In [1]:

```
import numpy as np
```

Broadcasting

In [3]:

```
# row  col
x = np.array([[1,2,3]]) #1 row 3 col(1,3) 1 row to add
print (x)
print(x.shape)

y = np.array([[4],[5]]) #2 row 1 col(2,1) 2nd col to be added to make it equal
print(y)
print(y.shape)
```

```
[[1 2 3]]
(1, 3)
[[4]
 [5]]
(2, 1)
```

In [6]:

```
z,v = np.broadcast_arrays(x,y) #row isne 2 repeat kia 1 time ziada krdia
z
```

Out[6]:

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

In [7]:

```
v
```

Out[7]:

```
array([[4, 4, 4],
       [5, 5, 5]])
```

In [8]:

```
a = np.array([1,2,3],[2,2,2],[3,3,3]) #(3,3)
b = np.array([[4],[5]])(2,1)
a,b= np.broadcast_arrays(a,b)  # yahan ye conflict he row 1 4 ko repeat krun ya 5 ko.
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-8-3cba664c1091> in <module>
----> 1 a = np.array([1,2,3],[2,2,2],[3,3,3])
      2 b = np.array([[4],[5]])
      3 a,b= np.broadcast_arrays(a,b)  # yahan ye conflict he row 1 4 ko rep
eat krun ya 5 ko. same in case of col

ValueError: only 2 non-keyword arguments accepted
```

In [9]:

```
c = np.array([[1,2,3]])  #(1,3)
d = np.array([[1],[2],[3]])#(3,1)
c+d
```

Out[9]:

```
array([[2, 3, 4],
       [3, 4, 5],
       [4, 5, 6]])
```

In [10]:

```
c.shape, d.shape
```

Out[10]:

```
((1, 3), (3, 1))
```

In [13]:

```
e,f=np.broadcast_arrays(c,d)
e #1,3 ko 3,3
```

Out[13]:

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

In [14]:

```
f#3,1 ko 3,3
```

Out[14]:

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

In [15]:

```
e+f
```

Out[15]:

```
array([[2, 3, 4],
       [3, 4, 5],
       [4, 5, 6]])
```

In [16]:

```
c
```

Out[16]:

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

In [17]:

```
d
```

Out[17]:

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

In [19]:

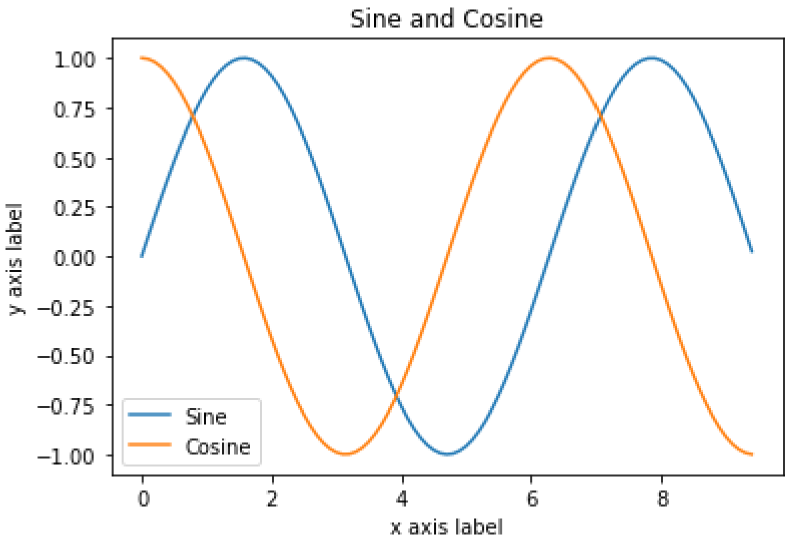
```
#EXTRA

import matplotlib.pyplot as plt

# Computes x and y coordinates for
# points on sine and cosine curves
x = np.arange(0, 3 * np.pi, 0.1)
y_sin = np.sin(x)
y_cos = np.cos(x)

# Plot the points using matplotlib
plt.plot(x, y_sin)
plt.plot(x, y_cos)
plt.xlabel('x axis label')
plt.ylabel('y axis label')
plt.title('Sine and Cosine')
plt.legend(['Sine', 'Cosine'])

plt.show()
```



In [21]:

```
x = np.arange(0, 3 * np.pi, 0.1)
x
```

Out[21]:

```
array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. , 1.1, 1.2,
       1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2. , 2.1, 2.2, 2.3, 2.4, 2.5,
       2.6, 2.7, 2.8, 2.9, 3. , 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, 3.8,
       3.9, 4. , 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 5. , 5.1,
       5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9, 6. , 6.1, 6.2, 6.3, 6.4,
       6.5, 6.6, 6.7, 6.8, 6.9, 7. , 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7,
       7.8, 7.9, 8. , 8.1, 8.2, 8.3, 8.4, 8.5, 8.6, 8.7, 8.8, 8.9, 9. ,
       9.1, 9.2, 9.3, 9.4])
```

In [22]:

```
arr = np.arange(32).reshape(8,4)
print(arr)
arr[[1, 5, 7, 2]][:, [0, 3, 1, 2]]

# 1[0, 3, 1, 2],5[0, 3, 1, 2],7[0, 3, 1, 2],2[[0, 3, 1, 2]]
# (1,0)(1,3)(1,1)(1,2)(5,0)()()()()()()()()()()().....
```

```
[[ 0  1  2  3]
 [ 4  5  6  7]
 [ 8  9 10 11]
 [12 13 14 15]
 [16 17 18 19]
 [20 21 22 23]
 [24 25 26 27]
 [28 29 30 31]]
```

Out[22]:

```
array([[ 4,  7,  5,  6],
       [20, 23, 21, 22],
       [28, 31, 29, 30],
       [ 8, 11,  9, 10]])
```

In [23]:

```
arr = np.empty((8, 4))

for i in range(8):
    arr[i] = i
```

In [24]:

```
arr
```

Out[24]:

```
array([[0., 0., 0., 0.],
       [1., 1., 1., 1.],
       [2., 2., 2., 2.],
       [3., 3., 3., 3.],
       [4., 4., 4., 4.],
       [5., 5., 5., 5.],
       [6., 6., 6., 6.],
       [7., 7., 7., 7.]])
```

In [25]:

```
a  = [22,33,44,55,77,55,44,22,11]
a
```

Out[25]:

```
[22, 33, 44, 55, 77, 55, 44, 22, 11]
```

In [26]:

```
b=set(a)
```

In [27]:

```
b
```

Out[27]:

```
{11, 22, 33, 44, 55, 77}
```

In [31]:

```
arr = np.arange(32)
arr
```

Out[31]:

```
array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
       17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31])
```

In [35]:

```
arr = np.arange(32).reshape(8,4)
arr
```

Out[35]:

```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
       [ 8,  9, 10, 11],
       [12, 13, 14, 15],
       [16, 17, 18, 19],
       [20, 21, 22, 23],
       [24, 25, 26, 27],
       [28, 29, 30, 31]])
```

In [ ]:

```

```

In [ ]:

```

```