

① matplotlib → plotting library

13 TUE

from matplotlib import pyplot as plt

② numpy → general purpose array processing package.

→ high performance multidimensional array, object & tools.

import numpy as np.

③ panda → easy to use data structure & data analysis.

import pandas as pd.

14 WED

④ tensorflow → end to end open source platform for machine learning.

- It has comprehensive, flexible ecosystem of tools, libraries.

19 MON

Panda fun^m

• Series

S = Pd.series([3, -5, 7, 4], index=['a', 'b', 'c', 'd'])

o/p index

A	3
B	-5
C	7
D	4

• data frame

20 TUE

creating a dictionary where each key will be dataframe column

data = { 'pais': ['Belgica', 'India', 'Brasil'],
 'capital': ['ABC', 'XYZ', 'JKL'],
 'populn': [12346, 12378, 981011] }

creating the Dataframe

df = Pd.DataFrame(data, columns = ['pais', 'capital', 'populn'])

o/p =>

	pais	capital	populatr
1	Belgica	ABC	12346
2	India	XYZ	12378
3	Brasil	JKL	981011

March 2010	Sun	Mon	Tue	Wed	Thu	Fri	Sat
		1	2	3	4	5	6
	7	8	9	10	11	12	13
	14	15	16	17	18	19	20
	21	22	23	24	25	26	27
	28	29	30	31			

* opening & writing csv files

21 WED

```
pd.read_csv('file-name.csv') # Reading csv file
pd.read_csv('file-name.csv', encoding='ISO-8859-1')
# Reading a csv files encode in ISO 8859
pd.to_csv(name-of-the-file-to-save.csv)
# writing a csv file
```

* opening Excel files.

XLSX = ~~pdf~~

```
pd.ExcelFile('your-excel-file.xlsx')
df = pd.read_excel(xlsx, 'sheet 1')
```

* Removing rows by index

22 THU

```
s.drop([0,1]) # Removing rows by index
df.drop('country', axis=1) # Remove columns
using the arguments 'axis=1'
```

* Collecting basic information about the Data Frame

```
df.shape # Amount of Rows & columns
df.index # Index description
df.columns # columns in the DataFrame
df.count() # non-null data counts.
```


23 FRI ★ creating a new column in a DataFrame

`df['new column'] = 0` # It'll create a column called 'new column' with 0 is its value.

★ Renaming columns from a DataFrame

`df.columns = ['column 1', 'column 2', 'column 3']`

★ Summary of data

`df.sum()` # sum of values in a DataFrame

`df.min()` # lowest values of a DataFrame

24 SAT `df.max()` # Highest value

`df.idxmax()` # index of Highest value

`df.min()` # index of lowest value

`df.describe()` # statistical summary of the DataFrame, with quartiles, median, etc.

`df.mean()` # Average values

`df.median()` # median values.

★ ordering values

`df.sort_values()` # ordering in ascending order

`df.sort_values(ascending = False)`.

Notes

March 2010	Sun	Mon	Tue	Wed	Thu	Fri	Sat
	7	8	9	10	11	12	13
	14	15	16	17	18	19	20
	21	22	23	24	25	26	27

* Boolean indexing

`df[df['population'] % 2 == 0]` # filtering DataFrame to show only even values

* Exporting Data

`df.to_csv(filename)` // write to csv file

`df.to_excel(filename)` // write to an Excel file

* Viewing / Inspecting Data

`df.head(n)` // first n rows of the data frame

`df.tail(n)` // last n rows of data frame

APRIL 2010

27 TUE Numpy ::

* Arithmetic operam on numpy :

• array1 = np.arange(1,12)

array1 * array1

array1 - array1

array1 + array1

array1 / array1

* using universal funtm In Each element

np.sqrt(array1) // Return square root

np.exp(array1) // Return the exponential

28 WED np.log(array1) // Return logarithm

np.std(array1) // Return standard deviation

* Indexing

array1[2] // print 3

array2[1,1] // print 5

array2[0,:] // array [1,2,3] : row 1

* slicing :

array1 ([1:3]) // prints array ([2,3])

Notes

array([1:3]) = [-2, -3] // prints 29 THU
array([1, -2, -3, 4, 5, 6])

array([-3 :]) // prints array([4, 5, 6])
last 3 numbers

* Importing of file

np.loadtxt('file.txt') // import from text file

np.savetxt('file.txt', arr, delimiter = ',')
// write to a text file

* ndim (dimension)

30 FRI

import numpy as np
a = np.array([(1, 2, 3), (4, 5, 6)])

print(a.ndim)

O/P \Rightarrow 2

* itemsize (byte size of each element)

import numpy as np
a = np.array([(1, 2, 3)])
print(a.itemsize)

O/P \Rightarrow 4

May 2010	Sun	Mon	Tue	Wed	Thu	Fri	Sat
	30	31					1
	2	3	4	5	6	7	8
	9	10	11	12	13	14	15
	16	17	18	19	20	21	22

Notes

MAY 2010

1 SAT * dtype .

```
import numpy as np
a = np.array([(1, 2, 3)])
print(a.dtype)
```

⇒ int32

* Reshape

```
import numpy as np
a = np.array([(1, 2, 3), (4, 5, 6)])
print(a)
a = a.reshape(3, 2)
print(a)
```

2 SUN

1	2	3
4	5	6



1	2
3	4
5	6