Python for Data Science Certification Training Course

**Python NumPy**

**What Is NumPy?**
A library consisting of multidimensional array objects and a collection of routines for processing those arrays.

**Why NumPy?**
Mathematical and logical operations on arrays can be performed. Also provides high performance.

**Import Convention**
Import numpy as np - Import numpy

**ND Array**
Space efficient multi-dimensional array, which provides vectorized arithmetic operations.

**Creating Array**
- `a=np.array([1,2,3])`
- `b=np.array([[1,2,3,4],[7,8,9,10]],dtype=int)`

**Initial Place holders**
- `np.zeros(3)` - 1D array of length 3 all zeros array
- `np.zeros((2,3))` - 2D array of all zeros array
- `np.zeros((3,2,4))` - 3D array of all zeros array
- `np.full((3,4),2)` - 3x4 array with all values 2
- `np.random.rand(3,5)` - 3x5 array of random floats between 0-1
- `np.ones((3,4))` - 3x4 array with all values 1
- `np.eye(4)` - 4x4 array of 0 with 1 on diagonal

**Saving and Loading**
On disk:
- `np.save("new_array",a)`
- `np.savez("new_array.npz")`

Text/CSV files:
- `np.savetxt("New_file.txt",a)` - From a text file
- `np.genfromtxt("New_file.csv",delimiter=",")` - From a CSV file
- `np.savetxt("New_file.csv",a,delimiter=",")` - Writes to a text file
- `np.savetxt("New_file.csv",a,delimiter=",",fmt=",f8")` - Writes to a CSV file

**Properties**
- `array.size` - Returns number of elements in array
- `array.shape` - Returns dimensions of array (rows, columns)
- `array.dtype` - Returns type of elements in array

**Operations**
- `np.copyto(array)` - Copies array to new memory array.
- `np.view(dtype)` - Creates view of array elements with type dtype
- `np.sort()` - Sorts array
- `np.sort(axis=0)` - Sorts specific axis of array
- `np.reshape(2,3)` - Reshapes array to 2 rows, 3 columns without changing data.

**Adding**
- `np.append(array,values)` - Appends values to end of array
- `np.insert(array,4,values)` - Inserts values into array before index 4

**Removing**
- `np.delete(array,2,axis=0)` - Deletes row on index 2 of array
- `np.delete(array,3,axis=1)` - Deletes column on index 3 of array

**Combining**
- `np.concatenate((array1,array2),axis=0)` - Adds array2 as rows to the end of array1
- `np.concatenate((array1,array2),axis=1)` - Adds array2 as columns to end of array1

**Slicing**
- `np.split(array,3)` - Splits array into 3 sub-arrays

**Indexing**
- `a[0]=5` - Assigns array element on index 0 the value 5
- `a[2,3]=1` - Assigns array element on index [2][3] the value 1

**Subsetting**
- `a[2]` - Returns the element of Index 2 in array a
- `a[3,5]` - Returns the 2D array element on index [3][5]

**Slicing**
- `a[0:4]` - Returns the elements at indices 0,1,2,3
- `a[0:4,3]` - Returns the elements on rows 0,1,2,3 at column 3
- `a[2]` - Returns the elements at indices 0,1
- `a[1,1]` - Returns the elements at index 1 on all rows

**Array Mathematics**
**Arithmetic Operations**
- Addition: `np.add(a,b)`
- Subtraction: `np.subtract(a,b)`
- Multiplication: `np.multiply(a,b)`
- Division: `np.divide(a,b)`
- Exponentiation: `np.exp(a)`
- Square Root: `np.sqrt(b)`

**Comparison**
- Element-wise: `a==b`
- Array-wise: `np.array_equal(a,b)`

**Functions**
- Array-wise Sum: `a.sum()`
- Array-wise min value: `a.min()`
- Array row max value: `a.max(axis=0)`
- Mean: `a.mean()`
- Median: `a.median()`

**Learn from industry experts and be sought-after by the industry!**
**Learn any technology, show exemplary skills and have an unmatched career!**
**The most trending technology courses to help you fast-track your career!**
**Logical modules for both beginners and mid-level learners**

IntelliPaat

**FURTHERMORE:**
Python for Data Science Certification Training Course