

```
import numpy as np
a=np.arange(10,20,1)
print(a)
print(a[0:10])
print(a[:])
print(a[0:])
print(a[:10])
```

[10 11 12 13 14 15 16 17 18 19]  
[10 11 12 13 14 15 16 17 18 19]  
[10 11 12 13 14 15 16 17 18 19]  
[10 11 12 13 14 15 16 17 18 19]  
[10 11 12 13 14 15 16 17 18 19]

```
import numpy as np      [10 11 12 13 14 15 16 17 18 19]
a=np.arange(10,20,1)  [10 11 12 13 14]
print(a)
print(a[0:5])
print(a[:5])
```



```
import numpy as np      [10 11 12 13 14 15 16 17 18 19]
a=np.arange(10,20,1)  [15 16 17 18 19]
print(a)              [15 16 17 18 19]
print(a[-5:10])
print(a[-5:])
```

```
import numpy as np      [10 11 12 13 14 15 16 17 18 19]
a=np.arange(10,20,1)  [19 18 17 16 15 14 13 12 11 10]
print(a)              [19 18 17 16 15 14 13 12 11 10]
print(a[-1:-11:-1])
print(a[-1::-1])
```

```
import numpy as np      [10 11 12 13 14 15 16 17 18 19]
a=np.arange(10,20,1)  [14 13 12 11 10]
print(a)               [14 13 12 11 10]
print(a[4::-1])       [14 13 12 11 10]
print(a[4:-11:-1])    [14 13 12 11 10]
print(a[-6::-1])
print(a[-6:-11:-1])
```

```
import numpy as np      [10 11 12 13 14 15 16 17 18 19]
a=np.arange(10,20,1)  [19 18 17 16 15]
print(a)               [19 18 17 16 15]
print(a[-1:-6:-1])    [19 18 17 16 15]
print(a[:-6:-1])      [19 18 17 16 15]
print(a[9:4:-1])       [19 18 17 16 15]
print(a[-1:4:-1])     [19 18 17 16 15]
print(a[:4:-1])
print(a[9:-6:-1])
```

---

```
import numpy as np
a = np.array([[1,2,3],[3,4,5],[4,5,6]])
print(a)
print(a[0])
print(a[1:3:1])
print(a[1:])
print(a[1:3])
```

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

---

```
import numpy as np
a = np.array([[1,2,3],[3,4,5],[4,5,6]])
print(a)
print(a[1,1])
print(a[1,1:3])
print(a[1,1:])
print(a[1:,2])
print(a[1:,1:])
```

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

```
import numpy as np
a = np.array([[1,2,3],[3,4,5],[4,5,6]])
print(a)
print(a[-1,-1])
print(a[-1,-1::-1])
print(a[-1::-1,-2])
print(a[-1::-1,-1::-1])
```

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

```
import numpy as np  
a=np.array([[1,2,3],[4,5,6],[7,8,9]])  
print(a[2:,2])
```

[9]

```
import numpy as np  
a=np.array([[1,2,3],[4,5,6],[7,8,9]])  
print(a[2,2:])
```

[9]

```
import numpy as np  
a=np.array([[1,2,3],[4,5,6],[7,8,9]])  
print(a[2:,2:])
```

[[9]]

```
import numpy as np  
a=np.array([[1,2,3],[4,5,6],[7,8,9]])  
print(a[1:,2:])
```

```
[[6]  
 [9]]
```

```
| # Create a one-dimensional array 'z' having elements from 1 to 100
| .
| .
| import numpy as np
| z=np.arange(1,101,1)
| print(z)
| .
```

---

```
# Write a command to print first 5 elements in z

import numpy as np
z=np.arange(1,101,1)
print(z[0:5:1])
print(z[:5])
```

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

```
# Write a command to print last 5 elements in z

import numpy as np
z=np.arange(1,101,1)
print(z[-5:]) [ 96  97  98  99 100]
print(z[-5::-1]) [ 96  97  98  99 100]
print(z[-5:100:1]) [ 96  97  98  99 100]
```

```
# Write a command to print first 5 elements of z in reverse order
```

```
import numpy as np  
z=np.arange(1,101,1)  
print(z[4::-1])  
print(z[4:-101:-1])  
print(z[-96:-101:-1])  
print(z[-96::-1])
```

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

```
# Write a command to print last 5 elements of z in reverse order
```

```
import numpy as np  
z=np.arange(1,101,1)  
print(z[-1:-6:-1])  
print(z[99:94:-1])  
print(z[-1:94:-1])  
print(z[99:-6:-1])
```

[100 99 98 97 96]
[100 99 98 97 96]
[100 99 98 97 96]
[100 99 98 97 96]

```
import numpy as np          45
x=np.arange(1,10)
print(np.sum(x[-9::]))      [10]
print(x[-9::-1]+x[-1::])    54
print(np.sum(x[-9::]+x[-9::-1])) 20
print(np.sum(x[1::2]))       [10]
print(x[-1::2]+x[0::-2])
```

```
import numpy as np                                     0
a=np.arange(10)                                      0
print(np.sum(a[-5::-1])-np.sum(a[5::-1]))          0
print(np.sum(a[:-4])-np.sum(a[:6]))                  0
print(np.sum(a[-11:-4])-np.sum(a[-11:6]))          0
print(np.sum(a[-5:])-np.sum(a[5:]))                  0
print(np.sum(a[:-6:-1])-np.sum(a[:4:-1]))          0
print(np.sum(a[9:-6:-1])-np.sum(a[-1:4:-1]))        0
```

```
import numpy as np  
a=np.arange(1,13).reshape(4,3)  
print(a)                                [[ 1  2  3]  
                                         [ 4  5  6]  
                                         [ 7  8  9]  
                                         [10 11 12]]  
print(a[-1:,-1:].shape)                   (1, 1)  
print(np.sum(a[:2,-3]))                  5  
print(a[:2,-3::-1])                     [[1]  
                                         [4]]  
print(a[:2,-3::-1].shape)                (2, 1)
```

```
import numpy as np      (10,)  
a=np.arange(1,11)  
print(a.shape)
```

---

```
import numpy as np  
x=np.array([1,3,5,6,7,9])  
print(x!=6)
```

```
[ True  True  True False  True  True]
```

```
import numpy as np  
x=np.array([1,3,5,6,7,9])  
print(x[x!=6])
```

```
[1 3 5 7 9]
```

```
import numpy as np  
x=np.array([1,3,5,6,7,9])  
print(np.all(x==6))
```

False

```
import numpy as np  
x=np.array([1,3,5,6,7,9])  
print(np.any(x<6))
```

True

```
#Write a command to print all odd indexed elements of x  
import numpy as np  
x=np.array([1,3,5,6,7,9])  
print(x[1::2])
```

[3 6 9]

#Write a command to print the elements between 3 and 7 in x

```
import numpy as np
x=np.array([1,3,5,6,7,9])
print(x>3)
print(x<7)
print((x>3) & (x<7))
print(x[(x>3) & (x<7)])
```

[False False True True True True]  
[ True True True True False False]  
[False False True True False False]  
[5 6]

```
import numpy as np  
height=np.array([189, 170, 189, 163, 183, 171, 185,  
168, 173, 183, 173, 173, 175, 178, 183, 193, 178, 173,  
174, 183, 183, 168, 170, 178, 182, 180, 183, 178, 182,  
188, 175, 179, 183, 193, 182, 183, 177, 185, 188, 188, 182, 185])
```

```
#Write a command to find the total number of elements in the array heights
```

```
print(len(heights))
```

42

#Write a command to find the range of the array heights

```
print(max(heights)-min(heights))
```

```
#Write a command to find the variance of the array heights
```

```
print(np.var(heights))
```

```
48.05045351473922
```

```
#Write a command to find the 50th percentile of the array heights
```

```
print(np.percentile(heights,50))
```

```
182.0
```

```
#Write a command to find the product of all the elements of the array heights  
print(np.prod(heights))
```

1501594074476445696

```
import numpy as np
a=np.arange(10,18,1).reshape(4,2)
print(a)
print(a.reshape(2,4))
print(a.T)
```

[[10 11]  
 [12 13]  
 [14 15]  
 [16 17]]  
 [[10 11 12 13]  
 [14 15 16 17]]  
 [[10 12 14 16]  
 [11 13 15 17]]

```
import numpy as np  
a=np.arange(1,13).reshape(4,3)  
print(a)  
print(a.T)  
print(a.reshape(3,4))  
print(np.sum(a.T-a.reshape(3,4)))
```

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

```
import numpy as np  
a=np.array([20,30,40,50])  
b=np.arange(4)  
c=a-b  
print(max(c)-min(c))
```

```
import numpy as np  
a = np.array([12,23,34,56,67,78,89])      3  
print(len(np.arange(10,25,5)))
```

```
import numpy as np  
a = np.array([12,23,34,56,67,78,89])  
print(a.shape)                                (7,)
```

```
import numpy as np          [[0]
a=np.array([[1,2,3]])        [0]
b=a.reshape(3,1)            [0]]
c=a.T
print(c-b)
```

---

```
import numpy as np
a=np.array([[1,2,3],[4,5,6]])
b=a.reshape(3,2)
c=a.T
print(np.sum(c-b))
print(np.sum(c-b)==np.sum(b-c))
```

0

True

```
import numpy as np  
print(np.array([1,2,3],dtype=float))  
print(np.array([1,2,3],dtype=complex))
```

```
[1. 2. 3.]  
[1.+0.j 2.+0.j 3.+0.j]
```

```
import numpy as np
a=np.arange(1,25).reshape(2,4,3)
print(a)
print(a[0].shape)
```

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

```
import numpy as np
a=np.arange(1,25).reshape(2,4,3)
print(a)
print(np.sum(a[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]]]  
78

```
import numpy as np
```

```
a=np.arange(1,25).reshape(2,4,3)
```

```
print(a)
```

```
print(a[0,3,2]-a[1,0,0])
```

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

```
[[13 14 15]
```

```
[16 17 18]
```

```
[19 20 21]
```

```
[22 23 24]]]
```

```
import numpy as np
```

```
a=np.arange(1,25).reshape(2,4,3)
```

```
print(a)
```

```
print(a[-2,-1,-1]-a[-1,-4,-3])
```

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

```
[[13 14 15]
```

```
[16 17 18]
```

```
[19 20 21]
```

```
[22 23 24]]]
```

-1

```
import numpy as np
a=np.arange(1,25).reshape(2,4,3)
print(a)
print(a[-2,-3::-1,-2:])
```

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

```
import numpy as np
a=np.arange(1,25).reshape(2,4,3)
print(a)
print(a[-2::-1,-3::-1,-2:])
```

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

```
import numpy as np
a=np.arange(1,25).reshape(2,4,3)
print(a)
print(a[-2::-1,-4::-1,-1:])
.
[[[ 1  2  3]
 [ 4  5  6]
 [ 7  8  9]
 [10 11 12]]]
[[[13 14 15]
 [16 17 18]
 [19 20 21]
 [22 23 24]]]
[[[3]]]
```

```
import numpy as np  
a=np.arange(1,25).reshape(2,4,3)  
print(a)  
print(a[-2::-1,-4::-1,-1:].shape)
```

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

```
[[13 14 15 ]  
 [16 17 18 ]  
 [19 20 21 ]  
 [22 23 24 ]]]  
(1, 1, 1)
```

```
import numpy as np
a=np.arange(1,19).reshape(2,3,3)
print(a)
print(a[0,2,0])
print(a[0,2,0].shape)
print(a[0,2,:])
print(a[0,2,:].shape)
print(a[0,2:,:])
print(a[0,2:,:].shape)
print(a[0::-1,:,:])
print(a[0::-1,:,:].shape)
```

```
7
()
[7 8 9]
(3,)
[[7 8 9]]
(1, 3)
[[[7 8 9]]]
(1, 1, 3)
```

```
import numpy as np
a=np.arange(1,19).reshape(2,3,3)
print(a)
print(a[-2,-1,-3])
print(a[-2,-1,-3].shape)
print(a[-2,-1,:])
print(a[-2,-1,:].shape)
print(a[-2,-1:,:,])
print(a[-2,-1:,:,].shape)
print(a[-2::-1,-1:,:,])
print(a[-2::-1,-1:,:].shape)
```

7  
()  
[7 8 9]  
(3,)  
[[7 8 9]]  
(1, 3)  
[[[7 8 9]]]  
(1, 1, 3)

```
import numpy as np  
a=np.arange(1,19).reshape(2,3,3)  
print(a)  
print(np.sum(a[:,0,:]))  
print(np.sum(a[:, -3, :]))  
print(np.sum(a[-2, 0:2, 1:]))  
print(np.sum(a[-2:, -3:, -1::-1]))
```

39

39

16

171

```
import numpy as np  
a=np.arange(1,19).reshape(2,3,3)  
print(a)  
print(a[:,0,:][-2,-1])  
print(a[:, -3, :][-2])  
print(a[-2,0:2,1:][-2::-1])  
print(a[-2:,-3:,-1::-1][:,:,:-3])
```

3  
[1 2 3]  
[[2 3]]  
[[ 3 6 9]  
 [12 15 18]]

```
import numpy as np
a=np.arange(1,19).reshape(2,3,3)
print(a[-2:,-3:,-1::-1][:,:,,-3::-1])
print(a[-2:,-3:,-1::-1][:,:,,-3::-1].shape)
```

[[[[ 3]  
[ 6]  
[ 9]]  
[[12]  
[15]  
[18]]]  
(2, 3, 1)

Create a 3x5 array having all elements 3.14

```
print(np.full((3,5),3.14))
```

```
[[3.14 3.14 3.14 3.14 3.14]
 [3.14 3.14 3.14 3.14 3.14]
 [3.14 3.14 3.14 3.14 3.14]]
```

**Create a diagonal matrix having elements [3,2,4,5]**

```
print(np.diag([3,2,4,5]))
```

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

**Create a 3-dimensional array with the elements [1,2,3,4,5,6,7,8]**

```
print(np.array([1,2,3,4,5,6,7,8]).reshape(2,2,2))
```

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

```
[[5 6]  
 [7 8]]]
```

**What is the output of the command: np.arange(1,10,3.0)**

[1. 4. 7.]

**What is the output of the command: np.linspace(1,6,5)**

[1. 2.25 3.5 4.75 6.]

```
import numpy as np [11 12 13 14 15]
import pandas as pd [11 12 13 14 15]
a=np.arange(11,16,1) [11 12 13 14 15]
print(a)
print(a[0:5])
print(a[0:50])
```

```
import numpy as np
import pandas as pd
print(np.linspace(2,3,5))
```

[2. 2.25 2.5 2.75 3.]

```
import numpy as np  
import pandas as pd  
print(len(np.arange(1,20,4.0)))
```

---

```
import numpy as np
print(np.eye(4))
print(np.eye(4, k=-1))
print(np.eye(4, k=1))
```

```
[[1.  0.  0.  0.]
 [0.  1.  0.  0.]
 [0.  0.  1.  0.]
 [0.  0.  0.  1.]]
 [[0.  0.  0.  0.]
 [1.  0.  0.  0.]
 [0.  1.  0.  0.]
 [0.  0.  1.  0.]]
 [[0.  1.  0.  0.]
 [0.  0.  1.  0.]
 [0.  0.  0.  1.]
 [0.  0.  0.  0.]]
```

```
import numpy as np                                int64
import pandas as pd                               (4, 3)
a=np.arange(1,13).reshape(4,3)
print(a.dtype)
print(a.shape)
```

```
import numpy as np  
a=np.full((3,3),10)  
print(np.sum(a[:-1]))
```

60

```
import numpy as np
a=np.zeros((3,3),dtype=int)
a[-1:-2:-1,-2:3]=1
print(a)
print(np.sum(a))
```

```
[[0 0 0]
 [0 0 0]
 [0 1 1]]
```

2

```
import numpy as np
A = np.arange(9).reshape(3,3)
print(np.sum(A[[0,1],:]))
```

15

```
import numpy as np          [1.      0.25  0.5   0.75  1.     ]  
arr=np.linspace(0,1,5)     3.5  
arr[np.argmin(arr)]=1  
print(arr)  
print(np.sum(arr))
```

---

```
import numpy as np
a=np.array([1,2,3,4])
b=np.array([5,6,7,8])
print(np.concatenate((a,b)))
```

[1 2 3 4 5 6 7 8]

```
import numpy as np  
a=np.array([1,2,3,4])  
b=np.array([5,6,7,8])  
print(np.concatenate((a,b),axis=-1))
```

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

```
import numpy as np  
a=np.array([[1,2],[3,4]])  
b=np.array([[5,6],[7,8]])  
print(a)  
print(b)  
print(np.concatenate((a,b),axis=0))  
print(np.concatenate((a,b),axis=1))
```

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

```
import numpy as np  
a=np.array([[1,2],[3,4]])  
b=np.array([[5,6],[7,8]])  
print(np.concatenate((a,b),axis=None))
```

[1 2 3 4 5 6 7 8]

```
import numpy as np
x=np.arange(9)
print(np.split(x,1))
print(np.split(x,3))
[array([0, 1, 2, 3, 4, 5, 6, 7, 8])]
[array([0, 1, 2]), array([3, 4, 5]), array([6, 7, 8])]
```

```
import numpy as np
x=np.arange(10,18)
print(x)
print(np.split(x,[3,5]))
[10 11 12 13 14 15 16 17]
[array([10, 11, 12]), array([13, 14]), array([15, 16, 17])]
```

```
import numpy as np
x=np.arange(10,18)
print(x)
print(np.split(x,[2,4,6]))
[10 11 12 13 14 15 16 17]
[array([10, 11]), array([12, 13]), array([14, 15]), array([16, 17])]
```

```
import numpy as np  
a=np.arange(16).reshape(4,4)  
print(a)  
print(np.hsplit(a,2))
```

```
[[ 0  1  2  3]  
 [ 4  5  6  7]  
 [ 8  9 10 11]  
 [12 13 14 15]]  
[array([[ 0,  1],  
       [ 4,  5],  
       [ 8,  9],  
       [12, 13]]), array([[ 2,  3],  
       [ 6,  7],  
       [10, 11],  
       [14, 15]])]
```

```
import numpy as np
a=np.arange(16).reshape(4,4)
print(a)
print(np.vsplit(a,2))
```

[[ 0 1 2 3]  
 [ 4 5 6 7]  
 [ 8 9 10 11]  
 [12 13 14 15]]  
[array([[0, 1, 2, 3],  
 [4, 5, 6, 7]]), array([[ 8, 9, 10, 11],  
 [12, 13, 14, 15]])]

```
import numpy as np
a=np.arange(16).reshape(4,4)
print(np.hsplit(a,4))
[ array([[ 0],
       [ 4],
       [ 8],
       [12]]), array([[ 1],
       [ 5],
       [ 9],
       [13]]), array([[ 2],
       [ 6],
       [10],
       [14]]), array([[ 3],
       [ 7],
       [11],
       [15]])]
```

```
import numpy as np
a=np.arange(16).reshape(4,4)
print(np.vsplit(a,4))
[ array([[0, 1, 2, 3]]), array([[4, 5, 6, 7]]), array([[ 8,  9, 10, 11]]), array([[12, 13, 14, 15]])]
```

```
import numpy as np  
y=np.arange(1,11).reshape(2,5)  
print(y)  
print(np.hsplit(y,[4]))
```

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

```
import numpy as np  
y=np.arange(1,11).reshape(2,5)  
print(y)  
print(np.hsplit(y,[1,4]))
```

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

```
import numpy as np
y=np.arange(1,11).reshape(2,5)
print(y)
print(np.vsplit(y,[2]))
[[ 1  2  3  4  5]
 [ 6  7  8  9 10]]
[array([[ 1,  2,  3,  4,  5],
       [ 6,  7,  8,  9, 10]]), array([], shape=(0, 5), dtype=int64)]
```

```
import numpy as np  
y=np.arange(1,11).reshape(2,5)  
print(y)  
print(np.split(y,[2], axis=1) )
```

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

```
import numpy as np
y=np.arange(1,11,1).reshape(2,5)
z=np.split(y,2)
print(z)
print(z[0].shape)
print(z[1].shape)
z1=np.split(y,[2])
print(z1)
print(z1[0].shape)
print(z1[1].shape)
```

[array([[1, 2, 3, 4, 5]]), array([[ 6, 7, 8, 9, 10]])]  
(1, 5)  
(1, 5)  
[array([[ 1, 2, 3, 4, 5],  
 [ 6, 7, 8, 9, 10]]), array([], shape=(0, 5), dtype=int64)]  
(2, 5)  
(0, 5)

```
import numpy as np
a=np.arange(1,11).reshape(2,5)
b=np.split(a,2)                                [[0 0 0 0 0]]
c=np.split(a,[1])                               (1, 5)
d=b[1]-c[1]
print(d)
print(d.shape)
.
```

```
import numpy as np      [array([10])]  
a=np.array([10])  
b=np.split(a,1)  
print(b)
```

```
import numpy as np
a=np.array([10])
b=np.split(a,[1])
print(b)
[ array([10]), array([], dtype=int64)]
```

```
import numpy as np
a=np.array([[10]])
b=np.split(a,1)
c=np.split(a,1,axis=0)
d=np.vsplit(a,1)
print(b)
print(c)
print(d)
```

```
[array([[10]])]
[array([[10]])]
[array([[10]])]
```

---

```
import numpy as np
a=np.array([[10]])
b=np.split(a,[1])
c=np.split(a,[1],axis=0)
d=np.vsplit(a,[1])
print(b)
print(c)
print(d)
```

[array([[10]]), array([], shape=(0, 1), dtype=int64)]  
[array([[10]]), array([], shape=(0, 1), dtype=int64)]  
[array([[10]]), array([], shape=(0, 1), dtype=int64)]

```
import numpy as np
a=np.array([[10]])
#b=np.split(a,1)
c=np.split(a,1, axis=1)
d=np.hsplit(a,1)
#print(b)
print(c)
print(d)
```

```
[array([[10]])]
[array([[10]])]
```

```
import numpy as np      [array([[10]]), array([], shape=(1, 0), dtype=int64)]
a=np.array([[10]])       [array([[10]]), array([], shape=(1, 0), dtype=int64)]
#b=np.split(a,1)
c=np.split(a,[1],axis=1)
d=np.hsplit(a,[1])
#print(b)
print(c)
print(d)
```

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
 [[ 1   2   3]
 [ 4   5   6]
 [ 7   8   9]
 [12  15  18]]
```

---

```
import numpy as np
import pandas as pd
a=np.arange(1,10).reshape(3,3)
print(a)
b=np.sum(a, axis=0).reshape(1,3)
a=np.concatenate([a,b], axis=0)
print(a)
```

```
import numpy as np  
import pandas as pd  
a=np.arange(1,10).reshape(3,3)  
print(a)  
b=np.sum(a, axis=1).reshape(3,1)  
a=np.concatenate([a,b], axis=1)  
print(a)
```

	[[1 2 3]
	[4 5 6]
	[7 8 9]]
	[[ 1 2 3 6]
	[ 4 5 6 15]
	[ 7 8 9 24]]

```
import numpy as np  
a=np.array([[1,2],[3,4]])  
b=np.array([[5,6],[7,8]])  
print(np.stack((a,b),axis=1))
```

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

```
import numpy as np  
a=np.array([[1,2],[5,6]])  
b=np.array([[3,4],[7,8]])  
c=np.stack((a,b),axis=1)  
print(np.sum(c[0]))
```

10

```
import numpy as np  
a=np.array([[1,2],[3,4]])  
b=np.array([[5,6],[7,8]])  
print(np.stack((a,b),axis=0))
```

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

```
[[[1 5]
  [2 6]]
import numpy as np
a=np.array([[1,2],[3,4]])
b=np.array([[5,6],[7,8]])
print(np.stack((a,b),axis=2))
[[[3 7]
  [4 8]]]
```

```
import numpy as np  
a=np.array([[1,2],[3,4]])  
b=np.array([[5,6],[7,8]])  
c=np.dstack((a,b))  
print(c)
```

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

```
import numpy as np
a=np.array([[1,2],[3,4]])
b=np.array([[5,6],[7,8]])
print(np.stack((a,b),axis=-1))
```

```
[[[1 5]
 [2 6]]
```

```
[[3 7]
 [4 8]]]
```

```
import numpy as np  
a=np.array([[1,2],[3,4]])  
b=np.array([[5,6],[7,8]])  
print(np.hstack((a,b)))  
print(np.concatenate((a,b),axis=1))
```

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

```
import numpy as np  
a=np.array([[1,2],[3,4]])  
b=np.array([[5,6],[7,8]])  
print(np.vstack((a,b)))  
print(np.concatenate((a,b),axis=0))
```

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

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

```
import numpy as np  
x = np.array([1, 2, 3])  
y=np.array([[9, 8, 7], [6, 5, 4]])  
print(np.vstack((x,y)))
```

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

```
import numpy as np  
x = np.array([[9, 8, 7], [6, 5, 4]])  
y= np.array([[99], [99]])  
print(np.hstack((x,y)))
```

```
[[ 9  8  7 99]  
 [ 6  5  4 99]]
```

```
import numpy as np
x = np.array((1,2,3))           [1 2 3 4 5 6]
y= np.array((4,5,6))           [[1 2 3]
print(np.hstack((x,y)))        [4 5 6]]
print(np.vstack((x,y)))
.
```

---

```
import numpy as np
a=np.array([1,2,3])
b=np.array([4,5,6])
print(np.concatenate((a,b)))
print(np.concatenate((a,b),axis=0))
print(np.concatenate((a,b),axis=-1))
print(np.hstack((a,b)))
```

[1 2 3 4 5 6]

[1 2 3 4 5 6]

[1 2 3 4 5 6]

[1 2 3 4 5 6]

```
import numpy as np
x = np.array((1,2,3))
y= np.array((4,5,6))
print(np.stack((x,y),axis=0))
print(np.stack((x,y),axis=1))
print(np.stack((x,y),axis=-1))
```

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

```
import numpy as np  
a=np.array([[1,2,3]])  
b=np.array([[4,5,6]])  
x=np.stack((a,b),axis=0)  
y=np.stack((a,b),axis=1)  
print(x)  
print(y)  
print(x.shape)  
print(y.shape)
```

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

```
import numpy as np  
a=np.array([[1,2,3]])  
b=np.array([[4,5,6]])  
x=np.stack((a,b),axis=2)  
y=np.stack((a,b),axis=-1)  
print(x)  
print(y)  
print(x.shape)  
print(y.shape)
```

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

```
import numpy as np  
a=np.array([[1,2,3],[2,3,4]])  
b=np.array([[4,5,6],[5,6,7]])  
x=np.stack((a,b),axis=2)  
print(x)  
print(np.sum(x[0]))
```

[[[1 4]  
 [2 5]  
 [3 6]]  
 [[2 5]  
 [3 6]  
 [4 7]]]  
21

```
import numpy as np
y=np.arange(1,11).reshape(2,5)
print(y)
print(np.vsplit(y,[1]))
print(np.hsplit(y,[1,4]))
print(np.hstack((y,y)))
print(np.vstack((y,y)))
```

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

```
import numpy as np          -99
a=np.array([1,2,3])
b=np.array([100])
c=np.hstack((a,b))
print(min(c)-max(c))
```

```
import numpy as np          (2, 3)
a=np.array([1,2,3])
b=np.array([100,101,102])
c=np.vstack((a,b))
print(c.shape)
```

```
import numpy as np          (3, 2)
a=np.array([1,2,3])
b=np.array([100,101,102])
c=np.stack((a,b),axis=-1)
print(c.shape)
```

```
import numpy as np          (1, 3, 2)
a=np.array([[1,2,3]])
b=np.array([[100,101,102]])
c=np.stack((a,b),axis=-1)
print(c.shape)
```

```
import numpy as np          (1, 2, 3)
a=np.array([[1,2,3]])
b=np.array([[100,101,102]])
c=np.stack((a,b),axis=1)
print(c.shape)
```

```
import numpy as np          0.0
a=np.array([[1.0,2.0,3.0],[4.0,5.0,6.0]])
b=np.array([[],[]])
c=np.concatenate((a,b),axis=1)
d=np.hstack((a,b))
print(np.sum(c-d))
```

```
import numpy as np
a=np.array([[1.0,2.0],[4.0,5.0]])
b=np.array([[0.,0.],[0.,0.]])
c=np.concatenate((a,b),axis=0)
d=np.vstack((a,b))
print(np.sum(c-d)==np.sum(d-c))
```

True

```
import numpy as np          0.0
a=np.array([[1.0,2.0],[4.0,5.0]])
b=np.array([[0.,0.],[0.,0.]])
c=np.stack((a,b),axis=2)
d=np.stack((a,b),axis=-1)
print(np.sum(c-d)-np.sum(d-c))
```

```
import numpy as np          0.0
a=np.array([1.0,2.0])
b=np.array([0.,0.])
c=np.stack((a,b),axis=1)
d=np.stack((a,b),axis=-1)
print(np.sum(c-d)-np.sum(d-c))
```

```
import numpy as np
a=np.array([1.0,2.0])
b=np.array([0.,0.])
c=np.stack((a,b),axis=1)
d=np.vstack((a,b))
print(c.shape==d.shape)
```

True

```
import numpy as np
a=np.array([1.0,2.0])
b=np.array([0.,0.])
c=np.stack((a,b),axis=1)
d=np.vstack((a,b))
print(c-d)
```

```
import numpy as np
a=np.array([1,2,3],ndmin=3)
print(a)                      [[[1 2 3]]]
print(a.shape)                (1, 1, 3)
```

---

```
import numpy as np [1 None 3 4]
import pandas as pd
a= np.array([1, None, 3, 4])
print(a)
print("\n")
print(a.dtype)
```

object

```
import numpy as np  
import pandas as pd  
a= np.array([1, None, 3, 4])  
print(np.sum(a))
```

TypeError: unsupported operand type(s) for +: 'int' and 'NoneType'

```
import numpy as np [ 1. nan  3.  4.]
import pandas as pd
a= np.array([1, np.nan, 3, 4])
print(a)           float64
print("\n")
print(a.dtype)
```

---

```
import numpy as np
import pandas as pd
print(1 + np.nan)
print(0 * np.nan)
print(np.nan + np.nan)
```

nan  
nan  
nan

```
import numpy as np          nan
import pandas as pd         nan
a= np.array([1, np.nan, 3, 4])    nan
print(a.sum())
print(a.min())
print(a.max())
-
```

```
import numpy as np          8.0
import pandas as pd
a= np.array([1, np.nan, 3, 4])
print(np.nansum(a))        1.0
print(np.nanmin(a))
print(np.nanmax(a))
.
```

```
import numpy as np
import pandas as pd
a= pd.Series([1, np.nan, 2, None])
print(a)
```

	0	1.0
	1	NaN
	2	2.0
	3	NaN
		dtype: float64

---

```
import numpy as np
import pandas as pd
x = pd.Series(np.arange(2), dtype=int)
print(x)
x[0]=None
print(x)
```

	0	0
	1	1
		dtype: int64
	0	NaN
	1	1.0
		dtype: float64

```
import numpy as np
import pandas as pd
print(pd.Series([1,np.nan,'Hello', None]))
```

0		1
1		NaN
2		Hello
3		None
		dtype: object

```
import numpy as np
import pandas as pd
data = pd.Series([1, np.nan, 'hello', None])
print(data.isnull())
0      False
1      True
2     False
3      True
dtype: bool
```

```
import numpy as np
import pandas as pd
data = pd.Series([1, np.nan, 'hello', None])
print(data[data.isnull()])
```

	1	NaN
	3	None
		dtype: object

```
import numpy as np
import pandas as pd
data = pd.Series([1, np.nan, 'hello', None])
print(data.notnull())
print(data[data.notnull()])
```

0	True	
1	False	
2	True	
3	False	
		dtype: bool
0	1	
2	hello	
		dtype: object

```
import numpy as np
import pandas as pd
data = pd.Series([1, np.nan, 'hello', None])
print(data.dropna())
```

0	1
2	hello
	dtype: object

```
import numpy as np
import pandas as pd
data = pd.Series([1, np.nan, 'hello', None])
print(data.fillna(0))

```

	0	1
	1	0
	2	hello
	3	0
		dtype: object

```
import numpy as np
import pandas as pd
data = pd.Series([1, np.nan, 3, None])
print(data.fillna(np.nanmean(data)))
```

0	1.0
1	2.0
2	3.0
3	2.0

dtype: float64

```
import numpy as np
import pandas as pd
df = pd.DataFrame([[1, np.nan, 2],
[2, 3, 5],
[np.nan, 4, 6]])
print(df)
```

	0	1	2
0	1.0	NaN	2
1	2.0	3.0	5
2	NaN	4.0	6

```
import pandas as pd  
  
df = pd.DataFrame([[1, np.nan, 2],  
[2, 3, 5],  
[np.nan, 4, 6]])  
print(df)  
print(df.dropna())
```

	0	1	2
0	1.0	NaN	2
1	2.0	3.0	5
2	NaN	4.0	6
	0	1	2
1	2.0	3.0	5

```
import numpy as np
import pandas as pd
df = pd.DataFrame([[1, np.nan, 2],
[2, 3, 5],
[np.nan, 4, 6]])
print(df)
print(df.dropna(axis=1))
```

	0	1	2
0	1.0	NaN	2
1	2.0	3.0	5
2	NaN	4.0	6
		2	
0	2		
1	5		
2	6		

```
import numpy as np
import pandas as pd
df = pd.DataFrame([[1, np.nan, 2],
[2, 3, 5],
[np.nan, 4, 6]])
print(df)
df[3] = np.nan
print(df)
```

	0	1	2	
0	1.0	NaN	2	
1	2.0	3.0	5	
2	NaN	4.0	6	
	0	1	2	3
0	1.0	NaN	2	NaN
1	2.0	3.0	5	NaN
2	NaN	4.0	6	NaN

```
import numpy as np
import pandas as pd
df = pd.DataFrame([[1, np.nan, 2],
[2, 3, 5],
[np.nan, 4, 6]])
print(df)
print(df.dropna(axis='columns', how='all'))
```

	0	1	2
0	1.0	NaN	2
1	2.0	3.0	5
2	NaN	4.0	6
	0	1	2
0	1.0	NaN	2
1	2.0	3.0	5
2	NaN	4.0	6

```
import numpy as np
import pandas as pd
df = pd.DataFrame([[1, np.nan, 2],
[2, 3, 5],
[np.nan, 4, 6]])
print(df)
print(df.dropna(axis='columns', thresh=3))
```

	0	1	2
0	1.0	NaN	2
1	2.0	3.0	5
2	NaN	4.0	6
	2		
0	2		
1	5		
2	6		

```
import numpy as np
import pandas as pd
df = pd.DataFrame([[1, np.nan, 2],
[2, 3, 5],
[np.nan, 4, 6]])
print(df)
print(df.dropna(axis='rows', how='any'))
```

	0	1	2
0	1.0	NaN	2
1	2.0	3.0	5
2	NaN	4.0	6

	0	1	2
1	2.0	3.0	5

```
import numpy as np
import pandas as pd
data=pd.Series([1, np.nan, 2, None, 3],index=list('abcde'))
print(data)
print(data.fillna(method='ffill'))
```

	a	b	c	d	e	dtype:
	1.0	NaN	2.0	NaN	3.0	float64
	1.0	1.0	2.0	2.0	3.0	float64

```
import numpy as np
import pandas as pd
data=pd.Series([1, np.nan, 2, None, 3],index=list('abcde'))
print(data)
print(data.fillna(method='bfill'))
```

	a	b	c	d	e	dtype:
	1.0	NaN	2.0	NaN	3.0	float64
	1.0	2.0	2.0	3.0	3.0	float64

```
import numpy as np
import pandas as pd
df = pd.DataFrame([[1, np.nan, 2],
[2, 3, 5],
[np.nan, 4, 6]])
print(df)
print(df.fillna(method='ffill', axis=1))
```

	0	1	2
0	1.0	NaN	2
1	2.0	3.0	5
2	NaN	4.0	6
	0	1	2
0	1.0	1.0	2.0
1	2.0	3.0	5.0
2	NaN	4.0	6.0

```
import numpy as np
import pandas as pd
df = pd.DataFrame([[1, np.nan, 2],
[2, 3, 5],
[np.nan, 4, 6]])
print(df)
print(df.fillna(method='ffill', axis=0))
```

	0	1	2
0	1.0	NaN	2
1	2.0	3.0	5
2	NaN	4.0	6
	0	1	2
0	1.0	NaN	2
1	2.0	3.0	5
2	2.0	4.0	6

```
import numpy as np
import pandas as pd
df = pd.DataFrame([[1, np.nan, 2],
[2, 3, 5],
[np.nan, 4, 6]])
print(df)
print(df.fillna(method='bfill', axis=0))
print(df.fillna(method='bfill', axis=1))
```

	0	1	2
0	1.0	NaN	2
1	2.0	3.0	5
2	NaN	4.0	6
	0	1	2
0	1.0	3.0	2
1	2.0	3.0	5
2	NaN	4.0	6
	0	1	2
0	1.0	2.0	2.0
1	2.0	3.0	5.0
2	4.0	4.0	6.0

**Write a command to fill all the missing values with 999**

	0	1	2	3
0	1.0	NaN	2.0	NaN
1	2.0	3.0	5.0	NaN
2	NaN	4.0	6.0	NaN
3	NaN	8.0	NaN	NaN
4	NaN	NaN	NaN	NaN

**print(df.fillna(999))**

	0	1	2	3
0	1.0	999.0	2.0	999.0
1	2.0	3.0	5.0	999.0
2	999.0	4.0	6.0	999.0
3	999.0	8.0	999.0	999.0
4	999.0	999.0	999.0	999.0

**Write a command to fill all the missing values using forward filling across columns**

	0	1	2	3
0	1.0	NaN	2.0	NaN
1	2.0	3.0	5.0	NaN
2	NaN	4.0	6.0	NaN
3	NaN	8.0	NaN	NaN
4	NaN	NaN	NaN	NaN

```
print(df.fillna(method='ffill',axis=1))
```

	0	1	2	3
0	1.0	1.0	2.0	2.0
1	2.0	3.0	5.0	5.0
2	NaN	4.0	6.0	6.0
3	NaN	8.0	8.0	8.0
4	NaN	NaN	NaN	NaN

**Write a command to drop the columns containing all missing values**

	0	1	2	3
0	1.0	NaN	2.0	NaN
1	2.0	3.0	5.0	NaN
2	NaN	4.0	6.0	NaN
3	NaN	8.0	NaN	NaN
4	NaN	NaN	NaN	NaN

```
print(df.dropna(axis=1,how='all'))
```

	0	1	2
0	1.0	NaN	2.0
1	2.0	3.0	5.0
2	NaN	4.0	6.0
3	NaN	8.0	NaN
4	NaN	NaN	NaN

**Write a command to print all the rows containing at least 2 non-missing values**

	0	1	2	3
0	1.0	NaN	2.0	NaN
1	2.0	3.0	5.0	NaN
2	NaN	4.0	6.0	NaN
3	NaN	8.0	NaN	NaN
4	NaN	NaN	NaN	NaN

**print(df.dropna(axis=0,thresh=2))**

	0	1	2	3
0	1.0	NaN	2.0	NaN
1	2.0	3.0	5.0	NaN
2	NaN	4.0	6.0	NaN

**Write a command to drop all the rows containing missing values.**

	0	1	2	3
0	1.0	NaN	2.0	NaN
1	2.0	3.0	5.0	NaN
2	NaN	4.0	6.0	NaN
3	NaN	8.0	NaN	NaN
4	NaN	NaN	NaN	NaN

`print(df.dropna())`

Empty DataFrame

Columns: [0, 1, 2, 3]

Index: []

```
import numpy as np
import pandas as pd
a=[[1,np.nan,3],[np.nan,5,6],[7,8,np.nan]]
df=pd.DataFrame(a)
print(df)
print(df.dropna())
print(df.dropna(axis=0,how='any'))
```

	0	1	2
0	1.0	NaN	3.0
1	NaN	5.0	6.0
2	7.0	8.0	NaN

Empty DataFrame

Columns: [0, 1, 2]

Index: []

Empty DataFrame

Columns: [0, 1, 2]

Index: []

```
import numpy as np
import pandas as pd
data1=np.array([7,3,80,45,2,95,21,6,66,8,0,23,3,8,99,11]).reshape(4,4)
ind1=[1,2,3,4]
col1=['A','B','C','D']
df1=pd.DataFrame(data1,index=ind1,columns=col1)
data2=np.array([2,95,21,6,66,8,0,23,7,3,80,45,3,8,99,11]).reshape(4,4)
cols2=['A','B','D','E']
df2=pd.DataFrame(data2,columns=cols2)
print(df1)
print(df2)
```

	A	B	C	D		A	B	D	E	
1	7	3	80	45		0	2	95	21	6
2	2	95	21	6		1	66	8	0	23
3	66	8	0	23		2	7	3	80	45
4	3	8	99	11		3	3	8	99	11

**print(df1+df2)**

	A	B	C	D		A	B	D	E		A	B	C	D	E
1	7	3	80	45	0	2	95	21	6	0	NaN	NaN	NaN	NaN	NaN
2	2	95	21	6	1	66	8	0	23	1	73.0	11.0	NaN	45.0	NaN
3	66	8	0	23	2	7	3	80	45	2	9.0	98.0	NaN	86.0	NaN
4	3	8	99	11	3	3	8	99	11	3	69.0	16.0	NaN	122.0	NaN
					4					4	NaN	NaN	NaN	NaN	NaN

```
print(df1.add(df2,fill_value=0))
```

	A	B	C	D		A	B	D	E		A	B	C	D	E
1	7	3	80	45	0	2	95	21	6	0	2.0	95.0	NaN	21.0	6.0
2	2	95	21	6	1	66	8	0	23	1	73.0	11.0	80.0	45.0	23.0
3	66	8	0	23	2	7	3	80	45	2	9.0	98.0	21.0	86.0	45.0
4	3	8	99	11	3	3	8	99	11	3	69.0	16.0	0.0	122.0	11.0
					4	3.0	8.0	99.0	11.0	4	3.0	8.0	99.0	11.0	NaN

```
print(df1-df1.iloc[1])
```

	A	B	C	D
1	7	3	80	45
2	2	95	21	6
3	66	8	0	23
4	3	8	99	11

	A	B	C	D
1	5	-92	59	39
2	0	0	0	0
3	64	-87	-21	17
4	1	-87	78	5

```
print(df1-df1.loc[1])
```

	A	B	C	D		A	B	C	D
1	7	3	80	45		1	0	0	0
2	2	95	21	6		2	-5	92	-59
3	66	8	0	23		3	59	5	-80
4	3	8	99	11		4	-4	5	19

```
import numpy as np
import pandas as pd
ind=['a','b','c','d','e','f','g','h']
col=['A', 'B', 'C', 'D']
df = pd.DataFrame(np.arange(32).reshape(8, 4),index=ind,columns=col)
print(df)
```

	A	B	C	D
a	0	1	2	3
b	4	5	6	7
c	8	9	10	11
d	12	13	14	15
e	16	17	18	19
f	20	21	22	23
g	24	25	26	27
h	28	29	30	31

```
print(df.loc[:, 'A'])
```

	A	B	C	D
a	0	1	2	3
b	4	5	6	7
c	8	9	10	11
d	12	13	14	15
e	16	17	18	19
f	20	21	22	23
g	24	25	26	27
h	28	29	30	31

	A
a	0
b	4
c	8
d	12
e	16
f	20
g	24
h	28

Name: A, dtype: int64

```
print(df.loc[['a','b', 'd','e'], ['A','C']])
```

	A	B	C	D		A	C	
a	0	1	2	3		a	0	2
b	4	5	6	7		b	4	6
c	8	9	10	11		d	12	14
d	12	13	14	15		e	16	18
e	16	17	18	19				
f	20	21	22	23				
g	24	25	26	27				
h	28	29	30	31				

```
print(df.loc['a']>0)
```

	A	B	C	D
a	0	1	2	3
b	4	5	6	7
c	8	9	10	11
d	12	13	14	15
e	16	17	18	19
f	20	21	22	23
g	24	25	26	27
h	28	29	30	31

A	False
B	True
C	True
D	True
	Name: a, dtype: bool

```
print(df.iloc[1:5, 2:4])
```

	A	B	C	D		C	D
a	0	1	2	3	b	6	7
b	4	5	6	7	c	10	11
c	8	9	10	11	d	14	15
d	12	13	14	15	e	18	19
e	16	17	18	19			
f	20	21	22	23			
g	24	25	26	27			
h	28	29	30	31			

```
print(df.loc['b':'e', 'B':'D'])
```

	A	B	C	D		B	C	D
a	0	1	2	3	b	5	6	7
b	4	5	6	7	c	9	10	11
c	8	9	10	11	d	13	14	15
d	12	13	14	15	e	17	18	19
e	16	17	18	19				
f	20	21	22	23				
g	24	25	26	27				
h	28	29	30	31				

```
import numpy as np          0
import pandas as pd
data=np.arange(1,17).reshape(4,4)
ind=[4,3,2,1]
col=['a','b','c','d']
df=pd.DataFrame(data,ind,col)
print(np.sum(df.iloc[0]-df.loc[4]))
```

```
import numpy as np                                10
import pandas as pd
data=np.arange(1,17).reshape(4,4)
ind=[4,3,2,1]
col=['a','b','c','d']
df=pd.DataFrame(data,ind,col)
print(np.sum(df.iloc[-4]))
```

```
import numpy as np
import pandas as pd
data=np.arange(1,17).reshape(4,4)
ind=[4,3,2,1]
col=['a','b','c','d']
df=pd.DataFrame(data,ind,col)
print(np.sum(np.sum(df.iloc[-4:-2])))
```

```
import numpy as np
import pandas as pd
data=np.arange(1,17).reshape(4,4)
ind=[4,3,2,1]
col=['a','b','c','d']
df=pd.DataFrame(data,ind,col)
print(np.sum(np.sum(df.loc[4:3:1])))
```

```
import numpy as np
import pandas as pd
data=np.arange(1,17).reshape(4,4)
ind=[4,3,2,1]
col=['a','b','c','d']
df=pd.DataFrame(data,ind,col)
print(np.sum(np.sum(df.loc[4:3:2])))
```

10

```
import numpy as np
import pandas as pd
a=[[1,2,np.nan],[4,np.nan,6],[np.nan,8,9]]
i1=[1,2,3]
c1=['A','B','C']
df1=pd.DataFrame(a,index=i1,columns=c1)
print(df1)
b=[[np.nan,20,30],[40,np.nan,60],[70,80,np.nan]]
i2=[3,4,5]
c2=['C','D','E']
df2=pd.DataFrame(b,index=i2,columns=c2)
print(df2)
print(df1+df2)
```

	A	B	C
1	1.0	2.0	NaN
2	4.0	NaN	6.0
3	NaN	8.0	9.0
	C	D	E
3	NaN	20.0	30.0
4	40.0	NaN	60.0
5	70.0	80.0	NaN

	A	B	C	D	E
1	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN
3	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN
5	NaN	NaN	NaN	NaN	NaN

---

```
import numpy as np
import pandas as pd
a=[[1,2,np.nan],[4,np.nan,6],[np.nan,8,9]]
i1=[1,2,3]
c1=['A','B','C']
df1=pd.DataFrame(a,index=i1,columns=c1)
print(df1)
b=[[np.nan,20,30],[40,np.nan,60],[70,80,np.nan]]
i2=[3,4,5]
c2=['C','D','E']
df2=pd.DataFrame(b,index=i2,columns=c2)
print(df2)
print(df1.add(df2,fill_value=0))
```

	A	B	C		A	B	C	D	E
1	1.0	2.0	NaN		1	1.0	2.0	NaN	NaN
2	4.0	NaN	6.0		2	4.0	NaN	6.0	NaN
3	NaN	8.0	9.0		3	NaN	8.0	9.0	20.0
	C	D	E		3	NaN	8.0	9.0	30.0
3	NaN	20.0	30.0		4	NaN	NaN	40.0	NaN
4	40.0	NaN	60.0		5	NaN	NaN	70.0	80.0
5	70.0	80.0	NaN						NaN

```
import numpy as np  
import pandas as pd  
data=[10,20,30,40,50]  
df=pd.Series(data,index=['E','D','C','B','A'])  
print(np.sum(df[1:4]))
```

90

```
import numpy as np          60
import pandas as pd         150
data=[10,20,30,40,50]
df=pd.Series(data,index=['E','D','C','B','A'])
print(np.sum(df['E':'C']))  90
print(np.sum(df[::-1]))    100
print(np.sum(df['B':'D'][::-1]))
```

```
import numpy as np
import pandas as pd
data1=[10,20,30,40,50]
s1=pd.Series(data1,index=['a','b','c','d','e'])
data2=[1,2,3,4,5]
s2=pd.Series(data2,index=['c','d','e','f','g'])
print(s1+s2)
```

a	NaN
b	NaN
c	31.0
d	42.0
e	53.0
f	NaN
g	NaN
	dtype: float64

```
import numpy as np
import pandas as pd
data1=[10,20,30,40,50]
s1=pd.Series(data1,index=['a','b','c','d','e'])
data2=[1,2,3,4,5]
s2=pd.Series(data2,index=['c','d','e','f','g'])
print(np.nansum(s1+s2))
```

126.0

```
import numpy as np
import pandas as pd
data1=[10,20,30,40,50]
s1=pd.Series(data1,index=['a','b','c','d','e'])
data2=[1,2,3,4,5]
s2=pd.Series(data2,index=['c','d','e','f','g'])
print(s1.add(s2,fill_value=5))
```

```
import numpy as np
import pandas as pd
data=[[1,2,np.nan],[np.nan,5,6],[7,np.nan,9]]
df=pd.DataFrame(data)
print(np.sum(np.sum(df.isnull())))
```

3

```
import numpy as np
import pandas as pd
s = pd.Series(['Tom', 'William Rick', 'John', 'Albert', 'Steve Smith'])
#Write a command to convert all the strings to lower case
print(s.str.lower())
print(s.str.upper())
```

```
0          tom
```

```
1    william rick
```

```
2          john
```

```
3          albert
```

```
4      steve smith
```

```
dtype: object
```

```
0          TOM
```

```
1    WILLIAM RICK
```

```
2          JOHN
```

```
3          ALBERT
```

```
4      STEVE SMITH
```

```
dtype: object
```

---

```
import numpy as np
import pandas as pd
s = pd.Series(['Tom', 'William Rick', 'John', 'Albert', 'Steve Smith'])
print(s.str.endswith('T'))
print(s[s.str.endswith('T')])
```

```
0    False
1    False
2    False
3    False
4    False
dtype: bool
Series([], dtype: object)
```

```
import numpy as np
import pandas as pd
s = pd.Series(['Tom', 'William Rick', 'John', 'Albert', 'Steve Smith'])
print(s.str[5])

```

---

	0	NaN
	1	a
	2	NaN
	3	t
	4	
		dtype: object

```
import numpy as np
import pandas as pd
s = pd.Series(['Tom', 'William Rick', 'John', 'Albert', 'Steve Smith'])
print(s.str[1:4])
0      om
1      ill
2      ohn
3      lbe
4      tev
dtype: object
```

```
import numpy as np
import pandas as pd
s = pd.Series(['abcnde fghnij'])
print(s.str.split(pat='n'))
print(s.str.split(pat='n',n=-1))
print(s.str.split(pat='n',n=0))
print(s.str.split(pat='n',n=1))
```

```
0      [abc, de fgh, ij]
dtype: object
0      [abc, de fgh, ij]
dtype: object
0      [abc, de fgh, ij]
dtype: object
0      [abc, de fghnij]
dtype: object
```

```
import numpy as np
import pandas as pd
s = pd.Series(['Tom', 'William Rick', 'John', 'Albert', 'Steve Smith'])
print(s.str.split().str.get(0))
```

0	Tom
1	William
2	John
3	Albert
4	Steve
	dtype: object

```
import numpy as np
import pandas as pd
s = pd.Series(['Tom', 'William Rick', 'John', 'Albert', 'Steve Smith'])
print(s.str.split().str.get(1))

```

---

		0	NaN
		1	Rick
		2	NaN
		3	NaN
		4	Smith
			dtype: object

```
import numpy as np
import pandas as pd
s = pd.Series(['Tom', 'William Rick', 'John', 'Albert', 'Steve Smith'])
print(s.str.split(pat=' '))

0      [Tom]
1      [William, Rick]
2      [John]
3      [Albert]
4      [Steve, Smith]
dtype: object
```

```
import numpy as np
import pandas as pd
s=pd.Series(['Kiran','Arun','Anuj'])
print(s)
s1=s.str.split()
print(s1)
print(s1.str.get(0))
print(s1.str.get(-1))
```

```
0    Kiran
1    Arun
2    Anuj
dtype: object
0    [Kiran]
1    [Arun]
2    [Anuj]
dtype: object
```

```
0    Kiran
1    Arun
2    Anuj
dtype: object
0    Kiran
1    Arun
2    Anuj
dtype: object
```

```
import numpy as np
import pandas as pd
s=pd.Series(['Kiran','Arun','Anuj'])
print(s)
0      NaN
1      NaN
2      NaN
dtype: float64
s1=s.str.split()
print(s1)
print(s1.str.get(1))
```

```
import numpy as np
import pandas as pd
s=pd.Series(['Kiran','Arun','Anuj'])
print(s)
s1=s.str.split(pat='A')
print(s1)
print(s1.str.get(1))
print(s1.str.get(-1))
print(s1.str.get(0))
```

```
          0      NaN
          1      run
          2     nuj
0    Kiran      dtype: object
1   Arun      0    Kiran
2   Anuj      1      run
dtype: object      2      nuj
dtype: object
0   [Kiran]      0    Kiran
1   [, run]      1
2   [, nuj]      2
dtype: object      dtype: object
```

```
import numpy as np          0      @khil
import pandas as pd         1      @bhijit
data=['Akhil','Abhijit','Ananya','Arpit']    2      @nanya
s=pd.Series(data)           3      @rpit
print(s.str.replace('A','@'))  dtype: object
```

```
import numpy as np
import pandas as pd
s=pd.Series(['ram','rahim','ryan'])
print(s.str.capitalize())
0      Ram
1    Rahim
2     Ryan
dtype: object
```

```
import numpy as np
import pandas as pd
s=pd.Series(['ram','rahim','ryan'])
print(s.str.islower())
s1=pd.Series(['ram1995','rahim2003','ryan2002'])
print(s1.str.isalpha())
print(s1.str.isalnum())
```

0	True	
1	True	
2	True	
	dtype: bool	
0	False	
1	False	
2	False	
	dtype: bool	
0	True	
1	True	
2	True	
	dtype: bool	

```
import numpy as np
import pandas as pd
s=pd.Series(['ram','rahim','Ananya'])
print(s[0].index('a'))
print(s[1].index('a'))
print(s[2].index('a'))
print(s[2].index('a',3,6))
```

1

1

2

5

```
import numpy as np  
import pandas as pd  
s=pd.Series(['ram','rahim','Ananya'])  
print(s[0].count('a'))  
print(s[1].count('a'))  
print(s[2].count('a'))  
print(s[2].count('a',3,5))
```

1

1

2

0

```
import numpy as np
import pandas as pd
s1=pd.Series([10,20,30])
s2=pd.Series([5,30,25])
print(s1<s2)
print(np.sum(s1<s2))
```

	0	<b>False</b>
	1	<b>True</b>
	2	<b>False</b>
		<b>dtype: bool</b>
	1	

```
import numpy as np
import pandas as pd
df1=pd.DataFrame([[1,2,3],[4,5,6],[7,8,9]])
df2=pd.DataFrame([[2,3,1],[6,4,5],[8,7,9]])
print(df1<df2)
print(np.sum(df1<df2))
print(np.sum(np.sum(df1<df2)))
```

	0	1	2
0	True	True	False
1	True	False	False
2	True	False	False
0	3		
1	1		
2	0		
		dtype: int64	
4			

```
import numpy as np          4
import pandas as pd         3
data=[[ 'A',24,78],[ 'B',25,69],[ 'C',22,89],[ 'D',23,82]]
df=pd.DataFrame(data,index=[1,2,3,4],columns=['Name','Age','Score'])
#Count the number of rows
print(len(df.index))
#Count the number of columns
print(len(df.columns))
```

```
import numpy as np
import pandas as pd
data=[[ 'Ram',26,87],['Sai',25,67],['Rahim',25,89],
['Amit',24,55],[ 'Alex',31,47],[ 'Ram',33,90]]
col=[ 'Name', 'Age', 'Score']
ind=[ 'A', 'B', 'C', 'D', 'E', 'F']
df=pd.DataFrame(data,index=ind,columns=col)
print(df)

#Print all records where the player name is 'Ram'
#Print the names of the players whose ages are greater than 30
#Print the names of the players whose scores are greater than
#the average score of all players
```

```
print(df[df['Name']=='Ram'])  
print(df[df['Age']>30]['Name'])  
print(df[df['Score']>df['Score'].mean()]['Name'])
```

	Name	Age	Score
A	Ram	26	87
F	Ram	33	90
E	Alex		
F	Ram		
	Name:	Name,	dtype: object
A	Ram		
C	Rahim		
F	Ram		
	Name:	Name,	dtype: object

```
import numpy as np
import pandas as pd
data=[['Ram',26,87],['Sai',25,67],['Rahim',25,89],
['Amit',24,np.nan],['Alex',31,47],['Ram',33,90]]
col=['Name','Age','Score']
ind=['A','B','C','D','E','F']
df=pd.DataFrame(data,index=ind,columns=col)
print(df['Score'].fillna(np.nanmean(df['Score'])))
```

---

```
import numpy as np
import pandas as pd
data=[['A',24,78],['B',25,69],['C',22,np.nan],['D',23,82]]
df=pd.DataFrame(data,index=[1,2,3,4],columns=['Name','Age','Score'])
print(df.loc[df['Age']>23,'Name'])
```

```
1    A
2    B
Name: Name, dtype: object
```

```
import numpy as np
import pandas as pd
data=[[ 'A',24,78],[ 'B',25,69],[ 'C',22,np.nan],[ 'D',23,82]]
df=pd.DataFrame(data)
print(df.loc[df[1]>23,0])
```

	0	A
1	B	

Name: 0, dtype: object

```
import numpy as np
import pandas as pd
data=[['A',30],['B',20],['C',-7],['D',35],['E',-2]]
df=pd.DataFrame(data,columns=['country','temperature'])
#Replace -ve temperature values with 0.
df.loc[df['temperature']<0,'temperature']=0
print(df)
```

	country	temperature
0	A	30
1	B	20
2	C	0
3	D	35
4	E	0

```
import numpy as np
import pandas as pd
data=[['A',30],['B',20],['C',10],['D',5],['E',4]]
df=pd.DataFrame(data,columns=['name','score'])
#Adding column with condition
df['result']=np.nan
df.loc[df['score']>=10,'result']='pass'
df.loc[df['score']<10,'result']='fail'
print(df)
```

	name	score	result
0	A	30	pass
1	B	20	pass
2	C	10	pass
3	D	5	fail
4	E	4	fail

```
import numpy as np
import pandas as pd
data=[[ 'A',24,78],[ 'B',25,69],[ 'C',22,89],[ 'D',23,82]]
df=pd.DataFrame(data,index=[1,2,3,4],columns=['Name','Age','Score'])
#Display first two rows
print(df.loc[1:2])
print(df.iloc[0:2])
```

	Name	Age	Score
1	A	24	78
2	B	25	69

	Name	Age	Score
1	A	24	78
2	B	25	69

```
import numpy as np  
import pandas as pd data=[['A',24,78],['B',25,69],['C',22,89],['D',23,82]]  
df=pd.DataFrame(data,index=[1,2,3,4],columns=['Name','Age','Score'])  
print(np.sum(df['Score']))  
print(np.prod(df['Score']))  
print(np.min(df['Score']))  
print(np.argmin(df['Score']))  
print(np.max(df['Score']))  
print(np.argmax(df['Score']))  
print(np.mean(df['Score']))  
print(np.median(df['Score']))  
print(np.var(df['Score']))  
print(np.std(df['Score']))  
print(np.percentile(df['Score'],50))
```

318	
39277836	
69	
1	
89	
2	
79 . 5	
80 . 0	
52 . 25	
7 . 22841614740048	
80 . 0	

```
import numpy as np
import pandas as pd
data=[['A',24,78],['B',25,69],['C',22,89],['D',23,82]]
df=pd.DataFrame(data,index=[1,2,3,4],columns=['Name','Age','Score'])
print(df)
df.loc[df['Score']==np.nanmin(df['Score']),'Score']=0
print(df)
```

	Name	Age	Score
1	A	24	78
2	B	25	69
3	C	22	89
4	D	23	82

	Name	Age	Score
1	A	24	78
2	B	25	0
3	C	22	89
4	D	23	82

```
import numpy as np
import pandas as pd
data=[['A',24,78],['B',25,69],['C',22,np.nan],['D',23,82]]
df=pd.DataFrame(data,index=[1,2,3,4],columns=['Name','Age','Score'])
print(df)
df['Score']=df['Score'].fillna(np.nanmean(df['Score']))
print(df)
```

	Name	Age	Score
1	A	24	78.0
2	B	25	69.0
3	C	22	NaN
4	D	23	82.0

	Name	Age	Score
1	A	24	78.000000
2	B	25	69.000000
3	C	22	76.333333
4	D	23	82.000000

```
import numpy as np
import pandas as pd
data=[[ 'A',24,78],[ 'B',25,69],[ 'C',22,np.nan],[ 'D',23,82]]
df=pd.DataFrame(data,index=[1,2,3,4],columns=['Name','Age','Score'])
print(df[1:3])
print(df.iloc[1:3])
print(df.loc[2:3])
```

	Name	Age	Score
2	B	25	69.0
3	C	22	NaN

	Name	Age	Score
2	B	25	69.0
3	C	22	NaN

	Name	Age	Score
2	B	25	69.0
3	C	22	NaN

```
import numpy as np
import pandas as pd
data=[['A',24,78],['B',25,69],['C',22,np.nan],['D',23,82]]
df=pd.DataFrame(data,index=[1,2,3,4],columns=['Name','Age','Score'])
print(df[1:3][['Age','Score']])
print(df.iloc[1:3][['Age','Score']])
print(df.loc[2:3][['Age','Score']])
.
```

	Age	Score
2	25	69.0
3	22	NaN

	Age	Score
2	25	69.0
3	22	NaN

	Age	Score
2	25	69.0
3	22	NaN

```
import pandas as pd  
  
data = {'Name':['A', 'B', 'C', 'D'],  
        'Age':[27, 24, 22, 32],  
        'Address':['W', 'X', 'Y', 'Z'],  
        'Qualification':['Msc', 'MA', 'MCA', 'Phd']}  
  
df = pd.DataFrame(data)  
  
print(df)
```

	Name	Age	Address	Qualification
0	A	27	W	Msc
1	B	24	X	MA
2	C	22	Y	MCA
3	D	32	Z	Phd

```
import pandas as pd  
  
data = {'Name':['A', 'B', 'C', 'D'],  
        'Age':[27, 24, 22, 32],  
        'Address':['W', 'X', 'Y', 'Z'],  
        'Qualification':['Msc', 'MA', 'MCA', 'Phd']}  
  
df = pd.DataFrame(data)  
  
print(df.iloc[1])
```

Name	B
Age	24
Address	X
Qualification	MA

```
import pandas as pd
data = {'Name':['A', 'B', 'C', 'D'],
        'Age':[27, 24, 22, 32],
        'Address':['W', 'X', 'Y', 'Z'],
        'Qualification':['Msc', 'MA', 'MCA', 'Phd']}
df = pd.DataFrame(data)
print(df.iloc[1:])
```

	Name	Age	Address	Qualification
1	B	24	X	MA
2	C	22	Y	MCA
3	D	32	Z	Phd

```
import pandas as pd
data = {'Name':['A', 'B', 'C', 'D'],
        'Age':[27, 24, 22, 32],
        'Address':['W', 'X', 'Y', 'Z'],
        'Qualification':['Msc', 'MA', 'MCA', 'Phd']}
df = pd.DataFrame(data)
print(df.iloc[-2:])
```

	Name	Age	Address	Qualification
2	C	22	Y	MCA
3	D	32	Z	Phd

```
import pandas as pd  
  
data = {'Name':['A', 'B', 'C', 'D'],  
        'Age':[27, 24, 22, 32],  
        'Address':['W', 'X', 'Y', 'Z'],  
        'Qualification':['Msc', 'MA', 'MCA', 'Phd']}  
  
df = pd.DataFrame(data)  
  
print(df.iloc[1:,1:])
```

	Age	Address	Qualification
1	24	X	MA
2	22	Y	MCA
3	32	Z	Phd

```
import pandas as pd  
  
data = {'Name':['A', 'B', 'C', 'D'],  
        'Age':[27, 24, 22, 32],  
        'Address':['W', 'X', 'Y', 'Z'],  
        'Qualification':['Msc', 'MA', 'MCA', 'Phd']}  
  
df = pd.DataFrame(data)  
  
print(df.iloc[-1::-1,1:])
```

	Age	Address	Qualification
3	32	Z	Phd
2	22	Y	MCA
1	24	X	MA
0	27	W	Msc

```
import pandas as pd  
  
data = {'Name': ['A', 'B', 'C', 'D'],  
        'Age': [27, 24, 22, 32],  
        'Address': ['W', 'X', 'Y', 'Z'],  
        'Qualification': ['Msc', 'MA', 'MCA', 'Phd']}  
  
df = pd.DataFrame(data, index=['ONE', 'TWO', 'THREE', 'FOUR'])  
  
print(df)
```

	Name	Age	Address	Qualification
ONE	A	27	W	Msc
TWO	B	24	X	MA
THREE	C	22	Y	MCA
FOUR	D	32	Z	Phd

```
import pandas as pd
data = {'Name':['A', 'B', 'C', 'D'],
        'Age':[27, 24, 22, 32],
        'Address':['W', 'X', 'Y', 'Z'],
        'Qualification':['Msc', 'MA', 'MCA', 'Phd']}
df = pd.DataFrame(data,index=['ONE', 'TWO', 'THREE', 'FOUR'])
df["Marks"]=[230,415,357,176]
df["Percentage"]=df["Marks"]/500*100
print(df)
```

	Name	Age	Address	Qualification	Marks	Percentage
ONE	A	27	W	Msc	230	46.0
TWO	B	24	X	MA	415	83.0
THREE	C	22	Y	MCA	357	71.4
FOUR	D	32	Z	Phd	176	35.2

```
import pandas as pd
data = { 'Name': ['A', 'B', 'C', 'D'],
          ...
          'Age':[27, 24, 22, 32],
          ...
          'Address':['W', 'X', 'Y', 'Z'],
          ...
          'Qualification':['Msc', 'MA', 'MCA', 'Phd']}
df = pd.DataFrame(data)
df1={'Name': 'E', 'Age': 28, 'Address': 'V', 'Qualification': 'BTech'}
print(df.append(df1, ignore_index = True))
```

	Name	Age	Address	Qualification
0	A	27	W	Msc
1	B	24	X	MA
2	C	22	Y	MCA
3	D	32	Z	Phd
4	E	28	V	BTech

```
import pandas as pd
data = {'Name':['A', 'B', 'C', 'D'],
        'Age':[27, 24, 22, 32],
        'Address':['W', 'X', 'Y', 'Z'],
        'Qualification':['Msc', 'MA', 'MCA', 'Phd']}
df = pd.DataFrame(data,index=['ONE', 'TWO', 'THREE', 'FOUR'])
df1=df.sort_values(by=["Age"])
print(df1)
```

	Name	Age	Address	Qualification
ONE	C	22	Y	MCA
TWO	B	24	X	MA
THREE	A	27	W	Msc
FOUR	D	32	Z	Phd

```
import pandas as pd
import numpy as np
data = {'Name':['A', 'B', 'C', 'D'],
        'Age':[27, np.nan, 22, 32],
        'Address':['W', 'X', 'Y', 'Z'],
        'Qualification':['Msc', 'MA', 'MCA', np.nan]}
df = pd.DataFrame(data)
print(df)
print(df.isnull())
print(df.isnull().sum())
```

	Name	Age	Address	Qualification
0	A	27.0	w	Msc
1	B	NaN	x	MA
2	C	22.0	y	MCA
3	D	32.0	z	NaN

	Name	Age	Address	Qualification
0	False	False	False	False
1	False	True	False	False
2	False	False	False	False
3	False	False	False	True

	0
Name	0
Age	1
Address	0
Qualification	1

dtype: int64

```
import pandas as pd
import numpy as np
data = {'Name':['A', 'B', 'C', 'D'],
        'Age':[27, np.nan, 22, 32],
        'Address':['W', 'X', 'Y', 'Z'],
        'Qualification':['Msc', 'MA', 'MCA', np.nan]}
df = pd.DataFrame(data)
newdf = df.dropna()
print(newdf)
```

	Name	Age	Address	Qualification
0	A	27.0	W	Msc
2	C	22.0	Y	MCA

```
import numpy as np [1 1 2]
import pandas as pd [False False  True]
a=np.array([[1,2,np.nan],[4,np.nan,6],[np.nan,8,np.nan]])) [[nan]
b=np.isnan(a) [ 6.]
c=np.sum(b, axis=0) [nan]
print(c)
print(c==2)
print(a[:,c==2])
```

```
import numpy as np
import pandas as pd
a=np.array([[1,2,np.nan],[4,np.nan,6],[np.nan,8,np.nan]])
b=a.shape
print(b[0])
print(b[1])
```

```
import numpy as np 5
import pandas as pd
data=[[1,'A',60,19],[2,'B',70,20],[3,'C',75,19],[4,'D',85,21],[5,'E',95,20]] 4
df=pd.DataFrame(data,columns=['Roll','Name','Marks','Age'])
print(len(df.index)) 5
print(len(df.columns)) 4
print(df.shape[0])
print(df.shape[1])
```

```
import numpy as np
import pandas as pd
data=[[1,'A',60,19],[2,'B',70,20],[3,'C',75,19],[4,'D',85,21],[5,'E',95,20]]
df=pd.DataFrame(data,columns=['Roll','Name','Marks','Age'])
new_row=[6,'F',90,20]
df.loc[len(df.index),:]=new_row
print(df)
lst=['M','M','F','F','M','M']
df.loc[:,len(df.columns)]=lst
df.columns=['Roll','Name','Marks','Age','Gender']
print(df)
```

	Roll	Name	Marks	Age
0	1.0	A	60.0	19.0
1	2.0	B	70.0	20.0
2	3.0	C	75.0	19.0
3	4.0	D	85.0	21.0
4	5.0	E	95.0	20.0
5	6.0	F	90.0	20.0

	Roll	Name	Marks	Age	Gender
0	1.0	A	60.0	19.0	M
1	2.0	B	70.0	20.0	M
2	3.0	C	75.0	19.0	F
3	4.0	D	85.0	21.0	F
4	5.0	E	95.0	20.0	M
5	6.0	F	90.0	20.0	M

```
import numpy as np
import pandas as pd
data = {
    'A': ['A1', 'A2', 'A3', 'A4', 'A5'],
    'B': ['B1', 'B2', 'B3', 'B4', 'B5'],
    'C': ['C1', 'C2', 'C3', 'C4', 'C5'],
    'D': ['D1', 'D2', 'D3', 'D4', 'D5'],
    'E': ['E1', 'E2', 'E3', 'E4', 'E5']}
df = pd.DataFrame(data)
df=df.drop(['A'], axis=1)
print(df)
```

	B	C	D	E
0	B1	C1	D1	E1
1	B2	C2	D2	E2
2	B3	C3	D3	E3
3	B4	C4	D4	E4
4	B5	C5	D5	E5

```
import numpy as np
import pandas as pd
details = {
    'A': ['A1', 'A2', 'A3', 'A4', 'A5'],
    'B': ['B1', 'B2', 'B3', 'B4', 'B5'],
    'C': ['C1', 'C2', 'C3', 'C4', 'C5'],
    'D': ['D1', 'D2', 'D3', 'D4', 'D5'],
    'E': ['E1', 'E2', 'E3', 'E4', 'E5']}
df = pd.DataFrame(details, index = ['a', 'b', 'c', 'd', 'e'])
df = df.drop(['b', 'c'])
print(df)
```

	A	B	C	D	E
a	A1	B1	C1	D1	E1
d	A4	B4	C4	D4	E4
e	A5	B5	C5	D5	E5

```
import numpy as np
import pandas as pd
data = {
    'A': ['A1', 'A2', 'A3', 'A4', 'A5'],
    'B': ['B1', 'B2', 'B3', 'B4', 'B5'],
    'C': ['C1', 'C2', 'C3', 'C4', 'C5'],
    'D': ['D1', 'D2', 'D3', 'D4', 'D5'],
    'E': ['E1', 'E2', 'E3', 'E4', 'E5']}
df = pd.DataFrame(data)
print(df.drop(df.columns[[0,1]], axis=1))
print(df.drop(df.index[[0,1]], axis=0))
```

0	C1	D1	E1		
1	C2	D2	E2		
2	C3	D3	E3		
3	C4	D4	E4		
4	C5	D5	E5		
	A	B	C	D	E
2	A3	B3	C3	D3	E3
3	A4	B4	C4	D4	E4
4	A5	B5	C5	D5	E5

---

```
import numpy as np
a=np.arange(10,19).reshape(3,3)
print(np.argmin(a))
print(np.argmax(a))
```

0

8

```
import numpy as np
a=np.ones((3,3))
a[:-1]=np.nan
print(a)
print(np.nansum(a))
```

[[nan nan nan]  
 [nan nan nan]  
 [ 1. 1. 1.]])  
3.0

```
import pandas as pd
import numpy as np
data=np.arange(1,11)
s=pd.Series(data)
print(s[:3])
print(s[-8::-1])
```

	0	1
0	1	2
1	2	3
2	3	

dtype: int64

	0	1
0	1	2
1	2	3
2	3	

dtype: int64

---

---

```
import pandas as pd
import numpy as np
data=np.arange(1,11)
s=pd.Series(data,index=np.arange(11,21))
print(np.sum(s.loc[11:13]))
print(np.sum(s[:-1]))
```

6

45

```
import pandas as pd
import numpy as np
data=np.arange(1,11)
s=pd.Series(data,index=np.arange(11,21))
print(np.sum(s[s<=5]))
```

```
import pandas as pd
import numpy as np
data=np.array([10,15,np.nan,18,16])
s=pd.Series(data,index=['E','D','C','B','A'])
print(s.sort_values())
print(s.sort_values(ascending=False))
```

E	10.0	
D	15.0	
A	16.0	
B	18.0	
C	NaN	
		dtype: float64
B	18.0	
A	16.0	
D	15.0	
E	10.0	
C	NaN	
		dtype: float64

---

```
import pandas as pd
import numpy as np
data=np.array([10,15,np.nan,18,16])
s=pd.Series(data,index=['E','D','C','B','A'])
print(s.sort_index())
print(s.sort_index(ascending=False))
```

A	16.0	
B	18.0	
C	NaN	
D	15.0	
E	10.0	
		dtype: float64
E	10.0	
D	15.0	
C	NaN	
B	18.0	
A	16.0	
		dtype: float64

---

```
import pandas as pd
import numpy as np
s1=pd.Series({'A':10,'B':20,'C':30})
s2=pd.Series([10,20,30],index=['C','B','A'])
print(s1+s2)
```

	A	40
	B	40
	C	40
		dtype: int64

```
import pandas as pd
import numpy as np
s1=pd.Series({'C':10,'B':20,'A':30})
s2=pd.Series([20,20,20],index=[x for x in 'ABC'])
print(s1+s2)
```

A	50
B	40
C	30
	dtype: int64

```
import pandas as pd
import numpy as np
s1=pd.Series([10,20,30],index=range(3))
s2=pd.Series([20,20,20],index=[x for x in 'ABC'])
print(s1+s2)
```

0	NaN
1	NaN
2	NaN
A	NaN
B	NaN
C	NaN

dtype: float64

---

```
import pandas as pd          0      110
import numpy as np           1      220
s1=pd.Series([10,20,30],index=range(3))    2      330
s2=pd.Series(s1*10)          dtype: int64
print(s1+s2)
```

```
import pandas as pd
import numpy as np
data = [[10, 18, 11], [13, 15, 8], [9, 20, 3]]
df = pd.DataFrame(data)
print(df.max())
```

	0	13
0	13	20
1	20	11
2	11	
		dtype: int64

```
import pandas as pd
import numpy as np
data = [[10, 18, 11], [13, 15, 8], [9, 20, 3]]
df = pd.DataFrame(data)
print(df.max(axis=0).max() == df.max(axis=1).max())
```

True

---

```
import numpy as np
import pandas as pd
a=np.array([[1,2,3],[4,5,6],[7,8,9]])
print(np.max(a))
print(a.max())
```

9

9

---

```
import numpy as np
import pandas as pd
a=np.diag(np.arange(10))
print(np.sum(a))
```

45

```
import numpy as np
import pandas as pd
a=np.diag(np.arange(10))
print(np.sum(np.sum(a, axis=0)==np.sum(a, axis=1)))
```

10

---

```
from numpy import random
x = random.randint(100)
print(x)
```

50

```
from numpy import random      0.25089968764803505
x = random.rand()
print(x)
```

```
from numpy import random [61 77 73 45 84]
x=random.randint(100, size=(5)) [[79 42 68 98 56]
print(x) [64 88 55 90 86]
y = random.randint(100, size=(3, 5)) [94 45 17 88 49]]
print(y)
```

```
import pandas as pd
data=[[1,2,3],[4,5,6],[7,8,9]]
df=pd.DataFrame(data,index=[10,20,30],columns=['A','B','C'])
s1=df.iloc[:,0]
s2=df.loc[:, 'A']
print(s1)
print(type(s1))
print(s2)
print(type(s2))
```

```
10    1
20    4
30    7
Name: A, dtype: int64
<class 'pandas.core.series.Series'>
10    1
20    4
30    7
Name: A, dtype: int64
<class 'pandas.core.series.Series'>
```

```
import pandas as pd
df = pd.DataFrame(
    [[1, 2, 3],
     [4, 5, 6],
     [7, 8, 9],
     [10, 11, 12]],
    columns=['a', 'b', 'c'])
arr = df.to_numpy()
print(type(arr))
```

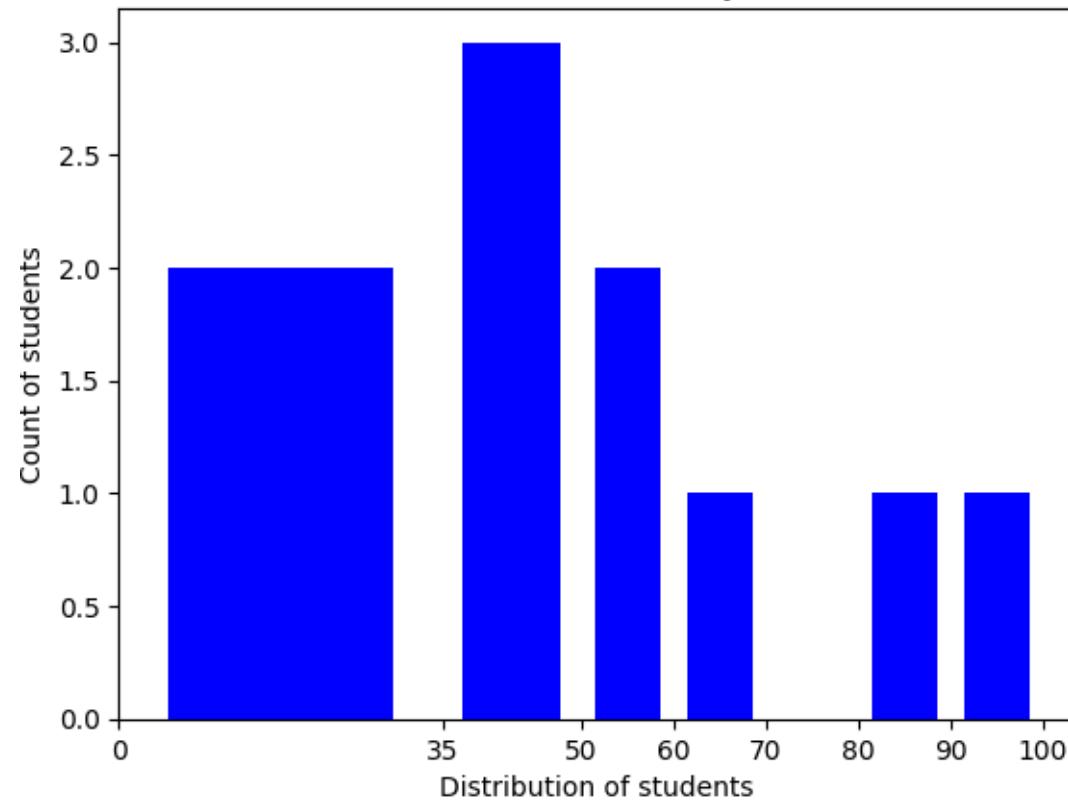
```
<class 'numpy.ndarray'>
```

```
import pandas as pd
sr=pd.Series([19.5, 16.8, 22.78, 20.124, 18.1002])
df=sr.to_frame()
print(df)
print(type(df))
```

	0	19.5000
	1	16.8000
	2	22.7800
	3	20.1240
	4	18.1002
		<class 'pandas.core.frame.DataFrame'>

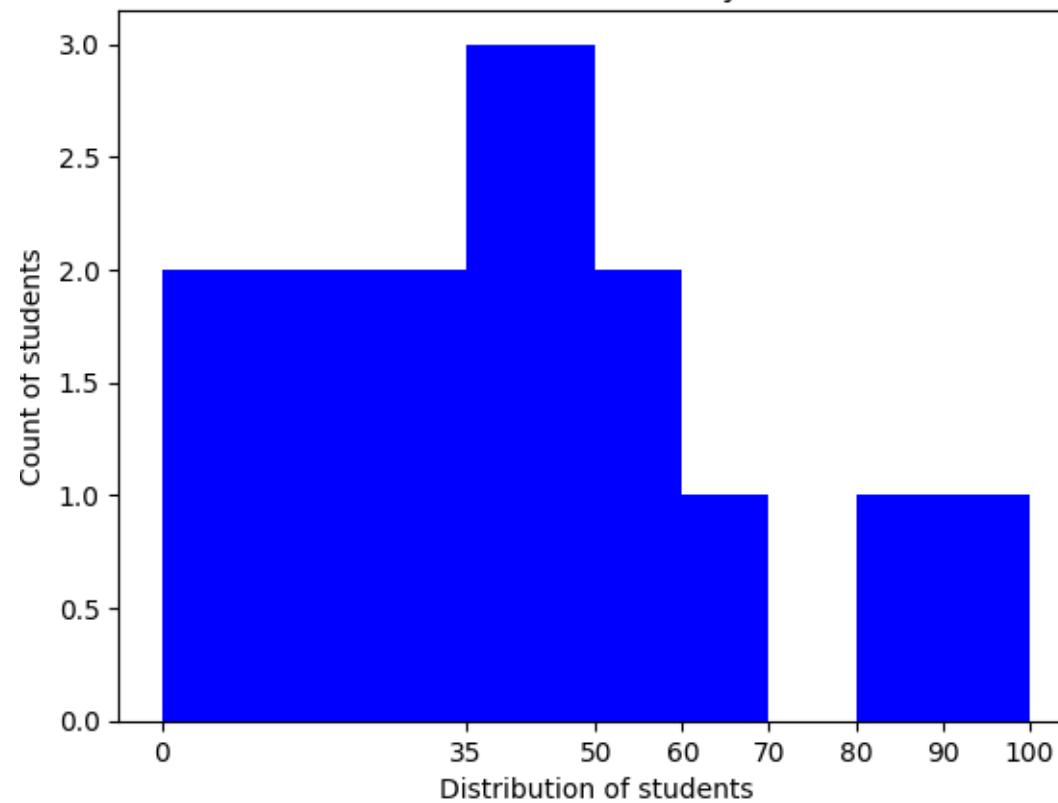
```
import matplotlib.pyplot as plt
import numpy as np
marks=[90,50,40,60,55,44,30,10,35,84]
grade_intervals=[0,35,50,60,70,80,90,100]
plt.title('Student Grade Analysis')
plt.hist(marks,grade_intervals,histtype='bar',rwidth=0.7,facecolor='b')
plt.xticks(grade_intervals)
plt.xlabel('Distribution of students')
plt.ylabel('Count of students')
plt.show()
```

### Student Grade Analysis

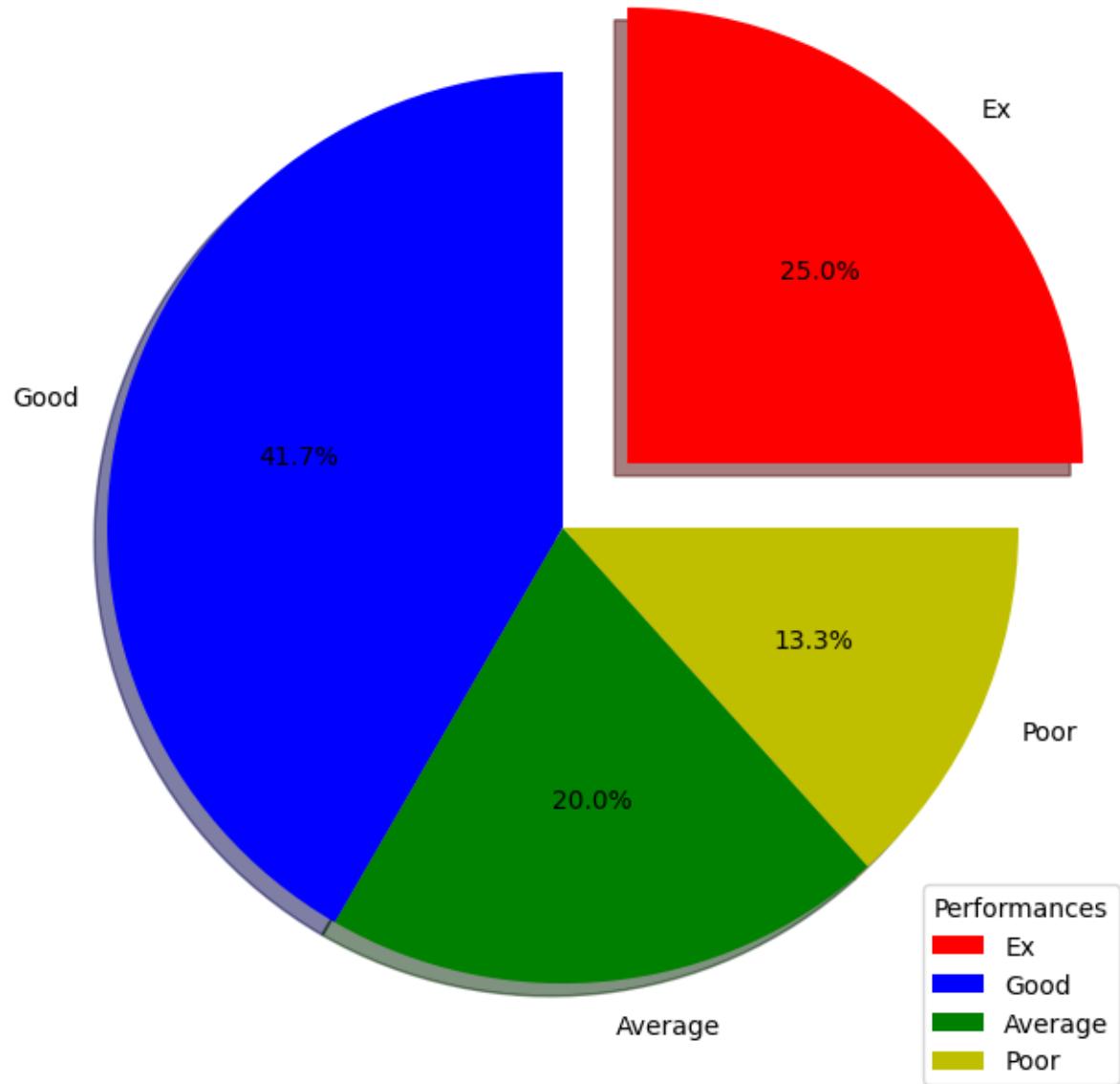


```
import matplotlib.pyplot as plt
import numpy as np
marks=[90,50,40,60,55,44,30,10,35,84]
grade_intervals=[0,35,50,60,70,80,90,100]
plt.title('Student Grade Analysis')
plt.hist(marks,grade_intervals,histtype='stepfilled',facecolor='b')
plt.xticks(grade_intervals)
plt.xlabel('Distribution of students')
plt.ylabel('Count of students')
plt.show()
```

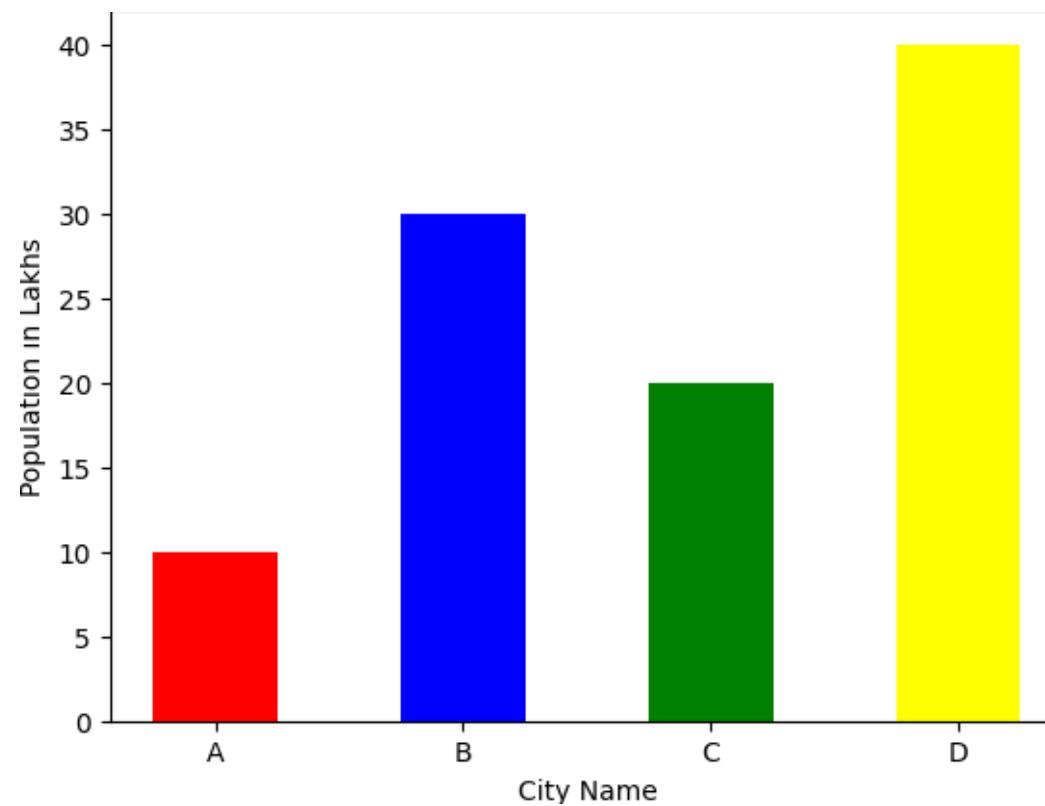
### Student Grade Analysis



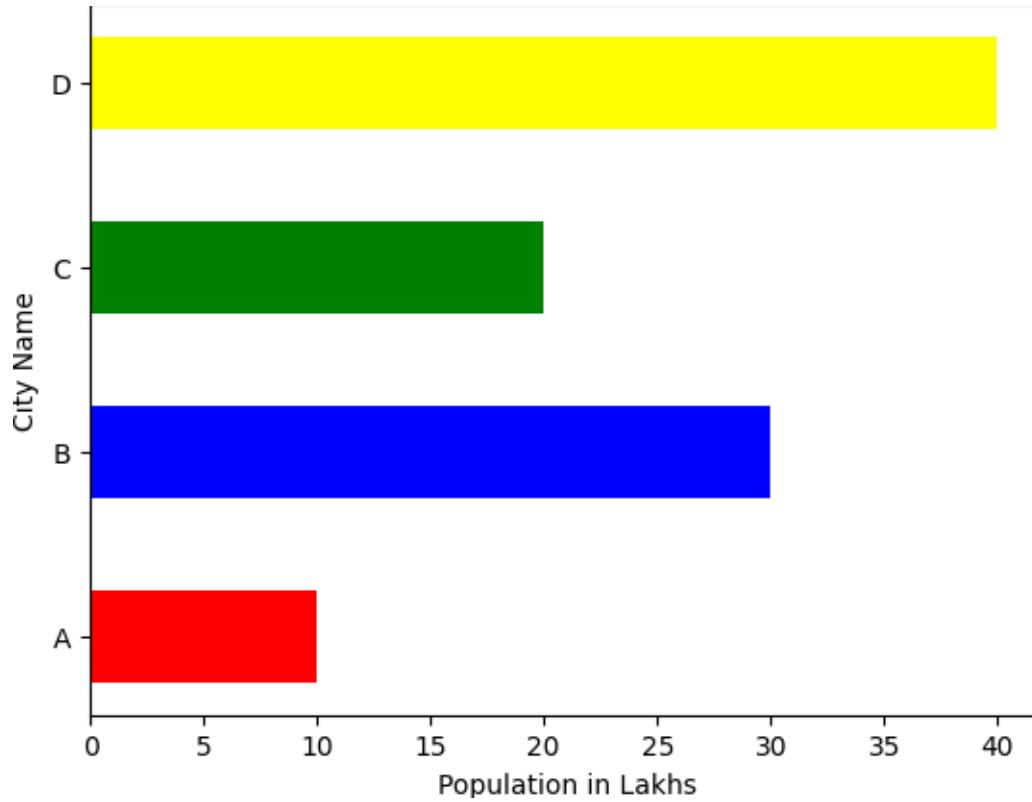
```
import matplotlib.pyplot as plt
import numpy as np
Student_Count=[15,25,12,8]
Student_Performance=['Ex','Good','Average','Poor']
plt.figure(figsize=(8,10))
plt.pie(Student_Count,labels=Student_Performance,startangle=0,explode=[0.2,0,0,0],
shadow=True,colors=['r','b','g','y'],autopct='%2.1f%%')
plt.legend(title='Performances')
plt.show()
```



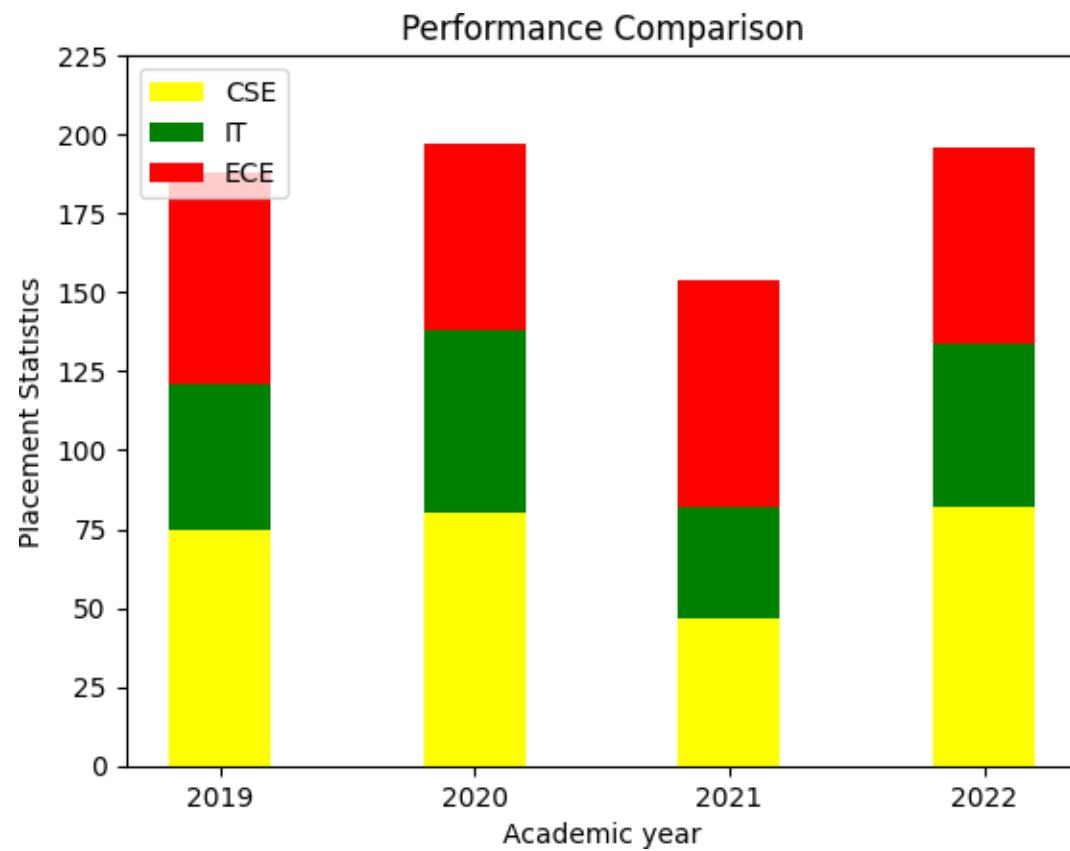
```
import matplotlib.pyplot as plt
import numpy as np
City=['A','B','C','D']
Polulation=[10,30,20,40]
c_lst=['red','blue','green','yellow']
plt.bar(City,Polulation,color=c_lst,width=0.5)
plt.xlabel('City Name')
plt.ylabel('Population in Lakhs')
plt.show()
```



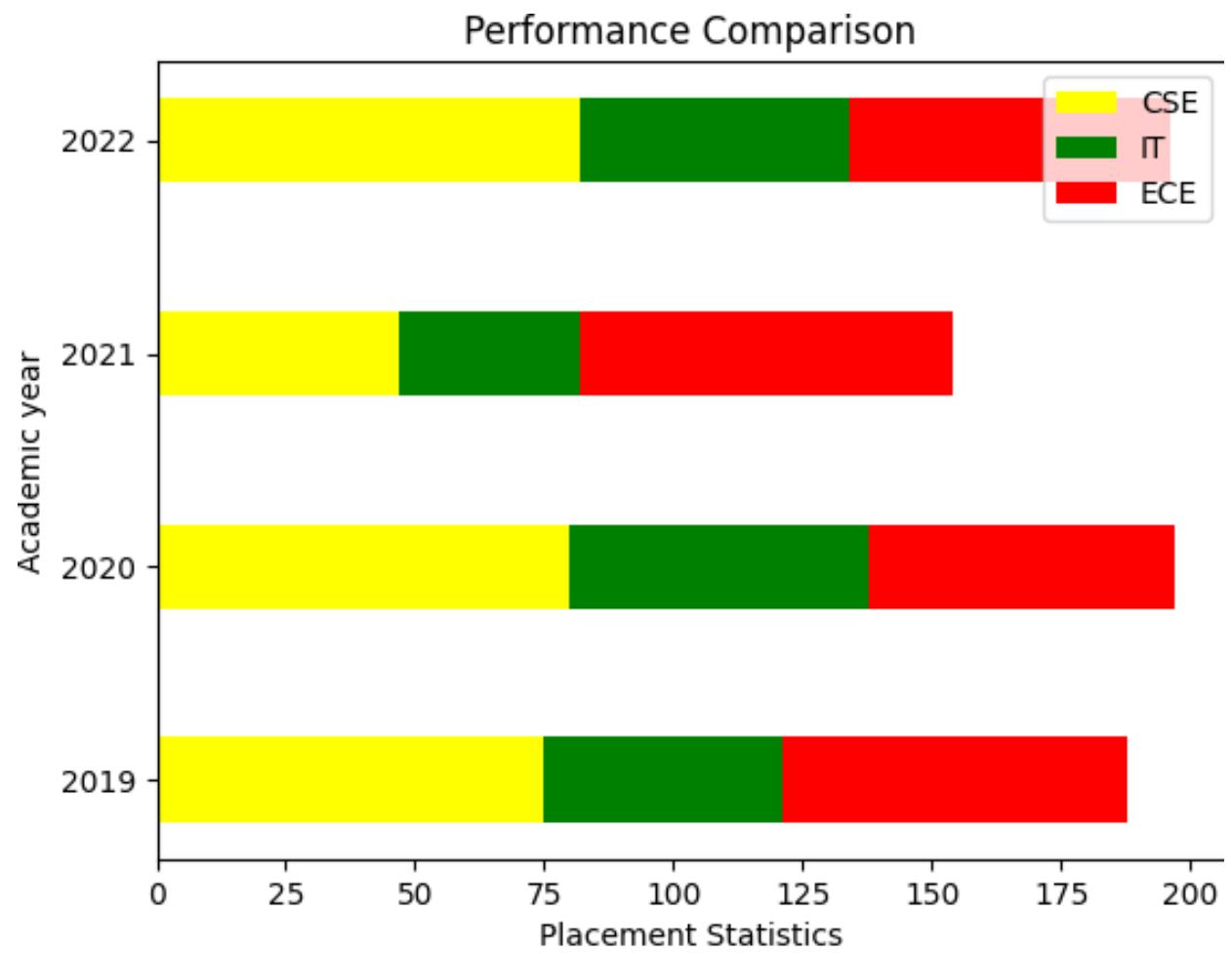
```
import matplotlib.pyplot as plt
import numpy as np
City=['A','B','C','D']
Polulation=[10,30,20,40]
c_lst=['red','blue','green','yellow']
plt.barh(City,Polulation,color=c_lst,height=0.5)
plt.ylabel('City Name')
plt.xlabel('Population in Lakhs')
plt.show()
```



```
import matplotlib.pyplot as plt
import numpy as np
Year=[2019,2020,2021,2022]
CSE=[75,80,47,82]
IT=[46,58,35,52]
ECE=[67,59,72,62]
ECE_Start=[CSE[i]+IT[i] for i in range(len(CSE))]
plt.title('Performance Comparison')
plt.bar(Year,CSE,width=0.4,color='yellow')
plt.bar(Year,IT,bottom=CSE,width=0.4,color='green')
plt.bar(Year,ECE,bottom=ECE_Start,width=0.4,color='red')
plt.ylim(0,225)
plt.xticks(Year)
plt.legend(['CSE','IT','ECE'],loc="upper left")
plt.xlabel('Academic year')
plt.ylabel('Placement Statistics')
plt.show()
```

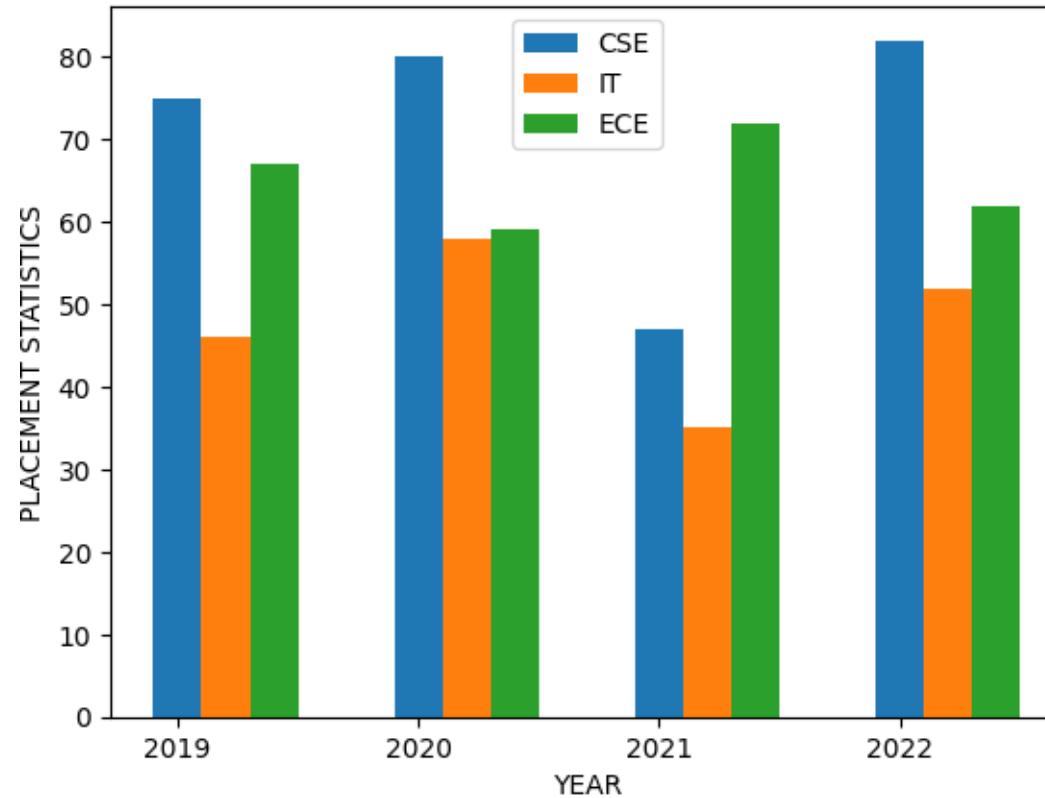


```
import matplotlib.pyplot as plt
import numpy as np
Year=[2019,2020,2021,2022]
CSE=[75,80,47,82]
IT=[46,58,35,52]
ECE=[67,59,72,62]
ECE_Start=[CSE[i]+IT[i] for i in range(len(CSE))]
plt.title('Performance Comparison')
plt.barh(Year,CSE,height=0.4,color='yellow')
plt.barh(Year,IT,left=CSE,height=0.4,color='green')
plt.barh(Year,ECE,left=ECE_Start,height=0.4,color='red')
plt.yticks(Year)
plt.legend(['CSE','IT','ECE'])
plt.ylabel('Academic year')
plt.xlabel('Placement Statistics')
plt.show()
```

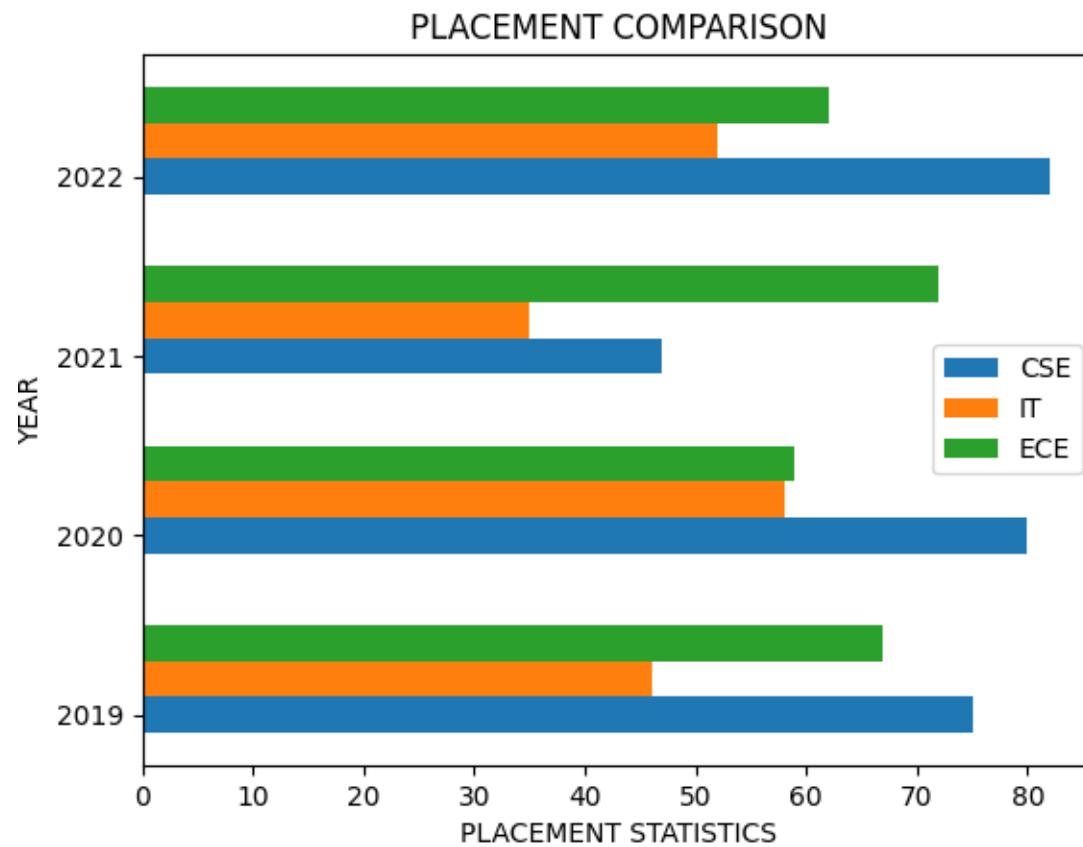


```
import matplotlib.pyplot as plt
import numpy as np
Year=[2019,2020,2021,2022]
CSE=[75,80,47,82]
IT=[46,58,35,52]
ECE=[67,59,72,62]
w=0.2
CSE_bar=np.arange(len(Year))
IT_bar=[i+w for i in CSE_bar]
ECE_bar=[i+w for i in IT_bar]
plt.bar(CSE_bar,CSE,width=w,label='CSE')
plt.bar(IT_bar,IT,width=w,label='IT')
plt.bar(ECE_bar,ECE,width=w,label='ECE')
plt.xticks(CSE_bar,Year)
plt.legend(['CSE','IT','ECE'])
plt.title('PLACEMENT COMPARISON')
plt.xlabel('YEAR')
plt.ylabel('PLACEMENT STATISTICS')
plt.ylim=(0,100)
plt.show()
```

### PLACEMENT COMPARISON



```
import matplotlib.pyplot as plt
import numpy as np
Year=[2019,2020,2021,2022]
CSE=[75,80,47,82]
IT=[46,58,35,52]
ECE=[67,59,72,62]
w=0.2
CSE_bar=np.arange(len(Year))
IT_bar=[i+w for i in CSE_bar]
ECE_bar=[i+w for i in IT_bar]
plt.barh(CSE_bar,CSE,height=w,label='CSE')
plt.barh(IT_bar,IT,height=w,label='IT')
plt.barh(ECE_bar,ECE,height=w,label='ECE')
plt.yticks(CSE_bar,Year)
plt.legend(['CSE','IT','ECE'])
plt.title('PLACEMENT COMPARISON')
plt.ylabel('YEAR')
plt.xlabel('PLACEMENT STATISTICS')
plt.xlim=(0,100)
plt.show()
```

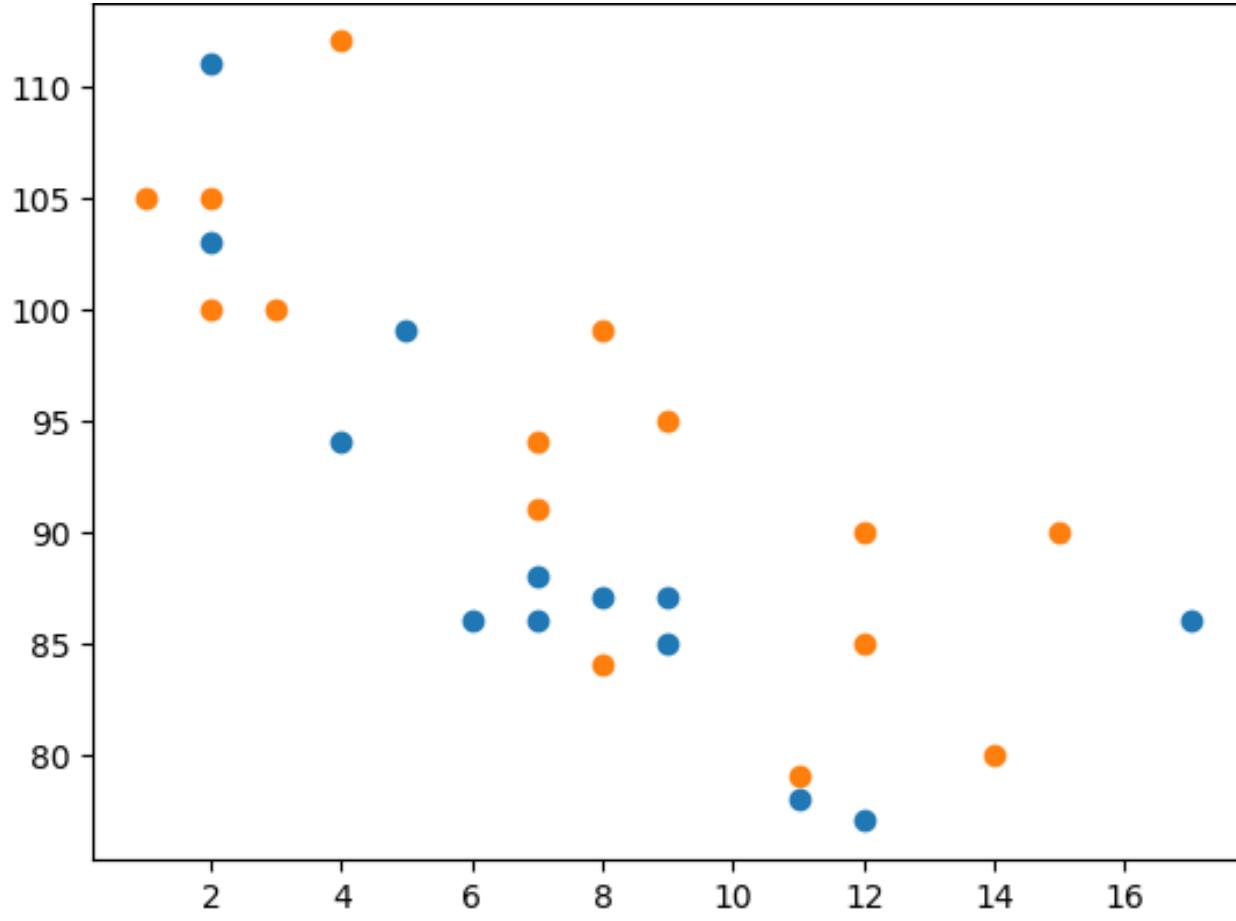


```
import matplotlib.pyplot as plt
import numpy as np

#day one, the age and speed of 13 cars:
x = np.array([5,7,8,7,2,17,2,9,4,11,12,9,6])
y = np.array([99,86,87,88,111,86,103,87,94,78,77,85,86])
plt.scatter(x, y)

#day two, the age and speed of 15 cars:
x = np.array([2,2,8,1,15,8,12,9,7,3,11,4,7,14,12])
y = np.array([100,105,84,105,90,99,90,95,94,100,79,112,91,80,85])
plt.scatter(x, y)

plt.show()
```



```
import matplotlib.pyplot as plt
import numpy as np

ypoints = np.array([3, 8, 1, 10])

plt.plot(ypoints, linestyle = 'dotted',marker="*")
plt.show()
```

