a9v1arimb

January 23, 2025

[ ]:
```
import numpy as np
first_array=np.array([1,2,3])
second_array=np.array([5,6,7])
first_array+second_array #does element wise addition i.e␣
 ↪first_array[0]+second_array[0]
np.add(first_array,second_array) #can take any no. of arrays
first_array-second_array #similar to addition index/element wise
np.subtract(first_array,second_array)
first_array*second_array
np.multiply(first_array,second_array)
first_array/second_array
np.divide(first_array,second_array)
first_array^second_array
np.power(first_array,second_array)
first_array % second_array
np.mod(first_array,second_array)
```

[ ]:
```
array([1, 2, 3])
```

[ ]:
```
#Array Functions
array1=np.array([[1,2,3],[4,5,6],[7,8,9]]) # ceating a multi dimentional array
```

[ ]:
```
array([[1, 2, 3],
       [4, 5, 6],
       [7, 8, 9]])
```

[ ]:
```
array=np.ones((3,3)) # creates a 3x3 array with elements all 1
array=np.zeros((3,3)) # creates a 3x3 array with elements all 0

array=np.array([[1,2,3],[4,5,6],[7,8,9]])
np.transpose(array)
```

[ ]:
```
array([[1, 4, 7],
       [2, 5, 8],
       [3, 6, 9]])
```

[ ]:
```
#ststistical function
marks=np.array([90,95,99,98,100])
np.average(marks)
np.median(marks)
np.min(marks)
np.max(marks)
```

[ ]:
```
100
```

[ ]:
```
array=np.array([1,2,3])
np.savetxt("test.txt",array)
```

[ ]:
```
load_data=np.loadtxt("test.txt")
load_data
```

[ ]:
```
array([1., 2., 3.])
```

[ ]:
```
#comparison operator
arr1=np.array([1,2,3])
arr2=np.array([6,7,8])
arr1>arr2
arr1<arr2
arr1==arr2
arr1!=arr2
np.less_equal(arr1,arr2)
```

[ ]:
```
array([ True,  True,  True])
```

[ ]:
```
#constant defination
np.pi #pi as a constant
class numpy:
  def __init__(self):
    self.pi=3.1415
  array=np.array([1,2,3])
  print(array+np.pi)
```

```
[4.14159265 5.14159265 6.14159265]
```