Monday, November 21, 2022 8:26 PM

Modules and Packages

Modules:- A group of functions, variables and classes saved a file, which is nothing but a module and it's in build.

ex:- math, os, sys, functools, unittest etc.

Every:- Python file show as a module.

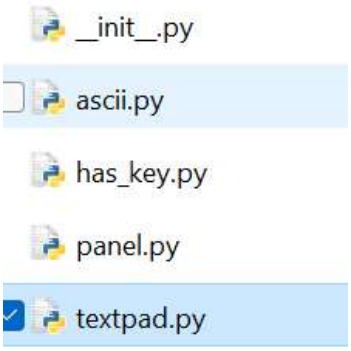
```
class Template:
    """A string class for supporting $-substitutio
    delimiter = '$'

    idpattern = r'(?a:[_a-z][_a-z0-9]*)'
    braceidpattern = None
    flags = _re.IGNORECASE

    def __init_subclass__(cls):
        super().__init_subclass__()
        if 'pattern' in cls.__dict__:
            pattern = cls.pattern
        else:
```

Packages :- It is an encapsulation mechanism to group related modules in a single unit, it's a package of module. It's look like a folder and it’s contain several python files(.py) and __init__.py(imp)

Ex:- pandas, numpy, pytest, tenserflow etc



Using **import** keyword we call python module

```
import math
import opcode
import os
import datetime
import sys
import zipfile
import json
import xml

print(dir(math))
```

```
>>> import math
>>> dir(math)
['__doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos', 'acosh', 'asin', 'asinh', 'atan',
'atan2', 'atanh', 'ceil', 'comb', 'copysign', 'cos', 'cosh', 'degrees', 'dist', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs',
'factorial', 'floor', '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', 'tan', 'tanh', 'tau', 'trunc', 'ulp']
>>> math.pi
3.141592653589793
>>> math.pow(3,2)
9.0
```

How to create user define module.:-

Create a python file, and create class, functions and variables

modules_1.py | modules_2.py

```
1
2   a = 70
3   def add(b,c):
4       add = b+c+a
5       return add
6
7   def product(b,c):
8       prod = b*c
9       return prod
10
```

Note:- If we want call this python file as a module, create another python file and using import keyword, import the module .

Ex:- 1st file :- modules_1.py
2nd file:- module_2.py

Import modules_1.py

modules_1.py modules_2.py

```
1  import modules_1
2  print(dir(modules_1))
3  print(modules_1.add(40,30))
4  print(modules_1.product(7,8))
```

Note:- Whenever we are using a module in our program . For that module complied file will be generated and stored in the disk permanetly .