TAKEMIND INTERNSHIP (ASSIGNMENT 1)

What is Numpy?

Numpy is an open source module of python that offers functions and routines for fast mathematical computation on arrays and matrices.In order to use numpy, we have to must import in our module by using a statement like:-

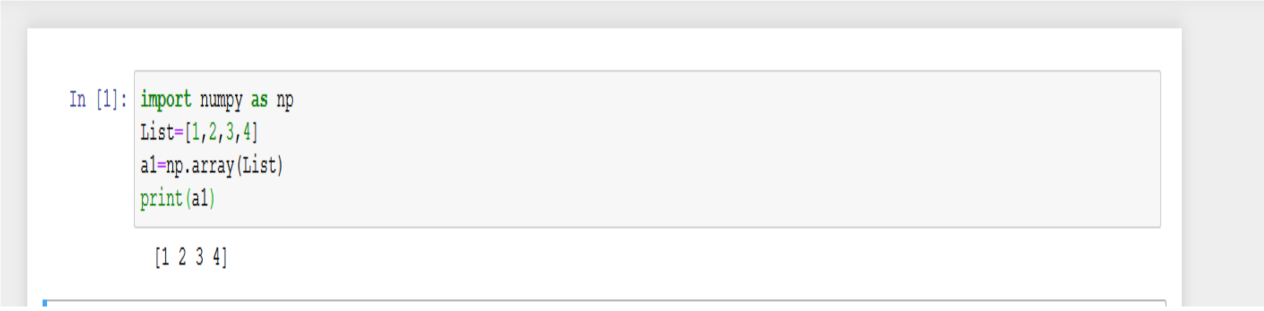
```
import numpy as np
```

#You can use any identifier in place of np but np has been a preferred choice generally which can be easily understandable in globally.

e.g:- numpy.array() is same as np.array()

Array in generally refers to a named group of homogenous of same type elements.

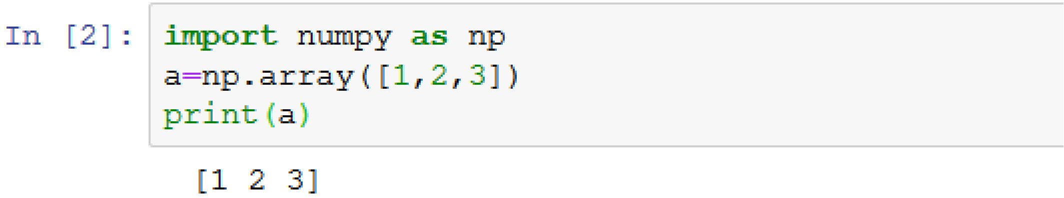
Consider the following code:-



```
In [1]: import numpy as np
        List=[1,2,3,4]
        a1=np.array(List)
        print(a1)

        [1 2 3 4]
```

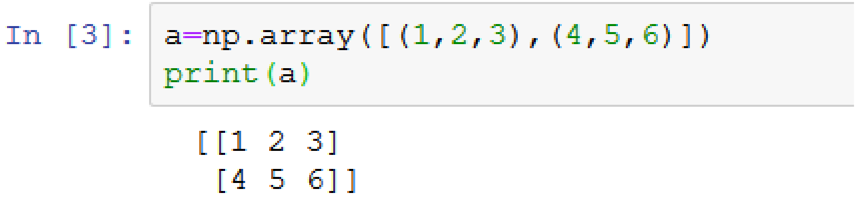
NumPy Array: Numpy array is a powerful N-dimensional array object which is in the form of rows and columns. We can initialize numpy arrays from nested Python lists and access it elements. In order to perform these numpy operations, the next question which will come in your mind is:

Single-dimensional Numpy Array:



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

        [1 2 3]
```

Multi-dimensional Array:



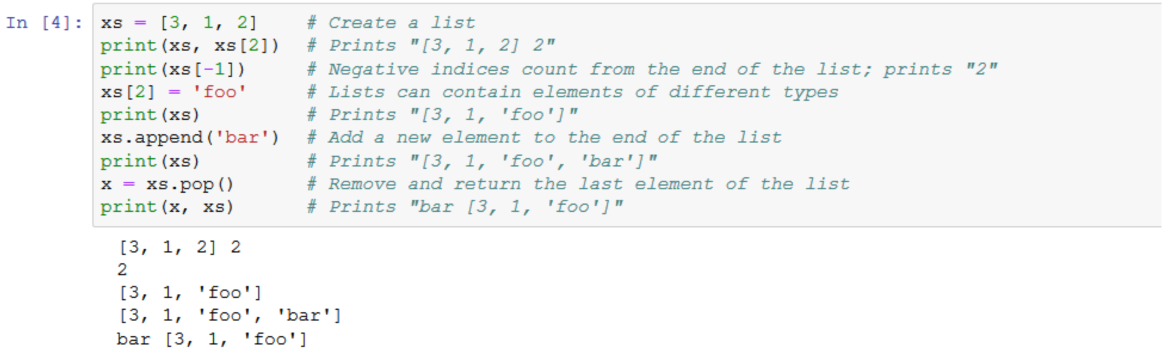
```
In [3]: a=np.array([(1,2,3),(4,5,6)])
        print(a)

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

Containers

Python includes several built-in container types: lists, dictionaries, sets, and tuples.

Lists

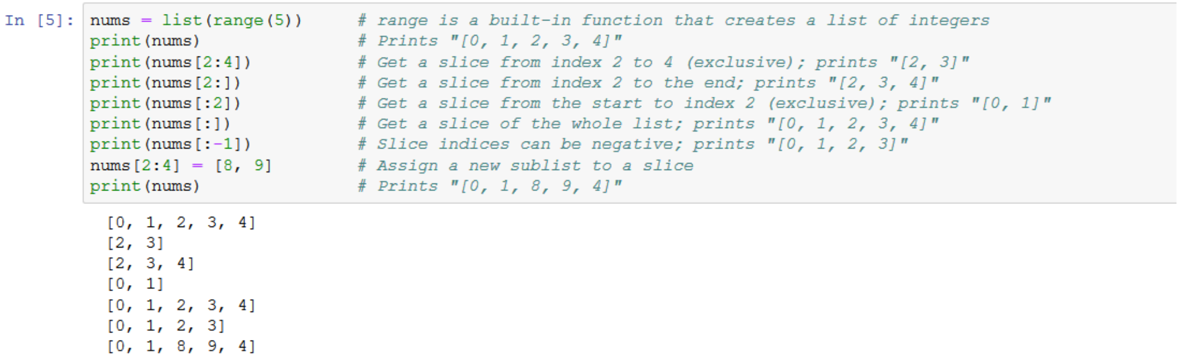
A list is the Python equivalent of an array, but is resizeable and can contain elements of different types:



```
In [4]: xs = [3, 1, 2]    # Create a list
        print(xs, xs[2])  # Prints "[3, 1, 2] 2"
        print(xs[-1])     # Negative indices count from the end of the list; prints "2"
        xs[2] = 'foo'     # Lists can contain elements of different types
        print(xs)         # Prints "[3, 1, 'foo']"
        xs.append('bar')  # Add a new element to the end of the list
        print(xs)         # Prints "[3, 1, 'foo', 'bar']"
        x = xs.pop()      # Remove and return the last element of the list
        print(x, xs)      # Prints "bar [3, 1, 'foo']"
```

```
[3, 1, 2] 2
2
[3, 1, 'foo']
[3, 1, 'foo', 'bar']
bar [3, 1, 'foo']
```

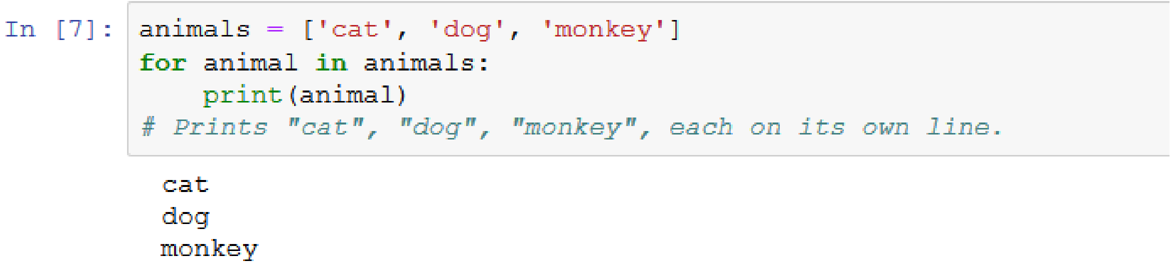
Slicing: In addition to accessing list elements one at a time, Python provides concise syntax to access sublists; this is known as *slicing*:



```
In [5]: nums = list(range(5))     # range is a built-in function that creates a list of integers
        print(nums)               # Prints "[0, 1, 2, 3, 4]"
        print(nums[2:4])          # Get a slice from index 2 to 4 (exclusive); prints "[2, 3]"
        print(nums[2:])           # Get a slice from index 2 to the end; prints "[2, 3, 4]"
        print(nums[:2])           # Get a slice from the start to index 2 (exclusive); prints "[0, 1]"
        print(nums[:])            # Get a slice of the whole list; prints "[0, 1, 2, 3, 4]"
        print(nums[:-1])          # Slice indices can be negative; prints "[0, 1, 2, 3]"
        nums[2:4] = [8, 9]        # Assign a new sublist to a slice
        print(nums)               # Prints "[0, 1, 8, 9, 4]"
```

```
[0, 1, 2, 3, 4]
[2, 3]
[2, 3, 4]
[0, 1]
[0, 1, 2, 3, 4]
[0, 1, 2, 3]
[0, 1, 8, 9, 4]
```

Loops: You can loop over the elements of a list like this:



```
In [7]: animals = ['cat', 'dog', 'monkey']
        for animal in animals:
            print(animal)
        # Prints "cat", "dog", "monkey", each on its own line.
```

```
cat
dog
monkey
```

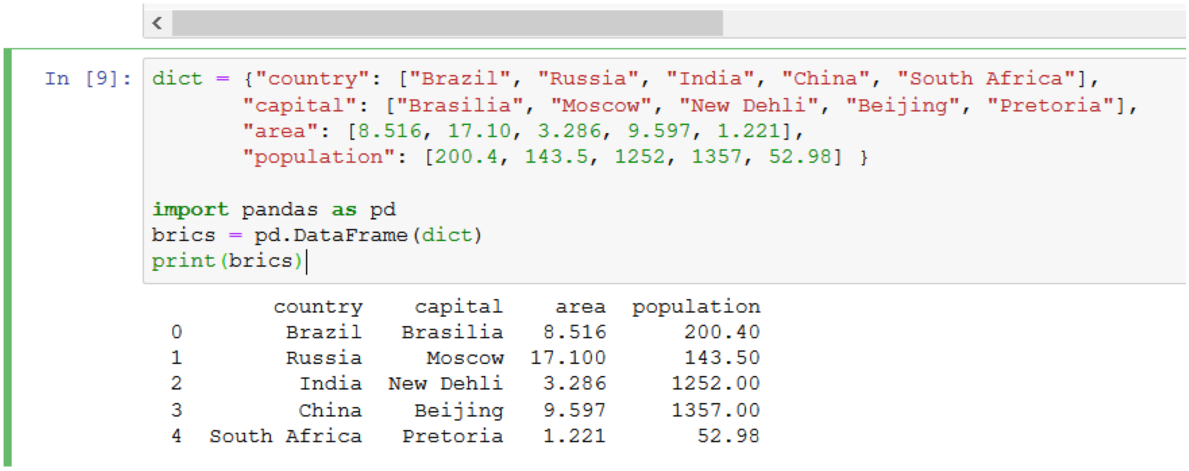
What is Pandas?

The pandas package is the most important tool at the disposal of Data Scientists and Analysts working in Python today. The powerful machine learning and glamorous visualization tools may get all the attention, but pandas is the backbone of most data projects.

Pandas DataFrame?

Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the DataFrame. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables.

There are several ways to create a DataFrame. One way way is to use a dictionary. For example:



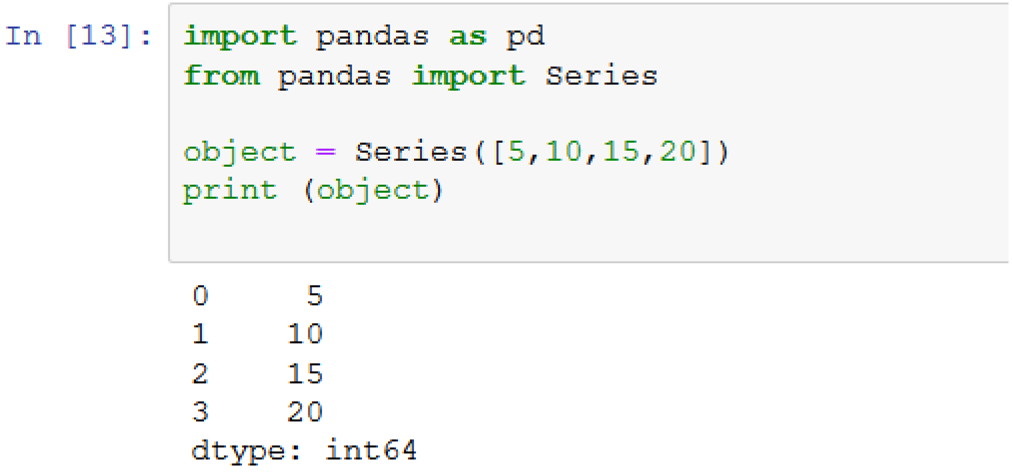
```
In [9]: dict = {"country": ["Brazil", "Russia", "India", "China", "South Africa"],
               "capital": ["Brasilia", "Moscow", "New Dehli", "Beijing", "Pretoria"],
               "area": [8.516, 17.10, 3.286, 9.597, 1.221],
               "population": [200.4, 143.5, 1252, 1357, 52.98] }

        import pandas as pd
        brics = pd.DataFrame(dict)
        print(brics)
```

```
        country    capital    area  population
0        Brazil   Brasilia   8.516      200.40
1        Russia     Moscow  17.100      143.50
2         India  New Dehli   3.286     1252.00
3         China    Beijing   9.597     1357.00
4  South Africa   Pretoria   1.221       52.98
```

Indexing DataFrames

There are several ways to index a Pandas DataFrame. One of the easiest ways to do this is by using square bracket notation.

In the example below, you can use square brackets to select one column of the cars DataFrame. You can either use a single bracket or a double bracket. The single bracket with output a Pandas Series, while a double bracket will output a Pandas DataFrame.



```
In [13]: import pandas as pd
         from pandas import Series

         object = Series([5,10,15,20])
         print (object)
```

```
0     5
1    10
2    15
3    20
dtype: int64
```

Major Applications Of Pandas?

1. Economics

Economics is in constant demand for data analysis. Analyzing data to form patterns and understanding trends about how the economy in various sectors is growing, is something very essential for economists. Therefore, a lot of economists have started using Python and Pandas to analyze huge datasets. Pandas provide a comprehensive set of tools, like dataframes and file-handling. These tools help immensely in accessing and manipulating data to get the desired results. Through these applications of Pandas, economists all around the world have been able to make breakthroughs like never before.

2. Statistics

Pure maths itself has made much progress with the various applications of Pandas. Since Statistic deals with a lot of data, a library like Pandas which deals with data handling has helped in a lot of different ways. The functions of mean, median and mode are just very basic ones which help in performing statistical calculations. There are a lot of other complex functions associated with statistics and pandas plays a huge role in these so as to bring perfect results.

3. Analytics

Analytics has become easier than ever with the use of Pandas. Whether it is website analytics or analytics of some other platform, Pandas do it all, with its amazing data manipulation and handling capabilities. The visualization capabilities of pandas play a big role too in this field. It not only takes in data and displays it but also helps in applying a lot of functions over the data.