PP Experiment 10

Aim: Exploring NumPy basics

Class: SE COMPS

Year: 2020-21

Performed by: Danyl Fernandes, 2020012004(72)

Performance date: 10-05-2021

In [3]:

```
import numpy as np

arr = np.array([[12, 13, 4, 5, 6], [3, 2, 7, 1]], dtype=object)

print("Array Metadata: \n")
print("Type (type): " + str(type(arr)))
print("N of dimensions (ndim): " + str(arr.ndim))
print("Shape (shape): " + str(arr.shape))
print("Size (size): " + str(arr.size))
print("Dimension Type (dtype): " + str(arr.dtype))
```

```
Array Metadata:

Type (type): <class 'numpy.ndarray'>
N of dimensions (ndim): 1
Shape (shape): (2,)
Size (size): 2
Dimension Type (dtype): object
```

In [2]:

```
import numpy as np

# Different ways of creating an array in numpy

# a) array from list and tuple
print("array from list and tuple:")
arr = np.array([[12, 13, 4, 5, 6], [3, 2, 7, 1]], dtype=object)
print(arr)

# b) zeros
print("\nzeros:")
arr = np.zeros((2, 2))
print(arr)

# c) full
print("\nfull:")
arr = np.full((2, 2), [1, 2])
print(arr)

# d) random
print("\nrandom:")
arr = np.random.rand(3,2)
print(arr)

# e) arange
print("\narrange:")
arr = np.arange(3,7)
print(arr)

# f) linspace
print("\nlinspace:")
arr = np.linspace(2, 3, num=5)
print(arr)

# g) reshape
print("\nreshape:")
arr = np.arange(6).reshape((3, 2))
print(arr)
```

```
array from list and tuple:
[list([12, 13, 4, 5, 6]) list([3, 2, 7, 1])]

zeros:
[[0. 0.]
 [0. 0.]]

full:
[[1 2]
 [1 2]]

random:
[[0.35225522 0.95431693]
 [0.88344597 0.5147357 ]
 [0.67215831 0.49112258]]

arrange:
[3 4 5 6]

linspace:
[2.   2.25 2.5  2.75 3.  ]

reshape:
```

```
[[0 1]
 [2 3]
 [4 5]]
```

In [14]:

```
import numpy as np

# Different arithmetic operations on an array in numpy
arr = np.arange(1,5)

# a) +=, -=, *= operators
print("+=, -=, *= operators:\n")
print("+= operator:")
arr += np.arange(5,9)
print(arr)

print("\n-= operator:")
arr -= np.arange(5,9)
print(arr)

print("\n*= operator:")
arr -= np.arange(5,9)
print(arr)

# b) sum, min, max operations
arr = np.array([12, 13, 4, 5, 6])

print("\nsum, min, max operations:\n")
print("sum operation:")
arr = np.sum(arr)
print(arr)

print("\nmin operation:")
arr = np.min(arr)
print(arr)

print("\nmax operation:")
arr = np.max(arr)
print(arr)

# c) +, - *, /
arr1 = np.array([12, 13, 4, 5, 6])
arr2 = np.array([3, 4, 2, 7, 5])

print("\n+, - *, / operations:\n")
print("+ operation:")
arr = arr1 + arr2
print(arr)

print("\n- operation:")
arr = arr1 - arr2
print(arr)

print("\n* operation:")
arr = arr1 * arr2
print(arr)
```

```
+=, -=, *= operators:

+= operator:
[ 6  8 10 12]

-= operator:
```

```
[1 2 3 4]

*= operator:
[-4 -4 -4 -4]

sum, min, max operations:

sum operation:
40

min operation:
40

max operation:
40

+, - *, / operations:

+ operation:
[15 17  6 12 11]

- operation:
[ 9  9  2 -2  1]

* operation:
```