

PART 1 : NumPy

1. Introduction to NumPy

In []: What **is** the NumPy ?

NumPy **is** a Python library used **for** working **with** arrays. It also has functions **for** working **in** domain of linear algebra, fourier transform, **and** NumPy was created **in 2005** by Travis Oliphant. It **is** an **open** source project **and** you can use it freely. NumPy stands **for** Numerical Python.

2. NumPy Arrays

Exercise: Create arrays of different dimensions and print their properties.

```
In [3]: import numpy as np
a = np.array([11,12,13,14])
print(a)
print(type(a))
```

```
[11 12 13 14]
<class 'numpy.ndarray'>
```

```
In [4]: import numpy as np
b=np.array([[11,12,13,14],[21,22,23,24]])
print(b)
```

```
[[11 12 13 14]
 [21 22 23 24]]
```

```
In [5]: import numpy as np
c=np.array([[11,12,13,14],[21,22,23,24],[31,32,33,34]])
print(c)
```

```
[[11 12 13 14]
 [21 22 23 24]
 [31 32 33 34]]
```

```
In [6]: import numpy as np
c=np.array([[11,12,13,14],[21,22,23,24],[31,32,33,34]])
print(c[0:1])
print("\n")
print(c[0:2])
print("\n")
print(c[0:3])
print("\n")
print(c[0:4])
```

```
[[11 12 13 14]]
```

```
[[11 12 13 14]
 [21 22 23 24]]
```

```
[[11 12 13 14]
 [21 22 23 24]
 [31 32 33 34]]
```

```
[[11 12 13 14]
 [21 22 23 24]
 [31 32 33 34]]
```

```
In [7]: import numpy as np
c=np.array([[11,12,13,14],[21,22,23,24],[31,32,33,34]])
print(c[1:1])
print("\n")
print(c[1:2])
print("\n")
print(c[1:3])
print("\n")
print(c[1:4])
```

```
[]
```

```
[[21 22 23 24]]
```

```
[[21 22 23 24]
 [31 32 33 34]]
```

```
[[21 22 23 24]
 [31 32 33 34]]
```

3. Array Operations

Exercise: Perform addition, subtraction, multiplication, and division on NumPy arrays.

```
In [9]: import numpy as np
a = np.array([11,12,13,14])
print(a+2)
print(type(a))
```

```
[13 14 15 16]
<class 'numpy.ndarray'>
```

```
In [10]: import numpy as np
a1 = np.array([11, 12, 13, 14])
a2 = np.array([15, 16, 17, 18])
print(a1 + a2)
```

```
[26 28 30 32]
```

```
In [11]: import numpy as np
a1 = np.array([11, 12, 13, 14])
a2 = np.array([15, 16, 17, 18])
print(a1 - a2)
```

```
[-4 -4 -4 -4]
```

```
In [12]: import numpy as np
a1 = np.array([11, 12, 13, 14])
a2 = np.array([15, 16, 17, 18])
print(a1 * a2)
```

```
[165 192 221 252]
```

```
In [13]: import numpy as np
a1 = np.array([11, 12, 13, 14])
a2 = np.array([15, 16, 17, 18])
print(a1 / a2)
```

```
[0.73333333 0.75      0.76470588 0.77777778]
```

```
In [14]: import numpy as np
a = np.array([11,12,13,14])
print(a*2)
```

```
[22 24 26 28]
```

4. Indexing and Slicing

Exercise: Practice indexing and slicing on multi-dimensional arrays.

```
In [16]: import numpy as np
a = np.array([11,12,13,14])
print(a[0])
print(a[1])
print(a[2])
print(a[3])
```

```
11
12
13
14
```

```
In [18]: import numpy as np
b=np.array([[1,2,3,4],[5,6,7,8]])
print(b)
```

```
[[1 2 3 4]
 [5 6 7 8]]
```

```
In [19]: import numpy as np
c=np.array([[11,12,13,14],[21,22,23,24],[31,32,33,34]])
print(c[1:1])
print("\n")
print(c[1:2])
print("\n")
print(c[1:3])
print("\n")
print(c[1:4])
```

```
[]
```

```
[[21 22 23 24]]
```

```
[[21 22 23 24]
 [31 32 33 34]]
```

```
[[21 22 23 24]
 [31 32 33 34]]
```

5. Mathematical Functions

Exercise: Apply various mathematical functions on arrays and interpret the results.

```
In [20]: import numpy as np
arr = np.array([1, 2, 3, 4, 5])
print(np.sum(arr))
print(np.mean(arr))
print(np.std(arr))
```

```
15
3.0
1.4142135623730951
```

6. Linear Algebra

Exercise: Perform matrix multiplication and compute eigenvalues and eigenvectors for given matrices.

```
In [22]: import numpy as np
matrix1 = np.array([[11, 12], [13, 14]])
matrix2 = np.array([[15, 16], [17, 18]])
print(np.dot(matrix1, matrix2))
```

```
[[369 392]
 [433 460]]
```

7. Random Module

Exercise: Generate random arrays and perform operations on them.

```
In [25]: import numpy as np
print(np.random.rand(3))
print(np.random.randint(0, 10, 5))
```

```
[0.2600269  0.42260272 0.63764519]
[8 4 5 6 4]
```

Part 2: Pandas

1. Introduction to Pandas

In []: What is Pandas ?

Pandas is a Python library used for working with data sets.
It has functions for analyzing, cleaning, exploring, and manipulating data.
The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and

2. Data Structures

Exercise: Create Series and DataFrames with different data types.

```
In [27]: import pandas as pd  
s = pd.Series([1, 2, 3, 4, 5])  
print(s)
```

```
0    1  
1    2  
2    3  
3    4  
4    5  
dtype: int64
```

```
In [28]: import pandas as pd  
data = {'A': [1, 2, 3], 'B': [4, 5, 6]}  
df = pd.DataFrame(data)  
print(df)
```

```
   A  B  
0  1  4  
1  2  5  
2  3  6
```

3. Data Manipulation

Exercise: Select, filter, and transform data in a DataFrame.

```
In [29]: import pandas as pd  
df = pd.DataFrame({'A': [11, 12, 13], 'B': [14, 15, 16]})  
print(df['A'])  
print(df[df['A'] > 1])
```

```
0    11  
1    12  
2    13  
Name: A, dtype: int64  
   A  B  
0  11 14  
1  12 15  
2  13 16
```

```
In [30]: import pandas as pd
df = pd.DataFrame({'A': [11, 12, 13], 'B': [14, 15, 16]})
df['C'] = df['A'] + df['B']
print(df['C'])
```

```
0    25
1    27
2    29
Name: C, dtype: int64
```

4. Data Aggregation

Exercise: Perform data aggregation using groupby and pivot tables.

```
In [33]: import pandas as pd
df = pd.DataFrame([[9, 4, 8, 9],
                  [8, 10, 7, 6],
                  [7, 6, 8, 5]],
                  columns=['Maths', 'English',
                          'Science', 'History'])
print(df)
```

	Maths	English	Science	History
0	9	4	8	9
1	8	10	7	6
2	7	6	8	5

```
In [38]: import pandas as pd
df = pd.DataFrame({'A': ['foo', 'bar', 'foo'], 'B': [1, 2, 3]})
grouped = df.groupby('A')
df.sum()
```

```
Out[38]: A    foobarfoo
B         6
dtype: object
```

```
In [37]: import pandas as pd
df = pd.DataFrame({'A': ['foo', 'bar', 'foo'], 'B': [1, 2, 3]})
df.pivot_table(values='B', index='A', aggfunc='sum')
```

```
Out[37]:
```

	B
A	
bar	2
foo	4

5. Handling Missing Data

Exercise: Handle missing data in a DataFrame.

```
In [39]: import pandas as pd
df = pd.DataFrame({'A': [1, 2, None], 'B': [4, None, 6]})
print(df.isnull())
```

	A	B
0	False	False
1	False	True
2	True	False