Workshop on Python (Day 3)

By Suriya G
Organized by Suresh Sir, UPNM

TABLE OF CONTENTS

01 Recap

- Functions & loops

02 NumPy

- Basic definitions
- Use cases

03 Arrays

- Building 1,2,3 D arrays
- Data Types
- Operators

04 Potential uses

- Getting advanced into numpy packages

What is NumPy?

- It is a general purpose array-processing packages.
- Used to compute faster scientific computations.
- It can integrate C++ code as well.
- Used for several mathematical operations.

How to use NumPy in python?

- The recommendation convention to import is:

numpy.py

```
# Basic syntax to import packages
Import numpy as np
```

Arrays in NumPy:

- NumPy’s main object is the homogeneous multidimensional array.
- It is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers.
- In NumPy dimensions are called **axes**. The number of axes is rank.
- NumPy’s array class is called **ndarray**. It is also known by the alias **array**.

1-D Arrays in NumPy:

2darrays.py

```
import numpy as np
a = np.array([0,1,2,3])
print(”Array = ”,a)
print(“Array shape = ”,a.shape)
print(“Array dimension = “,a.ndim)
print(“Array length = “,len(a))
print(“Array size = “,a.size)
```

2-D Arrays in NumPy:

2darrays.py

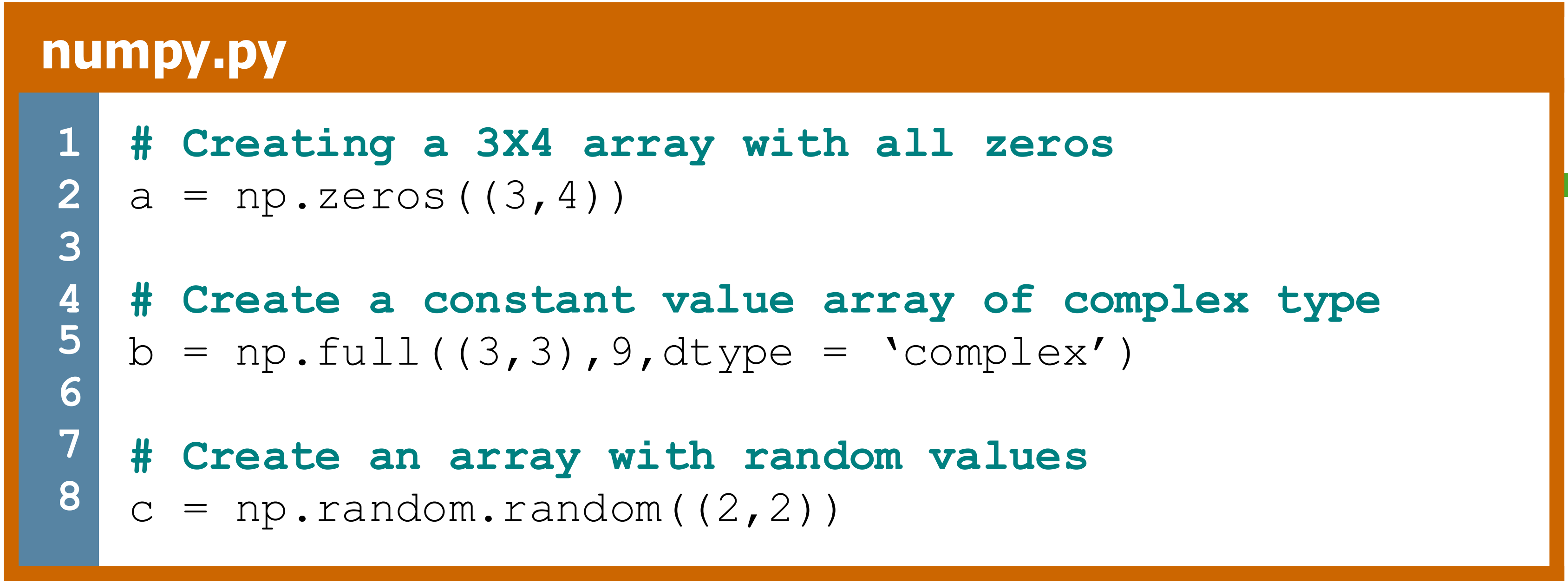
```
import numpy as np
a = np.array([[0,1,2,3],[4,5,6,7]])
print(”Array = ”,a)
print(“Array shape = ”,a.shape)
print(“Array dimension = “,a.ndim)
print(“Array length = “,len(a))#Return 1st Dimension’s size
print(“Array size = “,a.size)
```

3-D Arrays in NumPy:

3darrays.py

```python
import numpy as np
a =np.array([[[1,2,3],[4,5,6]],[[7,8,9],[10,11,12]]])
print(”Array = ”,a)
print(“Array shape = ”,a.shape)
print(“Array dimension = “,a.ndim)
print(“Array length = “,len(a))#Return 1st Dimension’s size
print(“Array size = “,a.size)
```

Other functions of NumPy:



numpy.py

```python
# Creating a 3X4 array with all zeros
a = np.zeros((3,4))

# Create a constant value array of complex type
b = np.full((3,3),9,dtype = ‘complex’)

# Create an array with random values
c = np.random.random((2,2))
```

Other functions of NumPy:

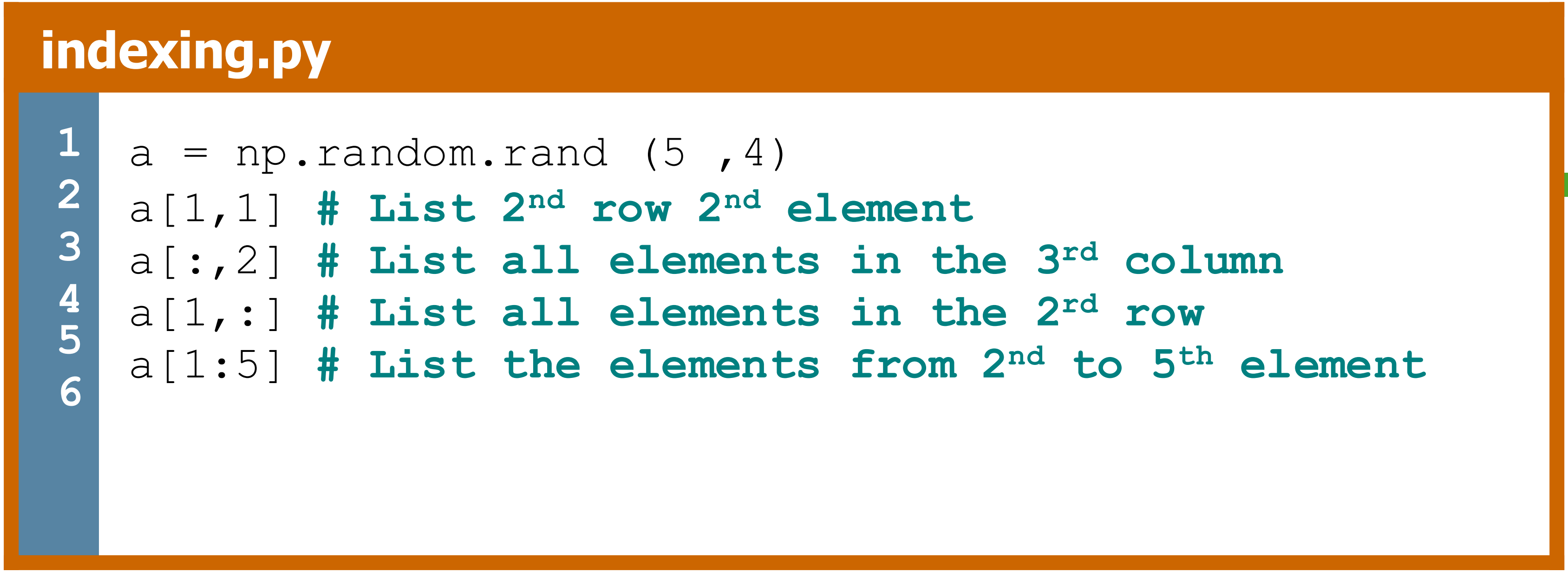
numpy.py

```
# Create a sequence from 0 to 30 with steps 5
f = np.arrange(0,30,5)

# Create a sequence of 10 values in range 0 to 5
g = np.linspace(0,5,10)

# Create an array with random values
c = np.random.random((2,2))
```

Indexing & Slicing:

- Slicing in python means taking elements from one given index to another given index.
- We pass slice instead of index like this: **[start:end].**
- We can also define the step, like this: **[start:end:step].**
- If we don’t pass start its considered 0
- If we don’t pass end its considered length of array in that dimension
- If we don’t pass step its considered 1

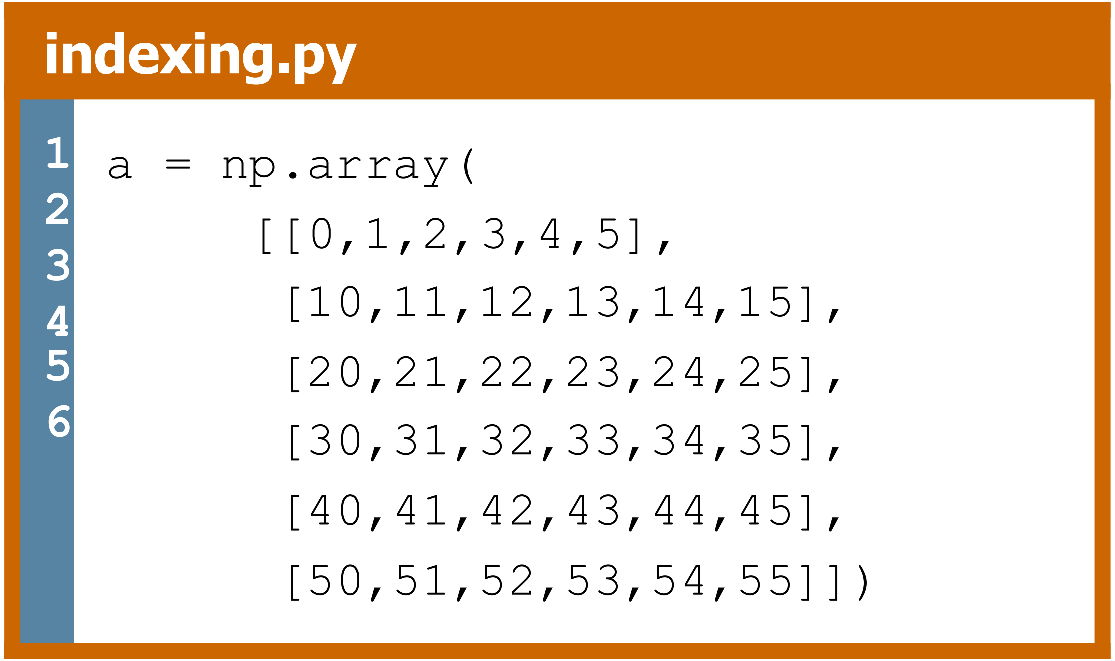
Indexing & Slicing:



indexing.py

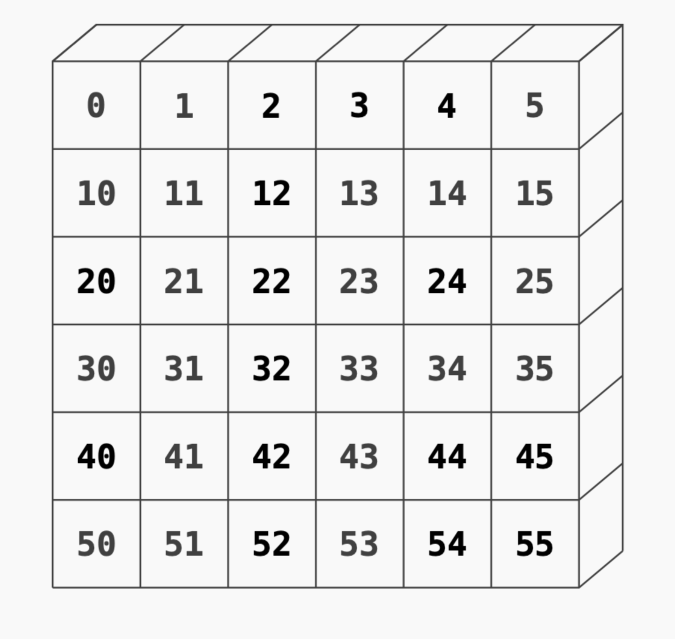
```
a = np.random.rand (5 ,4)
a[1,1] # List 2nd row 2nd element
a[:,2] # List all elements in the 3rd column
a[1,:] # List all elements in the 2rd row
a[1:5] # List the elements from 2nd to 5th element
```

Indexing & Slicing:



indexing.py

```
a = np.array(
        [[0,1,2,3,4,5],
         [10,11,12,13,14,15],
         [20,21,22,23,24,25],
         [30,31,32,33,34,35],
         [40,41,42,43,44,45],
         [50,51,52,53,54,55]])
```

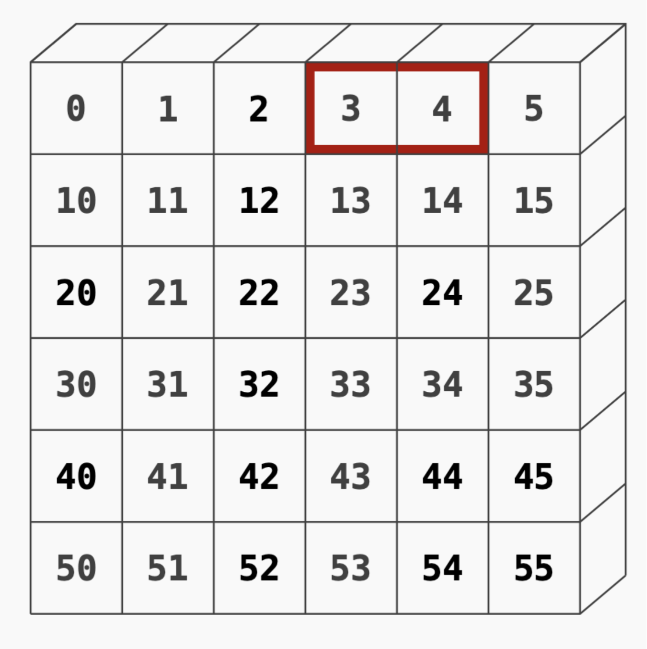


0	1	2	3	4	5
10	11	12	13	14	15
20	21	22	23	24	25
30	31	32	33	34	35
40	41	42	43	44	45
50	51	52	53	54	55

Indexing & Slicing:

indexing.py

```
a[0,3:5]
```

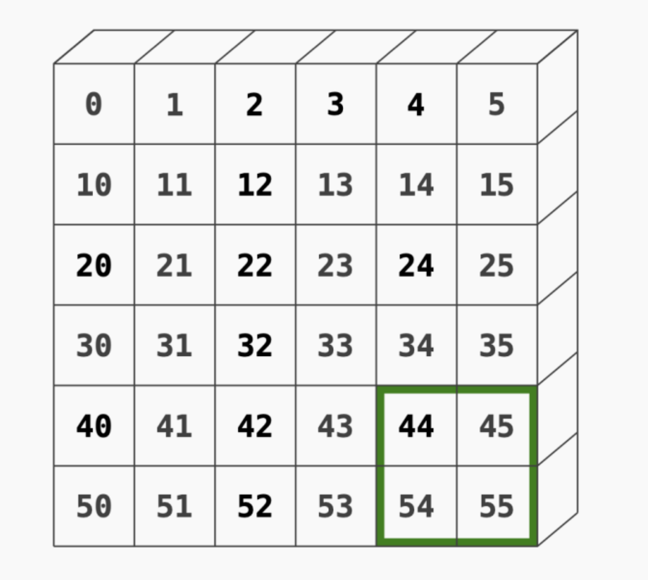


0	1	2	3	4	5
10	11	12	13	14	15
20	21	22	23	24	25
30	31	32	33	34	35
40	41	42	43	44	45
50	51	52	53	54	55

Indexing & Slicing:

indexing.py

```python
a[4:,4:]
```

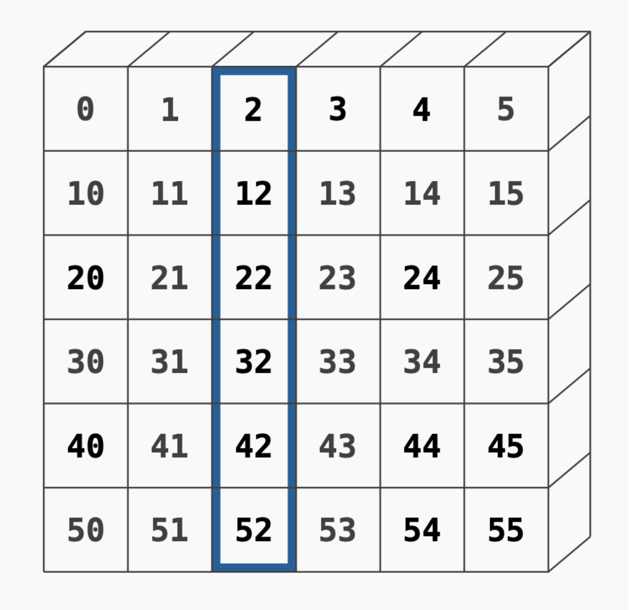


0	1	2	3	4	5
10	11	12	13	14	15
20	21	22	23	24	25
30	31	32	33	34	35
40	41	42	43	44	45
50	51	52	53	54	55

Indexing & Slicing:

indexing.py

```python
a[:,2]
```

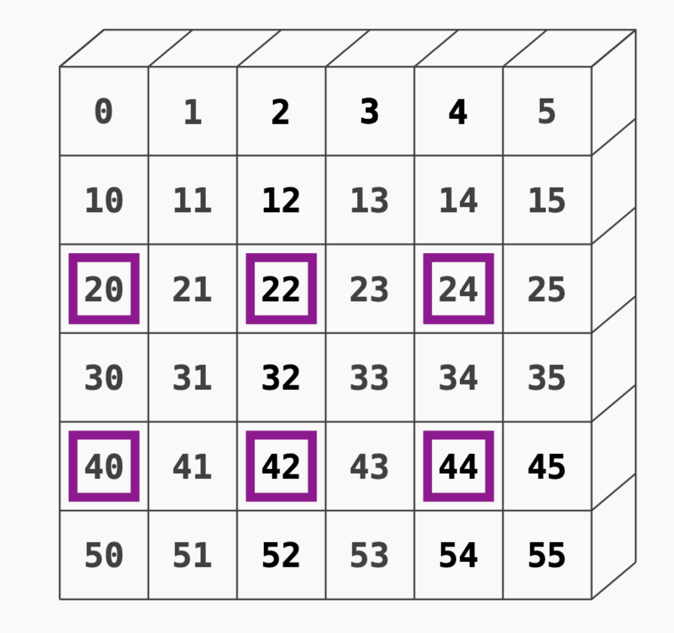


0	1	2	3	4	5
10	11	12	13	14	15
20	21	22	23	24	25
30	31	32	33	34	35
40	41	42	43	44	45
50	51	52	53	54	55

Indexing & Slicing:

indexing.py

```
a[2:2,::2]
```



0	1	2	3	4	5
10	11	12	13	14	15
20	21	22	23	24	25
30	31	32	33	34	35
40	41	42	43	44	45
50	51	52	53	54	55

Basic Operations in Arrays:

numpy.py

```
import numpy as np
a = np.array([1,2,5,3])
print(“Adding 1 to every element”, a+1)
print(“Subracting 3 to every element”, a-3)
print(“Multiplying each element by 10”, a*10)
print(“Squaring each element”, a**2)
```

Basic Operations in Matrix:

numpy.py

```
import numpy as np
arr = np.array([[1,5,6],
              [4,7,2],
              [3,1,9]])
print(“Largest element is:”, arr.max())
print(“Row-wise maximum elements:”,arr.max(axis=1))
print(“Col-wise maximum elements:”,arr.max(axis=0))
print(“Sum of all elements:”, arr.sum())
print(“Cumulative sum along each row:”,
                     arr.cumsum(axis=1)
```