

## Practical-2: Implement Python Program for NumPy Arrays

```
import numpy as np

# Creating array object
arr = np.array([[1, 2, 3],
                [4, 2, 5]])

# Printing type of arr object
print("Array is of type: ", type(arr))

# Printing array dimensions (axes)
print("No. of dimensions: ", arr.ndim)

# Printing shape of array
print("Shape of array: ", arr.shape)

# Printing size (total number of elements) of array
print("Size of array: ", arr.size)

# Printing type of elements in array
print("Array stores elements of type: ", arr.dtype)
```

### Output:

The screenshot shows a Jupyter Notebook interface with a single code cell executed. The cell contains code to print various properties of a NumPy array. The output pane displays the results of these prints.

```
# Printing size (total number of elements) of array
print("Size of array: ", arr.size)

# Printing type of elements in array
print("Array stores elements of type: ", arr.dtype)
```

Output:

```
[1]: ... Array is of type: <class 'numpy.ndarray'>
... No. of dimensions: 2
... Shape of array: (2, 3)
... Size of array: 6
... Array stores elements of type: int64
```

At the bottom, the status bar shows "9:33 AM" and "Python 3".

## Extra:

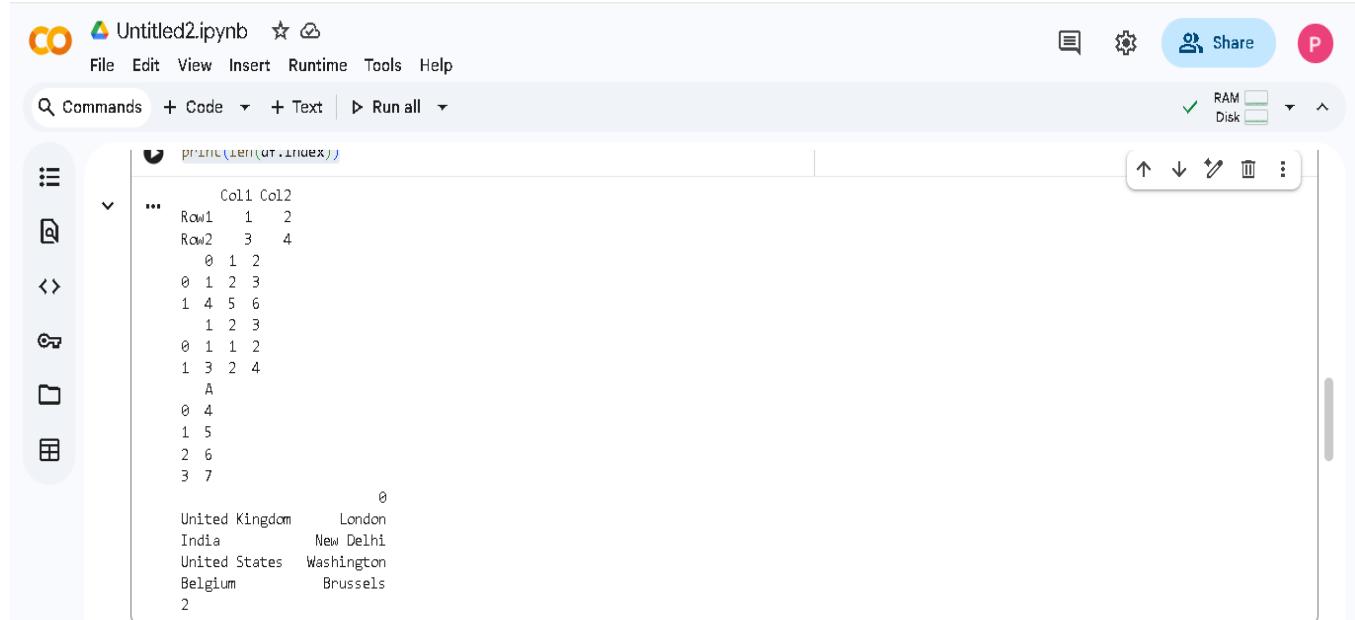
### Practical-2.1: More about NumPy Arrays and data frames

```
import numpy as np
import pandas as pd
data = np.array([[ '', 'Col1', 'Col2'], ['Row1', 1, 2],
['Row2', 3, 4]])
print(pd.DataFrame(data=data[1:,1:],
index = data[1:,0], columns=data[0,1:]))
# Take a 2D array as input to your DataFrame
my_2darray = np.array([[1, 2, 3], [4, 5, 6]])
print(pd.DataFrame(my_2darray))
# Take a dictionary as input to your DataFrame
my_dict = {1: ['1', '3'], 2: ['1', '2'], 3: ['2', '4']}
print(pd.DataFrame(my_dict))
# Take a DataFrame as input to your DataFrame
my_df = pd.DataFrame(data=[4,5,6,7], index=range(0,4), columns=['A'])
print(pd.DataFrame(my_df))
# Take a Series as input to your DataFrame
my_series = pd.Series({"United Kingdom": "London", "India": "New Delhi",
"United States": "Washington", "Belgium": "Brussels"})
print(pd.DataFrame(my_series))
df = pd.DataFrame(np.array([[1, 2, 3], [4, 5, 6]]))

# Use the `shape` property print(df.shape)

# Or use the `len()` function with the `index` property
print(len(df.index))
```

## Output:



The screenshot shows a Jupyter Notebook interface with the following details:

- File Bar:** Untitled2.ipynb, File, Edit, View, Insert, Runtime, Tools, Help.
- Toolbar:** Share, RAM/Disk status.
- Code Cell:** `print(df.index)` output:

	Col1	Col2
0	1	2
1	3	4
- Code Cell:** `df = pd.DataFrame(np.array([[1, 2, 3], [4, 5, 6]]))` output:

	0	1	2
0	1	2	3
1	4	5	6
- Code Cell:** `# Use the `shape` property print(df.shape)` output: `(2, 2)`
- Code Cell:** `# Or use the `len()` function with the `index` property` output: `2`
- Code Cell:** `print(len(df.index))` output: `2`

