

Numpy Activity

1. Write a NumPy program to create a 3x3 matrix with values ranging from 2 to 10.

```
1 import numpy as np
2 x = np.arange(2, 11).reshape(3,3)
3 print(x)
```

2. Write a NumPy program to convert a list of numeric value into a one-dimensional NumPy array.

```
1 import numpy as np
2 l = [12.23, 13.32, 100, 36.32]
3 print("Original List:",l)
4 a = np.array(l)
5 print("One-dimensional NumPy array: ",a)
```

3. Write a NumPy program to create an array with values ranging from 12 to 38.

```
1 import numpy as np
2 x = np.arange(12, 38)
3 print(x)
```

4. Write a NumPy program to add a border (filled with 0's) around an existing array.

```
1 import numpy as np
2 x = np.ones((3,3))
3 print("Original array:")
4 print(x)
5 print("0 on the border and 1 inside in the array")
6 x = np.pad(x, pad_width=1, mode='constant', constant_values=0)
7 print(x)
```

5. Write a NumPy program to append values to the end of an array.

```
1 import numpy as np
2 x = [10, 20, 30]
3 print("Original array:")
4 print(x)
5 x = np.append(x, [[40, 50, 60], [70, 80, 90]])
6 print("After append values to the end of the array:")
7 print(x)
```

6. Write a NumPy program to find the real and imaginary parts of an array of complex numbers.

```
1 import numpy as np
2 x = np.sqrt([1+0j])
3 y = np.sqrt([0+1j])
4 print("Original array:x ",x)
5 print("Original array:y ",y)
6 print("Real part of the array:")
7 print(x.real)
8 print(y.real)
9 print("Imaginary part of the array:")
10 print(x.imag)
11 print(y.imag)
```

7. Write a NumPy program to find the set difference of two arrays. The set difference will return the sorted, unique values in array1 that are not in array2.

```
1 import numpy as np
2 array1 = np.array([0, 10, 20, 40, 60, 80])
3 print("Array1: ",array1)
4 array2 = [10, 30, 40, 50, 70]
5 print("Array2: ",array2)
6 print("Unique values in array1 that are not in array2:")
7 print(np.setdiff1d(array1, array2))
```

8. Write a NumPy program to construct an array by repeating.

```
1 import numpy as np
2 a = [1, 2, 3, 4]
3 print("Original array")
4 print(a)
5 print("Repeating 2 times")
6 x = np.tile(a, 2)
7 print(x)
8 print("Repeating 3 times")
9 x = np.tile(a, 3)
10 print(x)
```

9. Write a NumPy program to create a contiguous flattened array.

```
1 import numpy as np
2 x = np.array([[10, 20, 30], [20, 40, 50]])
3 print("Original array:")
4 print(x)
5 y = np.ravel(x)
6 print("New flattened array:")
7 print(y)
```

10. Write a NumPy program to create a new array of 3*5, filled with 2.

```
1 import numpy as np
2 #using no.full
3 x = np.full((3, 5), 2, dtype=np.uint)
4 print(x)
5 #using no.ones
6 y = np.ones([3, 5], dtype=np.uint) *2
7 print(y)
```