaking a large, unwieldy programming task into separate, smaller, more manageable subtasks or modules.
- **Individual modules** can then be **bonded** together like building blocks to create a larger application.
- **Advantages** to **modularizing** code in a **large** application
    - **Simplicity**: A module typically **focuses** on **one** relatively **small** portion of the **problem instead** of **focusing** on the **entire** problem.
    - **Maintainability**: Modules are typically **designed** so that they **enforce logical** boundaries between **different** problem **domains,** makes it more viable for a team of many programmers to work collaboratively on a large application.
    - **Reusability**: **Functionality** defined in a **single** module can be easily reused by other parts of the application, eliminating the need to duplicate code.
    - **Scoping**: **Modules** typically define a **separate** namespace, which helps **avoid collisions** between **identifiers** in different areas of a program.
- **Functions**, **modules** and **packages** are python **constructs** that **promote** code **modularization**.

Modules and Packages in python

- There are **three** different ways to define a **module** in python
  - Can be **written** in **python** itself.
  - Can be **written** in **C** and **loaded** dynamically at **run-time**
  - Can be a **built-in** module is **intrinsically contained** in the **interpreter**
- A **module's** contents are **accessed** in all three cases: with the **import** statement.
- **Modules written** in python are very **straightforward** to **build**.

```
University = 'PES University, Bengaluru-85'
Details = ['B.Tech First Semester','P Section',70,'Nitin V Pujari']

def MyDetails():
    print('MyClass--> ',Details[0])
    print('Section--> ',Details[1])
    print('Student Strength Approximately--> ',Details[2])
    print('MyName--> ',Details[3])
```

```
import MyModule

print(MyModule.University)
print(MyModule.Details)
print(MyModule.MyDetails)
print(MyModule.MyDetails())
```

```
PES University, Bengaluru-85
['B.Tech First Semester', 'P Section', 70, 'Nitin V Pujari']
<function MyDetails at 0x7f2ae3c43f70>
MyClass-->  B.Tech First Semester
Section-->  P Section
Student Strength Approximately-->  70
MyName-->  Nitin V Pujari
None
```

Modules and Packages in python - The Module Search Path

- import **MyModule**

- When the **interpreter executes** the **above** import statement, it searches for MyModule.py in a list of directories assembled from the following sources:

  - The **directory** from which the input **script** was **run** or the **current** directory if the interpreter is being run interactively

  - The list of directories contained in the **PYTHONPATH** environment variable, **if** it is **set**

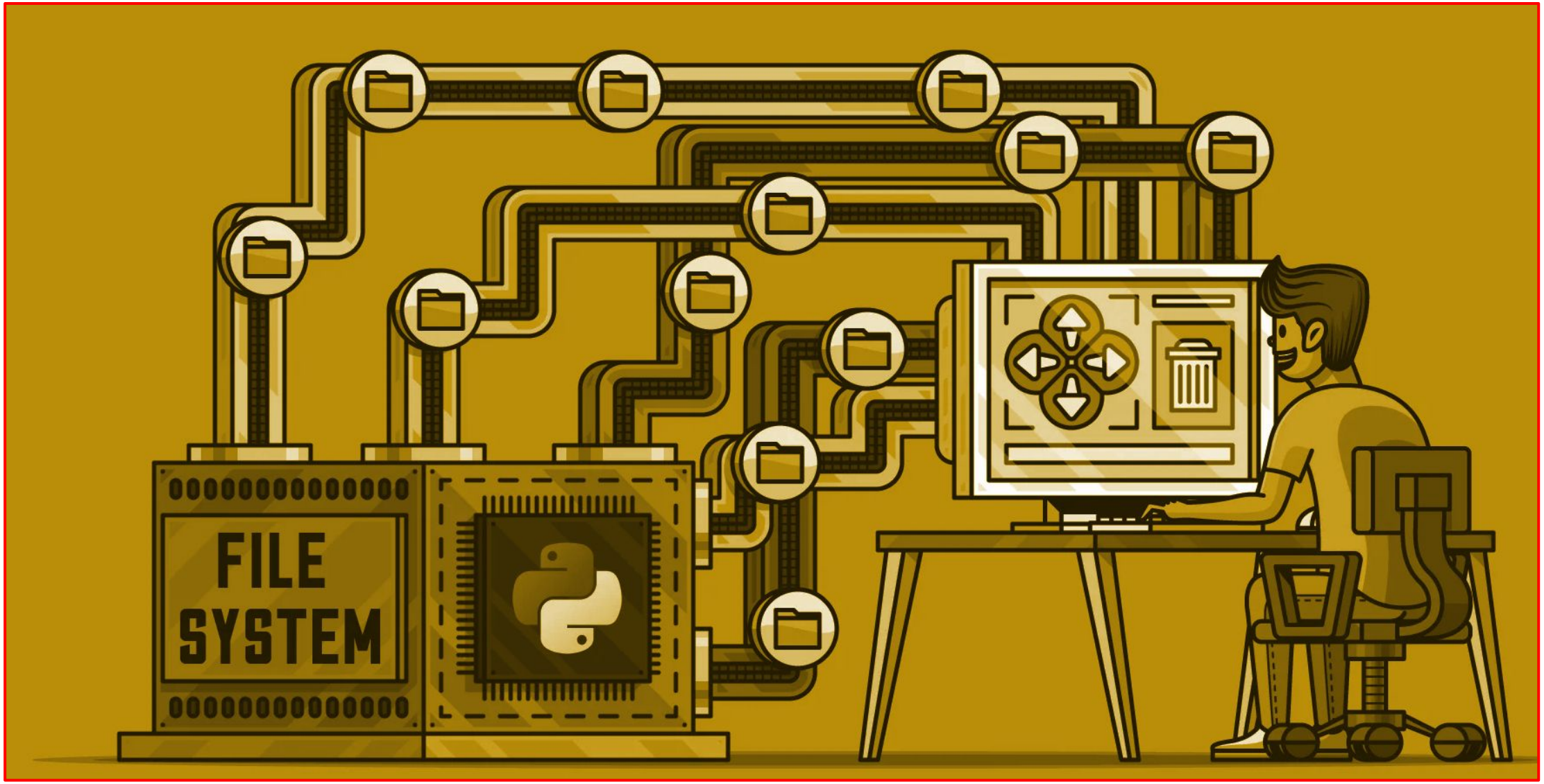
  - An **installation-dependent** list of directories **configured** at the time python **installation**

```
University = 'PES University, Bengaluru-85'
Details = ['B.Tech First Semester','P Section',70,'Nitin V Pujari']

def MyDetails():
    print('MyClass--> ',Details[0])
    print('Section--> ',Details[1])
    print('Student Strength Approximately--> ',Details[2])
    print('MyName--> ',Details[3])
```

```
import MyModule

print(MyModule.University)
print(MyModule.Details)
print(MyModule.MyDetails)
print(MyModule.MyDetails())
```

```
PES University, Bengaluru-85
['B.Tech First Semester', 'P Section', 70, 'Nitin V Pujari']
<function MyDetails at 0x7f2ae3c43f70>
MyClass-->  B.Tech First Semester
Section-->  P Section
Student Strength Approximately-->  70
MyName-->  Nitin V Pujari
None
```

Modules and Packages in python - The Module Search Path



Modules and Packages in python - The Module Search Path

```
import sys
print(sys.path)
# /home/datta/Downloads/MyModuleOne.py
print('----------------------------------------------------------------------------------------------------')
sys.path.append('/home/datta/Downloads')
print(sys.path)

import MyModule
import MyModuleOne

print(MyModule.University)
print(MyModule.Details)
print(MyModule.MyDetails)
print(MyModule.MyDetails())

print('-------------------------------------------From MyModuleOne-----------------------------------------------')
print(MyModuleOne.University)
print(MyModuleOne.Details)
print(MyModuleOne.MyDetails)
print(MyModuleOne.MyDetails())
```

```
['/home/datta', '/home/datta/anaconda3/lib/python38.zip', '/home/datta/anaconda3/lib/python3.8', '/home/datta/anacon
da3/lib/python3.8/lib-dynload', '', '/home/datta/anaconda3/lib/python3.8/site-packages', '/home/datta/anaconda3/lib/
python3.8/site-packages/locket-0.2.1-py3.8.egg', '/home/datta/anaconda3/lib/python3.8/site-packages/IPython/extensio
ns', '/home/datta/.ipython', '/home/datta/Downloads', '/home/datta/Downloads']
----------------------------------------------------------------------------------------------------
['/home/datta', '/home/datta/anaconda3/lib/python38.zip', '/home/datta/anaconda3/lib/python3.8', '/home/datta/anacon
da3/lib/python3.8/lib-dynload', '', '/home/datta/anaconda3/lib/python3.8/site-packages', '/home/datta/anaconda3/lib/
python3.8/site-packages/locket-0.2.1-py3.8.egg', '/home/datta/anaconda3/lib/python3.8/site-packages/IPython/extensio
ns', '/home/datta/.ipython', '/home/datta/Downloads', '/home/datta/Downloads', '/home/datta/Downloads']
PES University, Bengaluru-85
['B.Tech First Semester', 'P Section', 70, 'Nitin V Pujari']
<function MyDetails at 0x7f78522acf70>
MyClass-->  B.Tech First Semester
Section-->  P Section
Student Strength Approximately-->  70
MyName-->  Nitin V Pujari
None
-------------------------------------------From MyModuleOne-----------------------------------------------
PES University, Bengaluru-85
['B.Tech First Semester', 'P Section', 70, 'Nitin V Pujari']
<function MyDetails at 0x7f78522acb80>
MyClass-->  B.Tech First Semester
Section-->  P Section
Student Strength Approximately-->  70
MyName-->  Nitin V Pujari
None
```

Modules and Packages in python - The Module Search Path

```
import MyModule
import MyModuleOne
import math
print('Location of MyModule--> ',MyModule.__file__)
print('Location of MyModuleOne--> ',MyModuleOne.__file__)
print('Location of math Module--> ',math.__file__)

# A file with the . SO file extension is a Shared Library file.
# They contain information that can be used by one or more programs to offload resources
# so that the application(s) calling the SO file doesn't have to actually provide the file.
```

```
Location of MyModule-->  /home/datta/MyModule.py
Location of MyModuleOne-->  /home/datta/Downloads/MyModuleOne.py
Location of math Module-->  /home/datta/anaconda3/lib/python3.8/lib-dynload/math.cpython-38-x86_64-linux-gnu.so
```

Modules and Packages in python - The import Statement

- **Module** contents are made **available** to the **caller** with the **import** statement.

- The **import** statement takes **many** different **forms**
  - **import <module_name>**
  - The above statement does **not** make the module contents **directly** accessible to the **caller**.
  - Each **module** has its own **private symbol** table, which **serves** as the **global** symbol table for all **objects** defined in the module.
  - A **module** creates a **separate namespace**
  - The statement import <module_name> **only places** <module_name> in the **caller's symbol table**.
  - The **objects** that are defined in the **module** remain in the module's **private** symbol table.
  - From the caller, objects in the module are only accessible when prefixed with <module_name> via **dot notation**
- **Several comma-separated modules** may be specified in a **single import** statement

```
import MyModule
print(MyModule.Details)
print(Details)
```

```
['B.Tech First Semester', 'P Section', 70, 'Nitin V Pujari']

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_9680/2586708043.py in <module>
      1 import MyModule
      2 print(MyModule.Details)
----> 3 print(Details)

NameError: name 'Details' is not defined
```

```
import MyModule
import MyModuleOne
print(MyModule.Details)
print(MyModuleOne.MyDetails())
```

```
['B.Tech First Semester', 'P Section', 70, 'Nitin V Pujari']
MyClass-->  B.Tech First Semester
Section-->  P Section
Student Strength Approximately-->  70
MyName-->  Nitin V Pujari
None
```

Modules and Packages in python - The import Statement

- The **import** statement takes **many** different **forms**
  - from **<module_name> import <name(s)>**
  - An alternate form of the import statement allows individual objects from the module to be imported directly into the caller's symbol table
  - This form of import places the object names directly into the **caller's symbol table**, any objects that already exist with the **same name** will be **overwritten**
  - It is even possible to **indiscriminately** import everything from a module using ***
    - from **<module_name> import ***
  - This will **place** the **names** of all objects from <module_name> into the local symbol table, with the **exception** of any that begin with the **underscore (_) character**.

```
import sys
sys.path.append('/home/datta/Downloads')

from MyModule import University
from MyModule import MyDetails
from MyModuleOne import MyDetails

print(University)
print(MyDetails())
```

```
PES University, Bengaluru-85
I am from MyModuleOne
MyClass-->  B.Tech First Semester
Section-->  P Section
Student Strength Approximately-->  70
MyName-->  Nitin V Pujari
None
```

```
import sys
sys.path.append('/home/datta/Downloads')

from MyModule import University
from MyModuleOne import MyDetails
from MyModule import MyDetails


print(University)
print(MyDetails())
```

```
PES University, Bengaluru-85
I am from MyModule
MyClass-->  B.Tech First Semester
Section-->  P Section
Student Strength Approximately-->  70
MyName-->  Nitin V Pujari
None
```

Modules and Packages in python - The import Statement

- The **import** statement takes **many** different **forms**
  - It is also possible to import individual objects but enter them into the local symbol table with alternate names
  - This makes it possible to place names directly into the local symbol table but avoid conflicts with previously existing names
  - from <module_name> import <name> as <alt_name>[, <name> as <alt_name> …]
  - You can also import an entire module under an alternate name
  - **import <module_name> as <alt_name>**
  - Python 3 does not allow the indiscriminate import * syntax from within a function
  - a try statement with an except ImportError clause can be used to guard against unsuccessful import attempts

```
from MyModule import University as U
from MyModuleOne import MyDetails as MD1
from MyModule import MyDetails as MD2

print(U)
print(MD1())
print(MD2())
```

```
PES University, Bengaluru-85
I am from MyModule
MyClass-->  B.Tech First Semester
Section-->  P Section
Student Strength Approximately-->  70
MyName-->  Nitin V Pujari
None
I am from MyModule
MyClass-->  B.Tech First Semester
Section-->  P Section
Student Strength Approximately-->  70
MyName-->  Nitin V Pujari
None
```

```
import MyModuleOne as MD2
import MyModule as MD1

print(MD1.University)
print(MD1.MyDetails())
print(MD2.MyDetails())

try:
    import Nitin as Pujari
except:
    print('Module Not Found')
```

```
PES University, Bengaluru-85
I am from MyModule
MyClass-->  B.Tech First Semester
Section-->  P Section
Student Strength Approximately-->  70
MyName-->  Nitin V Pujari
None
I am from MyModuleOne
MyClass-->  B.Tech First Semester
Section-->  P Section
Student Strength Approximately-->  70
MyName-->  Nitin V Pujari
None
Module Not Found
```

Modules and Packages in python - The import Statement

- The **import** statement takes **many** different **forms**
  - When a **.py** file is imported as a **module**, python sets the special **dunder** variable **__name__** to the **name** of the **module**.
  - If a file is run as a standalone script, __name__ is set to the string '__main__'.
  - Using this fact, you can discern which is the case at run-time and alter behavior accordingly

```
University = 'PES University, Bengaluru-85'
Details = ['B.Tech First Semester','P Section',70,'Nitin V Pujari']

def MyDetails():
    print('I am from MyModule')
    print('MyClass--> ',Details[0])
    print('Section--> ',Details[1])
    print('Student Strength Approximately--> ',Details[2])
    print('MyName--> ',Details[3])

if __name__=='__main__':
    print(MyDetails())
```

```
(base) aspirations-2020@nitinpujari:~/Desktop/Class#41_#42$ python MyModule.py
I am from MyModule
MyClass-->  B.Tech First Semester
Section-->  P Section
Student Strength Approximately-->  70
MyName-->  Nitin V Pujari
None
(base) aspirations-2020@nitinpujari:~/Desktop/Class#41_#42$ python
Python 3.8.12 (default, Oct 12 2021, 13:49:34)
[GCC 7.5.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import MyModule as MD1
>>> MD1.MyDetails()
I am from MyModule
MyClass-->  B.Tech First Semester
Section-->  P Section
Student Strength Approximately-->  70
MyName-->  Nitin V Pujari
```

End of class #41, #42

Thank you

Nitin V Pujari
Faculty, Computer Science
Dean - IQAC, PES University
nitin.pujari@pes.edu

For Course Digital Deliverables visit www.pesuacademy.com