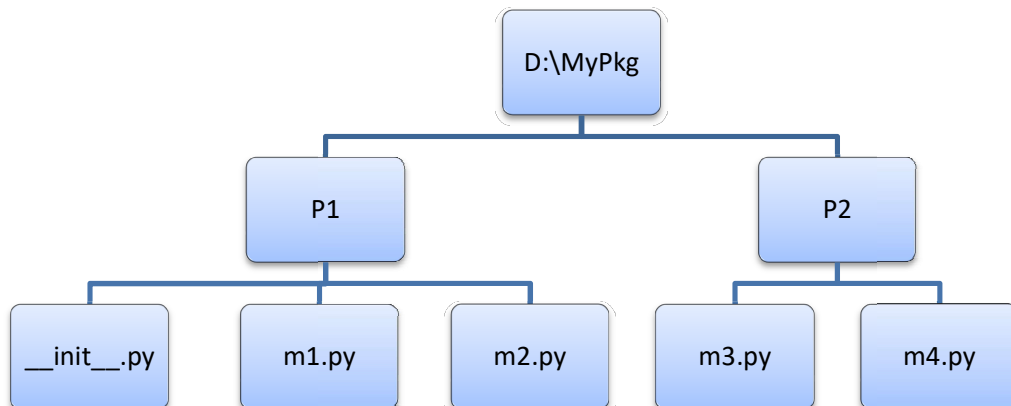


Programming and Problem Solving through Python Language O Level / A Level

Chapter - 8: Scope and Modules

Packages



Module m1.py

```
Employee={ "name":"Ajay M1", "age":27, "salary":30000}  
def show(name):  
    print("Name :", name)
```

Module m2.py

```
Employee={ "name":"Vijay M2", "age":27, "salary":30000}  
def sum(x,y):  
    return(x+y)
```

Module m3.py

```
def show(name,age):  
    print("Name :", name)  
    print("Age :", age)  
Employee={ "name":"Ajay M3", "age":27, "salary":30000}
```

Module m4.py

```
def mul(x,y):  
    print("sum : ", x*y)  
  
Employee={ "name":"Vijay M4", "age":27, "salary":30000}
```

Using the from ...import

- This is used to import the part of the Package as required.
- If the module from the package is imported using the “**from..import**” syntax, then module name is not used before the function or variable imported, as applicable in DOT Notation.
- For example if **show()** is imported from P1.m1 module, we use show() instead of P1.m1.show().

Example-1

```
import sys
sys.path.append("D:\\")

from MyPkg.P1.m1 import show
from MyPkg.P1.m2 import sum

print("Objects loaded in Memory")
print(dir())

print("Accessing P1 - m1 module ")
show("Ajay")

print("Accessing P1 - m2 module ")
print( "Sum :", sum(20,20))

print("\n Accessing Variable from Module\n")
print(Employee["name"])          # This is not accessible, as not imported.
```

Output

Objects loaded in Memory

```
['_annotations_', '__builtins__', '__doc__', '__file__', '__loader__', '__name__',
 '__package__', '__spec__', 'show', 'sum', 'sys']
```

Accessing P1 - m1 module

Name : Ajay

Accessing P1 - m2 module

Sum : 40

Accessing P1 - m1 module

Traceback (most recent call last):

File "D:\Python38\p3.py", line 17, in <module>

print(Employee["name"])

NameError: name 'Employee' is not defined

Example-2

```
import sys
sys.path.append("D:\\")

from MyPkg.P1.m1 import Employee
from MyPkg.P1.m2 import Employee

print("Objects loaded in Memory")
print(dir())

print("Accessing P1 - m1 module ")
print( Employee["name"])

print("Accessing P1 - m2 module ")
print( Employee["name"])
```

Output

```
Objects loaded in Memory
['Employee', '__annotations__', '__builtins__', '__doc__', '__file__', '__loader__',
 '__name__', '__package__', '__spec__', 'sys']

Accessing P1 - m1 module
Vijay M2

Accessing P1 - m2 module
Vijay M2
```

In this, **Employee** object is present in both the modules (**m1 and m2**), so the **Employee** object from the module-m2 replaces **Employee** from the module-m1. Thus, output is same in above case. To resolve this issue, follow the following example

Example-3

```
import sys
sys.path.append("D:\\")

from MyPkg.P1.m1 import Employee as E1
from MyPkg.P1.m2 import Employee as E2

print("Objects loaded in Memory")
print(dir( ))

print("Accessing P1 - m1 module ")
print( E1["name"])

print("Accessing P1 - m2 module ")
print( E2["name"])
```

Output

Objects loaded in Memory

```
['E1', 'E2', '__annotations__', '__builtins__', '__doc__', '__file__', '__loader__', '__name__', '__package__', '__spec__', 'sys']
```

Accessing P1 - m1 module

Ajay M1

Accessing P1 - m2 module

Vijay M2

Importing all the Modules using from...import *

- It can be used to import all the modules from the package.
- If the module is imported using the “**from...import**” syntax, then module name is not used before the function or variable imported.
- For module , import * allows importing all the objects defined in the module.
- For Package, import * does not import anything. It will allow importing, only if the `__init__.py` file is present in the folder and `__all__` is defined in it.

Content of `__init__.py` file present in the “D:\MyPkg\P1” folder

```
__all__=["m1","m2"]
```

Example-1

```
import sys
sys.path.append("D:\\")

from MyPkg.P1 import *

print("Objects loaded in Memory")
print(dir())

print("Accessing P1 - m1 module ")
print( m1.Employee["name"])

print("Accessing P1 - m2 module ")
print( m2.Employee["name"])
```

Output

Objects loaded in Memory

```
['__annotations__', '__builtins__', '__doc__', '__file__', '__loader__', '__name__', '__package__', '__spec__', 'm1', 'm2', 'sys']
```

Accessing P1 - m1 module

Ajay M1

Accessing P1 - m2 module

Vijay M2

Example-2

```
#__init__.py file not present in D:\MyPkg\P2 folder

import sys
sys.path.append("D:\\")

from MyPkg.P2 import *

print("Objects loaded in Memory")
print(dir())

print("Accessing P1 - m1 module ")
print( m3.Employee["name"])

print("Accessing P1 - m2 module ")
print( m4.Employee["name"])
```

Output

Objects loaded in Memory

```
['__annotations__', '__builtins__', '__doc__', '__file__', '__loader__', '__name__',
 '__package__', '__spec__', 'sys']
```

Accessing P1 - m1 module

Traceback (most recent call last):

File "D:/Python38/p4.py", line 12, in <module>

```
    print( m3.Employee["name"])
```

NameError: name 'm3' is not defined