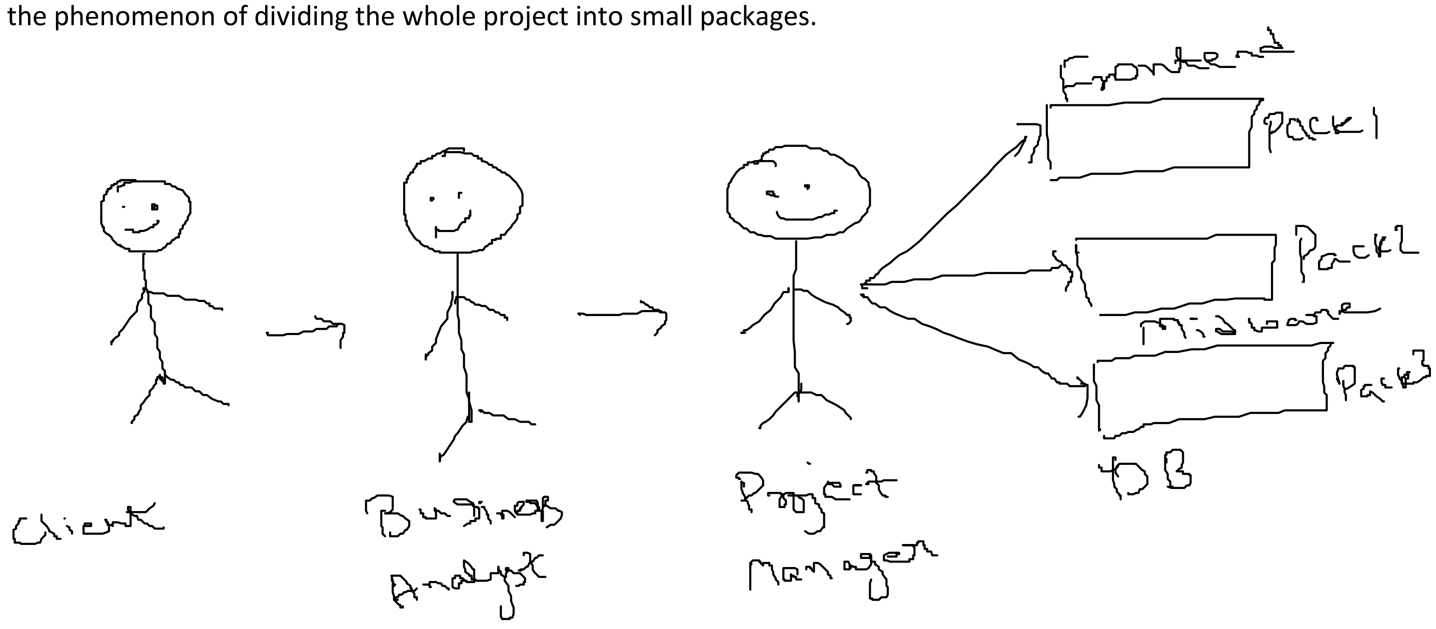
Package Architecture

Thursday, August 14, 2025 8:14 PM

It is the phenomenon of dividing the whole project into small packages.



Package is the folder that consists of python files.
Python files are known as modules and collection of modules is known as package.

Package	Modules
Folder containing modules	Python file with functions, class var etc.
For accessing, installation is required. pip install package_name Or Conda install package_name	For accessing, import is required. For importing: import module_name as var or from module_name import *

Types of modules:

i. In-built module
ii. User-defined module

i. Inbuilt modules:

Syntax to use modules:

1. Import mod_name
   Mod_name.fname()

2. From mod_name import *
   Fname()

3. From mod_name import fname()
   Fname()

Math Module:

```
import math
math.sqrt(25)
5.0
math.factorial(5)
120
math.pi
3.141592653589793
math.lcm(2,4,6)
12
math.pow(2,3)
8.0
dir(math)
['__doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos',
'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'cbrt', 'ceil', 'comb', 'copysign',
'cos', 'cosh', 'degrees', 'dist', 'e', 'erf', 'erfc', 'exp', 'exp2', 'expm1', 'fabs',
'factorial', 'floor', 'fma', 'fmod', 'frexp', 'fsum', 'gamma', 'gcd', 'hypot', 'inf',
'isclose', 'isfinite', 'isinf', 'isnan', 'isqrt', 'lcm', 'ldexp', 'lgamma', 'log', 'log10',
'log1p', 'log2', 'modf', 'nan', 'nextafter', 'perm', 'pi', 'pow', 'prod', 'radians',
'remainder', 'sin', 'sinh', 'sqrt', 'sumprod', 'tan', 'tanh', 'tau', 'trunc', 'ulp']
```

Random module:

```
from random import *
randint(1,5)
3
random()
0.4899740686166213
choice([10,20,30,40,50])
50
a=[10,20,30,40]
print(shuffle(a))
None
shuffle(a)
a
[20, 30, 40, 10]
dir(random)
['__call__', '__class__', '__delattr__', '__dir__', '__doc__', '__eq__',
'__format__', '__ge__', '__getattribute__', '__gt__', '__hash__', '__init__',
'__init_subclass__', '__le__', '__lt__', '__module__', '__name__', '__ne__',
'__new__', '__qualname__', '__reduce__', '__reduce_ex__', '__repr__',
'__self__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__',
'__text_signature__']
```

Calendar Module:

```
from calendar import *
calendar(2025)
dir(calendar)
```

['__call__', '__class__', '__delattr__', '__dir__', '__doc__', '__eq__',
'__format__', '__func__', '__ge__', '__get__', '__getattribute__', '__gt__',
'__hash__', '__init__', '__init_subclass__', '__le__', '__lt__', '__ne__',
'__new__', '__reduce__', '__reduce_ex__', '__repr__', '__self__', '__setattr__',
'__sizeof__', '__str__', '__subclasshook__']

ii. User-defined Modules: The python files with class, functions or variables are known as user-defined modules.

Packages

Packages are of 2 types:

i. Inbuilt package
ii. User-defined package

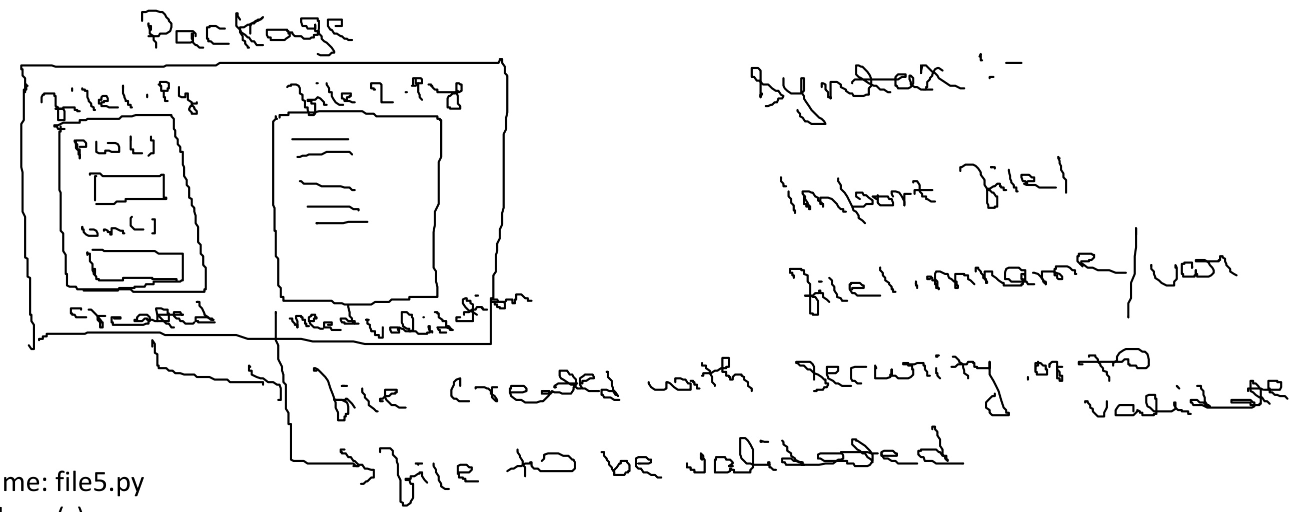
1. Inbuilt- package: Numpy, pandas, django, selenium etc.

   Pip install pack_name

   Or

   Conda install pack_name

2. User-defined package: The package/ folder that is created by the developer as per the user convenience.



Syntax :-

import file1

file1.name|var

Eg:

File name: file5.py

```
def val_pw(s):
  if len(s)>=8:
    u,l,d,sc=0,0,0,0
    for i in s:
      if 'A'<=i<='Z':
        u+=1
      elif 'a'<=i<='z':
        l+=1
      elif '0'<=i<='9':
        d+=1
      elif i in '@_$':
        sc+=1
    if u>=1 and l>=1 and d>=1 and sc>=1:
      return True
```

```
        return False
    return False
def val_un(s):
    if '_' in s:
        return True
    else:
        return False

if __name__=='__main__':
    print(val_pw('Apple@12'))
    print(val_un('Newton_Apple'))
```

File name: file6.py

```
import file5
un=input('Enter the Username: ')
pw=input('Enter the password: ')
print(file5.val_un(un))
print(file5.val_pw(pw))
```

- __main__ is used for the file which should not be executed in linked file.
- It will execute in the file where it is created only, not in other linked file.

If the file is in different package:
From pack_name import file_name

Or if we just want to use method, then:
From pack.file import method_name.

Eg:
Create a new folder and save file7.py

```
From pack_1 import file5
un=input('Enter the Username: ')
pw=input('Enter the password: ')
print(file5.val_un(un))
print(file5.val_pw(pw))
```