```
print("Welcome to Numpy-5")
```

```
Welcome to Numpy-5
```

```
import numpy as np
```

```
d=np.arange(1,31).reshape((6,5))
d
```

```
array([[ 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]])
```

```
np.split(d,3,axis=0)
```

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

```
np.split(d,2,axis=0)
```

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

```
d
```

```
array([[ 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]])
```

```
np.split(d,[3,4,5],axis=0)
```

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

```
np.split(d,[1,4],axis=0)
```

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

```
d=np.arange(1,25).reshape((6,4))
d
```

```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12],
       [13, 14, 15, 16],
       [17, 18, 19, 20],
       [21, 22, 23, 24]])
```

```
np.vsplit(d,3)
```

```
[array([[1, 2, 3, 4],
        [5, 6, 7, 8]]),
 array([[ 9, 10, 11, 12],
        [13, 14, 15, 16]]),
 array([[17, 18, 19, 20],
        [21, 22, 23, 24]])]
```

```
np.hsplit(d,2)
```

```
[array([[ 1,  2],
        [ 5,  6],
        [ 9, 10],
        [13, 14],
        [17, 18],
        [21, 22]]),
 array([[ 3,  4],
        [ 7,  8],
        [11, 12],
        [15, 16],
        [19, 20],
        [23, 24]])]
```

```
d
```

```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12],
       [13, 14, 15, 16],
       [17, 18, 19, 20],
       [21, 22, 23, 24]])
```

```
np.hsplit(d,[1,2])
```

```
[array([[ 1],
        [ 5],
        [ 9],
        [13],
        [17],
        [21]]),
 array([[ 2],
        [ 6],
        [10],
        [14],
        [18],
        [22]]),
 array([[ 3,  4],
        [ 7,  8],
        [11, 12],
        [15, 16],
        [19, 20],
        [23, 24]])]
```

```
np.vsplit(d,[2,4,5])
```

```
[array([[1, 2, 3, 4],
        [5, 6, 7, 8]]),
 array([[ 9, 10, 11, 12],
        [13, 14, 15, 16]]),
 array([[17, 18, 19, 20]]),
 array([[21, 22, 23, 24]])]
```

```
#concatenate
```

```
d=np.arange(1,5)
e=np.arange(2,6)
print(d)
print(e)
```

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

```
np.concatenate([d,e])
```

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

```
np.concatenate([d,e],axis=0)
```

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

```
# np.concatenate([d,e],axis=1) #error
```

```
d=np.arange(1,5)
e=np.arange(2,10)
print(d)
print(e)
```

```
[1 2 3 4]
[2 3 4 5 6 7 8 9]
```

```
np.concatenate([d,e])
```

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

```
#example -2

d=np.arange(1,13).reshape((3,4))
e=np.arange(21,33).reshape((3,4))
print(d)
print(e)
```

```
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
[[21 22 23 24]
 [25 26 27 28]
 [29 30 31 32]]
```

```
np.concatenate([d,e])
```

```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12],
       [21, 22, 23, 24],
       [25, 26, 27, 28],
       [29, 30, 31, 32]])
```

```
np.concatenate([d,e],axis=0)
```

```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12],
       [21, 22, 23, 24],
       [25, 26, 27, 28],
       [29, 30, 31, 32]])
```

```
np.concatenate([d,e],axis=1)
```

```
array([[ 1,  2,  3,  4, 21, 22, 23, 24],
       [ 5,  6,  7,  8, 25, 26, 27, 28],
       [ 9, 10, 11, 12, 29, 30, 31, 32]])
```

```
# 2d with 1d
```

```
d=np.arange(1,13).reshape((3,4))
e=np.arange(1,5)
print(d)
print(e)
```

```
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
[1 2 3 4]
```

```
# np.concatenate([d,e])
```

```
np.concatenate([d,e],axis=0)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_87736/236814901.py in <mo
----> 1 np.concatenate([d,e],axis=0)

<__array_function__ internals> in concatenate(*args, **kwargs)

ValueError: all the input arrays must have same number of dimensions, but the array a
array at index 1 has 1 dimension(s)
```

SEARCH STACK OVERFLOW

```
# np.concatenate([e,d],axis=0)
```

```
d=np.arange(1,13).reshape((3,4))
e=np.arange(1,5).reshape((1,4))
print(d)
print(e)
```

```
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
[[1 2 3 4]]
```

```
np.concatenate([d,e])
```

```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12],
       [ 1,  2,  3,  4]])
```

```
np.concatenate([d,e],axis=0)
```

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

```
       [ 9, 10, 11, 12],
       [ 1,  2,  3,  4]])
```

```
np.concatenate([d,e],axis=1)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_87736/1668449446.py in <m
----> 1 np.concatenate([d,e],axis=1)

<__array_function__ internals> in concatenate(*args, **kwargs)

ValueError: all the input array dimensions for the concatenation axis must match exac
at index 0 has size 3 and the array at index 1 has size 1
```

SEARCH STACK OVERFLOW

```
np.concatenate([d,e.T],axis=1)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_87736/1122353861.py in <m
----> 1 np.concatenate([d,e.T],axis=1)

<__array_function__ internals> in concatenate(*args, **kwargs)

ValueError: all the input array dimensions for the concatenation axis must match exac
at index 0 has size 3 and the array at index 1 has size 4
```

SEARCH STACK OVERFLOW

```
#hastack and vstack
```

```
d=np.arange(1,5)
e=np.arange(2,6)
print(d)
print(e)
```

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

```
np.vstack([d,e])
```

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

```
np.hstack([d,e])
```

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

```
#2d array with 1 row
d=np.arange(1,5).reshape((1,4))
e=np.arange(2,6).reshape((1,4))
print(d)
print(e)
```

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

```
np.hstack([d,e])
```

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

```
np.vstack([d,e])
```

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

```
#2d array with 1 row
d=np.arange(1,13).reshape((3,4))
e=np.arange(2,14).reshape((3,4))
print(d)
print(e)
```

```
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
[[ 2  3  4  5]
 [ 6  7  8  9]
 [10 11 12 13]]
```

```
np.vstack([d,e])
```

```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12],
       [ 2,  3,  4,  5],
       [ 6,  7,  8,  9],
       [10, 11, 12, 13]])
```

```
np.hstack([d,e])
```

```
array([[ 1,  2,  3,  4,  2,  3,  4,  5],
       [ 5,  6,  7,  8,  6,  7,  8,  9],
       [ 9, 10, 11, 12, 10, 11, 12, 13]])
```

```
#2d with 1d
```

```
d=np.arange(1,13).reshape((3,4))
e=np.arange(1,5)
print(d)
print(e)
```

```
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
[1 2 3 4]
```

```
np.vstack([d,e])
```

```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12],
       [ 1,  2,  3,  4]])
```

```
np.hstack([d,e])
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_87736/1454330102.py in <m
----> 1 np.hstack([d,e])

<__array_function__ internals> in hstack(*args, **kwargs)

~/opt/anaconda3/lib/python3.9/site-packages/numpy/core/shape_base.py in hstack(tup)
    344         return _nx.concatenate(arrs, 0)
    345     else:
--> 346         return _nx.concatenate(arrs, 1)
    347
    348

<__array_function__ internals> in concatenate(*args, **kwargs)

ValueError: all the input arrays must have same number of dimensions, but the array a
array at index 1 has 1 dimension(s)
```

SEARCH STACK OVERFLOW

```
# 2d with 2d but different shapes
```

```
d=np.arange(1,13).reshape((3,4))
e=np.arange(1,5).reshape((1,4))
print(d)
print(e)
```

```
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
[[1 2 3 4]]
```

```
np.hstack([d,e])
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_87736/1454330102.py in <m
----> 1 np.hstack([d,e])

<__array_function__ internals> in hstack(*args, **kwargs)

~/opt/anaconda3/lib/python3.9/site-packages/numpy/core/shape_base.py in hstack(tup)
    344         return _nx.concatenate(arrs, 0)
    345     else:
--> 346         return _nx.concatenate(arrs, 1)
    347
    348

<__array_function__ internals> in concatenate(*args, **kwargs)

ValueError: all the input array dimensions for the concatenation axis must match exac
at index 0 has size 3 and the array at index 1 has size 1
```

SEARCH STACK OVERFLOW

```
np.vstack([d,e])
```

```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12],
       [ 1,  2,  3,  4]])
```

```
#Broadcasting
```

Double-click (or enter) to edit

```
d=np.arange(1,5)
e=np.arange(2,6)
print(d)
print(e)
d+e
```

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

```
d=np.arange(1,5)
e=np.arange(2,7)
print(d)
print(e)
```

```
[1 2 3 4]
[2 3 4 5 6]
```

```
d+e
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_87736/1872672908.py in <m
----> 1 d+e

ValueError: operands could not be broadcast together with shapes (4,) (5,)
```

SEARCH STACK OVERFLOW

```
d
```

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

```
d+2
```

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

```
x=np.arange(1,7).reshape((3,2))
y=np.arange(1,3)
z=np.arange(1,4)
print(x)
print(x.shape)
print(x.ndim)
print("-"*50)
print(y)
print(y.shape)
print(y.ndim)
print("-"*50)
print(z)
print(z.shape)
print(z.ndim)
```

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

```
x+y
```

```
array([[2, 4],
       [4, 6],
       [6, 8]])
```

x+z

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_87736/2335943885.py in <m
----> 1 x+z

ValueError: operands could not be broadcast together with shapes (3,2) (3,)
```

SEARCH STACK OVERFLOW

```
a=np.array([10,20,30,40])
b=np.array([1,2,3])
print(a.shape)
print(b.shape)
print(a.ndim)
print(b.ndim)
```

```
(4,)
(3,)
1
1
```

a+b

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_87736/3553919051.py in <m
----> 1 a+b

ValueError: operands could not be broadcast together with shapes (4,) (3,)
```

SEARCH STACK OVERFLOW

```
a= np.arange(1,7).reshape((3,2))
b= np.array([10,20])

print(a)
print(a.shape)
print(a.ndim)
print("-"*50)
print(b)
print(b.shape)
print(b.ndim)
```

```
[[1 2]
 [3 4]
 [5 6]]
```

```
(3, 2)
2
--------------------------------------------------
[10 20]
(2,)
1
```

```
a+b
```

```
array([[11, 22],
       [13, 24],
       [15, 26]])
```

```
c=np.array([4,5,6])
print(c)
print(c.shape)
```

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

```
a+c
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_87736/3032047929.py in <m
----> 1 a+c

ValueError: operands could not be broadcast together with shapes (3,2) (3,)
```

SEARCH STACK OVERFLOW

Broadcasting

Rules

For each dimension (going from right side)

1. The size of each dimension should be same OR

2. The size of one dimension should be 1

Rule 1 : If two array differ in the number of dimensions, the shape of one with fewer dimensions is padded with ones on its leading(Left Side).

Rule 2 : If the shape of two arrays doesnt match in any dimensions, the array with shape equal to 1 is stretched to match the other shape.

Rule 3 : If in any dimesion the sizes disagree and neither equal to 1 , then Error is raised.

[ ] ↳ *36 cells hidden*