

## **NumPy:**

### **Creation of NumPy Arrays:**

- i. `linspace(start, end, parts)` - divides values b/w start and end by equal parts value
- ii. `logspace(' ', ' ', '')` - same thing but uses logarithmic values
- iii. `zeros(value, dtype)` - creates zero matrix of specified i\*j form
- iv. `ones(value)` - creates matrix full of 1's of i\*j form

### **Matrices:**

- v. `full(value, number)` - creates matrix full of number of i\*j form
- vi. `np.eye(n)` - creates identity matrix of i\*j
- vii. `diag([values])` - as the name suggests
- viii. `np.random.rand([values])` - random matrix of specified dimensions
- ix. `np.random.randn([values])` - random matrix with values from a Gaussian Distribution
- x. `np.random.randint(start,end,[dimensions])` - random matrix with values within given range

### **Functions in NumPy:**

- xi. `.ndim` - to check dimensions of array
- xii. `.shape` - similar to `len()` function for other dtypes
- xiii. `.size` - no. of values
- xiii. `array.T` - transpose of matrix
- xiv. `.reshape((dimensions))` - reshaped array into specified dimensions
- xv. `.flatten()` - converts to 1-dim array
- xvi. `vstack()`, `hstack()`, `column_stack()` - represents given array in said direction
- xvii. `.concatenate((arr1,arr2),axis=0,1,2...)`
- xviii. `linalg.lstsq(a,b)` - linearalgebra's linear least squares function