In [1]:

```
# Joining NumPy Arrays
# Joining means putting contents of two or more arrays in a single array.
import numpy as np
```

In [2]:

```
a1 = np.array([1, 2, 3])

a2 = np.array([4, 5, 6])
```

In [3]:

```
aa = np.concatenate((a1, a2))

aa
```

Out[3]:

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

In [ ]:

```

```

In [4]:

```
# Join two 2-D arrays along rows (axis=1):

a3 = np.array([[1, 2], [3, 4]])

a4 = np.array([[5, 6], [7, 8]])

aa = np.concatenate((a3, a4), axis=1)
aa
```

Out[4]:

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

In [ ]:

```

```

In [5]:

```
a3 = np.array([[1, 2], [3, 4]])

a4 = np.array([[5, 6], [7, 8]])

aa = np.concatenate((a3, a4), axis=0)
aa
```

Out[5]:

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

In [ ]:

```

```

In [6]:

```
# Joining Arrays Using Stack Functions
# Stacking is same as concatenation, the only difference is that stacking is done along a n

a3 = np.array([[1, 2], [3, 4]])

a4 = np.array([[5, 6], [7, 8]])

aa = np.stack((a3, a4), axis=0)
aa
```

Out[6]:

```
array([[[1, 2],
        [3, 4]],

       [[5, 6],
        [7, 8]]])
```

In [7]:

```
a3 = np.array([[1, 2], [3, 4]])

a4 = np.array([[5, 6], [7, 8]])

aa = np.stack((a3, a4), axis=1)
aa
```

Out[7]:

```
array([[[1, 2],
        [5, 6]],

       [[3, 4],
        [7, 8]]])
```

In [8]:

```
a3 = np.array([[1, 2], [3, 4]])

a4 = np.array([[5, 6], [7, 8]])

aa = np.stack((a3, a4))
aa
```

Out[8]:

```
array([[[1, 2],
        [3, 4]],

       [[5, 6],
        [7, 8]]])
```

In [ ]:

```

```

In [ ]:

```
# hstack() to stack along rows.
```

In [9]:

```
a3 = np.array([[1, 2, 3, 4]])

a4 = np.array([[5, 6, 7, 8]])

aa = np.hstack((a3, a4))
aa
```

Out[9]:

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

In [ ]:

```

```

In [10]:

```
a3 = np.array([[1, 2], [3, 4]])

a4 = np.array([[5, 6], [7, 8]])

aa = np.hstack((a3, a4))
aa
```

Out[10]:

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

In [ ]:

```
```

In [ ]:

```
# vstack()  to stack along columns.
```

In [11]:

```
a3 = np.array([[1, 2, 3, 4]])

a4 = np.array([[5, 6, 7, 8]])

aa = np.vstack((a3, a4))
aa
```

Out[11]:

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

In [12]:

```
a3 = np.array([[1, 2], [3, 4]])

a4 = np.array([[5, 6], [7, 8]])

aa = np.vstack((a3, a4))
aa
```

Out[12]:

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

In [ ]:

```
```

In [ ]:

```
```

In [ ]:

```
# dstack() to stack along height, which is the same as depth.
```

In [13]:

```
a3 = np.array([[1, 2, 3, 4]])

a4 = np.array([[5, 6, 7, 8]])

aa = np.dstack((a3, a4))
aa
```

Out[13]:

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

In [14]:

```
a3 = np.array([[1, 2], [3, 4]])

a4 = np.array([[5, 6], [7, 8]])

aa = np.dstack((a3, a4))
aa
```

Out[14]:

```
array([[[1, 5],
        [2, 6]],

       [[3, 7],
        [4, 8]]])
```

In [ ]:

```

```

In [ ]:

```

```