Python for Data Science

CLASSES AND MODULES

Lecture No. 03

By- Kashif Sir

RECAP

→ Types of errors

→ Bytecode

→ Exception handling
  - → try
  - → except
  - → finally
  - → else

→ Comments

→ __name__

AGENDA

modules

→ 2 types → In build

→ User Defined

- Numpy
- collections
- math
- datetime
- sys
- os

Module

Any program in Python is called module.

There are two types of modules:

1) Built in module
   numpy, math, datetime, sys, os

2) User defined module
   The code that we write & save in .py extension files

If one module wants to use another module then it uses import keyword.

import modulename , import package.* , from module import class

NUMPY

1) Numpy is a python library used for working with arrays.

2) It also has functions for working in domain of linear algebra, fourier transform & matrices.

3) Numpy stands for numerical python.

4) Numpy aims to provide an array object that is upto 50 times faster than traditional python list.

5) The array objects in numpy is called <u>ndarray</u>, it provides a lot of supporting functions that make working with ndarray very easy.

Ex1)

```
import numpy as np

a = np.array([1,2,3,4,5])
print(a)

    [1,2,3,4,5]
```

Ex2)

```
import numpy as np

a = np.array([1, 2, 3, 4, 5])
b = np.array([[1,2,3], [4,5,6]])
c = np.array([[[1,2,3],[4,5,6]],
              [[7,8,9],[10,11,12]]])
```

	0	1	2
0	1	2	3
1	4	5	6

```
1 = print(a.ndim)
2 ⇐ print(b.ndim)
3 ⇐ print(c.ndim)
```

0

	0	1	2
0	1	2	3
1	4	5	6

1

	0	1	2
0	7	8	9
1	10	11	12

Example 3

```
import numpy as np
                0 1 2 3 4
a = np.array([1,2,3,4,5])
arr = np.array([[1,2,3,4,5], [6,7,8,9,10]])
arr1 = np.array([[[1,2,3],[4,5,6]], [[7,8,9], [10,11,12]]]

print(a[2]+ a[3]) = 7
print(arr[0,1]) = 2
print(arr[1,4]) = 10
print(arr1[0,1,2]) 6
```

Example 4

```
import numpy as np
                -7 -6 -5 -4 -3 -2 -1
arr = np.array([1, 2, 3, 4, 5, 6, 7])
                0  1  2  3  4  5  6
                   ↑
print( arr[1:5])     [2,3,4,5]
print( arr[4:])      [5,6,7]
print( arr[:4])      [1,2,3,4]
print( arr[-3:-1]))  [5,6]
print( arr[1:5:2])   [2,4]
print( arr[::2])     [1,3,5,7]
```

Example 5

```
import numpy as np

arr = np.array([[1,2,3,4,5],
                [6,7,8,9,10]])

[7,8,9]
[3,8]  print( arr[1, 1:4])
                 row  column
       print( arr[0:2, 2]
                 row   column
       print( arr[0:2, 1:4])

[[2,3,4]
 [7,8,9]]
```

	0	1	2	3	4
0	1	2	3	4	5
1	6	7	8	9	10

Example 6

```
import numpy as np
arr = np.array([1,2,3,4])
arr1 = np.array(['apple', 'banana', 'cherry'])
print(arr.dtype)    int64 or int32
print(arr1.dtype)   <U6
```

Example 7

```
import numpy as np
arr= np.array([1,2,3,4], dtype=
                          'i4')
print(arr.dtype)
     int32
```

Example 8

```
import numpy as np
```

$\texttt{arr = np.array(}[\underbrace{[1,2,3]}_{0},\underbrace{[4,5,6]}_{1}]\texttt{)}$

```
for x in arr :
    for y in x :
        print(y)
```

Ex 9)

```
import numpy as np
```

$\texttt{arr = np.array(}[\overbrace{[[1,2,3], [4,5,6]]}, \underbrace{[[7,8,9], [10,11,12]]}]\texttt{)}$

```
for x in arr :
    for y in x :
        for z in y:
            print(z)
```

Math module

The math module in python is a built in standard module that provide acess to mathematical functions.

Category	Functions
Basic Math	ceil( ), floor( ), fabs( ), factorial( )
Trigonometry	sin( ), cos( ), tan( ), degrees( ), radians( )
Logarithmic/Exponents	log( ), log10( ), exp( ), pow( )
Root	sqrt( )
constants	pi, e, nan, inf, tau
combinatorics	comb( ), perm( )
Hyperbolic functions	sinh( ), cosh( ), tanh( )

```
import math

print(math.floor(4.21)) = 4

print(math.ceil(5.5)) = 6

print(math.fabs(-10)) = 10

print(math.sqrt(16)) 4.0

print(math.factorial(5)) 120
```

**

$$\underbrace{2**3}$$

```
print(math.pow(2,3))
```

$\boxed{8}$

```
math.pow(2,3,2)
```

```
(2**3)%2
```

```
math.pow(2,
math.pow(3,2))
```

Collections Module

The collections module in python is a built-in module that provides specialized container datatypes that are alternatives to Python's general purpose build in containers like dict, list, set, & tuple.

Some of these types offer extra features, better performance

NamedTuple

NamedTuple returns a tuple object with names for each position which the ordinary tuple lack.

```
a = ("Rohit", 21, "Student")

⇒ a[0]
  a[1]
  a[2]
```

```
from collections import namedtuple

Student = namedtuple('Student', ['name',
                                 'age',
                                 'prog'])

⇒ S = Student('Rohit', 21, 'student')

print(s.name) 'Rohit'
```

2) deque (Double ended queue) (Fast append and pop from both ends)

```
from collections import deque

d = deque([1,2,3])
⇒ d. append(4)
d. appendleft(0)
d.pop( )
d.popleft( )
print(d )
```

[1,2,3]

[1,2,3,4]

[0,1,2,3,4]

[0,1,2,3]

$\boxed{[1,2,3]}$

3) Counter

```
a = { }
a['x'] = 1 => error
```

```
a = {'x':1, 'y':2]
a['z'] => error
```

```
from collections import Counter

c = Counter("banana")
print(c) = {'a':3, 'n':2, 'b':1}

print(c.most_common(1))   [('a',3)]
```

4) defaultdict

Like a dictionary but provides a default value for missing key.

```
a = defaultdict(int) => default 0
a["x"] += 1
a["b"] += 2
print(a["z"]) = 0
```

datetime module

The datetime module in python is used to work with dates & time.

```
import datetime
```

1)
```
from datetime import date
d = date(2025, 7, 25)
print(d)            2025-07-25
print(d.year)       2025
print(d.month)      7
print(d.day)        25
print(date.today())
```

3)
```
from datetime import datetime
a = datetime.now( )
print(a)
print(a.strftime("%d-%b-%Y %H:%M:%S))
```

2)
```
from datetime import time
t = time(14, 30, 0)
print(t)            14.30:00
print(t.hour)       hh:mm:ss
print(t.minute)
print(t.second)
```

Code	Meaning	Example
%Y	Year (4-digit)	2025
%m	month(01-12)	07
%d	Day(01-31)	25
%H	Hour (00-23)	23
%M	minute(00-59)	54
%S	Second (00-59)	51
%b	Month Name (short)	Jul

Os module

Os module stands Operating system.

It helps your python program to interact with file system and operating system

```
import Os

Os.remove("filename")
Os.rmdir("folder name")
Os.mkdir("folder name")
Os.rename("filename")
```

THANK - YOU