

```
In [1]: import numpy as np
```

```
In [2]: #np.where()
```

```
In [3]: x=np.arange(1,21).reshape(5,4)
```

```
In [4]: x
```

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

```
In [5]: x1=x.flatten()
x1
```

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

```
In [6]: a=np.where(x1%3==0)
a
```

```
Out[6]: (array([ 2,  5,  8, 11, 14, 17], dtype=int64),)
```

```
In [7]: x1
```

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

```
In [8]: a
```

```
Out[8]: (array([ 2,  5,  8, 11, 14, 17], dtype=int64),)
```

```
In [9]: x1[a]
```

```
Out[9]: array([ 3,  6,  9, 12, 15, 18])
```

```
In [10]: x
```

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

```
In [11]: q=np.where(x%3==0)
```

```
In [12]: q#rowfirst
         #colsecond
```

```
Out[12]: (array([0, 1, 2, 2, 3, 4], dtype=int64),
         array([2, 1, 0, 3, 2, 1], dtype=int64))
```

```
In [13]: q[0]
```

```
Out[13]: array([0, 1, 2, 2, 3, 4], dtype=int64)
```

```
In [14]: q[1]
```

```
Out[14]: array([2, 1, 0, 3, 2, 1], dtype=int64)
```

```
In [15]: x[q[0],q[1]]
```

```
Out[15]: array([ 3,  6,  9, 12, 15, 18])
```

```
In [16]: x[[0,1,2,2,3,4],[2,1,0,3,2,1]]
```

```
Out[16]: array([ 3,  6,  9, 12, 15, 18])
```

```
In [17]: np.where(x1%3==0,"div by 3","not div by 3")
```

```
Out[17]: array(['not div by 3', 'not div by 3', 'div by 3', 'not div by 3',  
               'not div by 3', 'div by 3', 'not div by 3', 'not div by 3',  
               'div by 3', 'not div by 3', 'not div by 3', 'div by 3',  
               'not div by 3', 'not div by 3', 'div by 3', 'not div by 3',  
               'not div by 3', 'div by 3', 'not div by 3', 'not div by 3'],  
              dtype='<U12')
```

```
In [18]: x1
```

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

```
In [19]: x3=np.where(x1%3==0,True,False)  
x3
```

```
Out[19]: array([False, False,  True, False, False,  True, False, False,  True,  
               False, False,  True, False, False,  True, False, False,  True,  
               False, False])
```

```
In [20]: x1[x3]
```

```
Out[20]: array([ 3,  6,  9, 12, 15, 18])
```

```
In [21]: x=np.arange(1,101).reshape(20,5)
x
```

```
Out[21]: 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],
 [31, 32, 33, 34, 35],
 [36, 37, 38, 39, 40],
 [41, 42, 43, 44, 45],
 [46, 47, 48, 49, 50],
 [51, 52, 53, 54, 55],
 [56, 57, 58, 59, 60],
 [61, 62, 63, 64, 65],
 [66, 67, 68, 69, 70],
 [71, 72, 73, 74, 75],
 [76, 77, 78, 79, 80],
 [81, 82, 83, 84, 85],
 [86, 87, 88, 89, 90],
 [91, 92, 93, 94, 95],
 [96, 97, 98, 99, 100]])
```

```
In [22]: np.where((x%3==0)&(x%5==0),np.nan,np.inf)
```

```
Out[22]: array([[inf, inf, inf, inf, inf],
               [inf, inf, inf, inf, inf],
               [inf, inf, inf, inf, nan],
               [inf, inf, inf, inf, inf],
               [inf, inf, inf, inf, inf],
               [inf, inf, inf, inf, nan],
               [inf, inf, inf, inf, inf],
               [inf, inf, inf, inf, inf],
               [inf, inf, inf, inf, nan],
               [inf, inf, inf, inf, inf],
               [inf, inf, inf, inf, inf],
               [inf, inf, inf, inf, nan],
               [inf, inf, inf, inf, inf],
               [inf, inf, inf, inf, inf],
               [inf, inf, inf, inf, nan],
               [inf, inf, inf, inf, inf],
               [inf, inf, inf, inf, inf],
               [inf, inf, inf, inf, nan],
               [inf, inf, inf, inf, inf],
               [inf, inf, inf, inf, inf]])
```

```
In [23]: x1=np.array([5,3,2,1,5,4,6])
```

```
In [24]: z=np.where(x1%2==0)
z
```

```
Out[24]: (array([2, 5, 6], dtype=int64),)
```

```
In [25]: x1.take(z)
```

```
Out[25]: array([[2, 4, 6]])
```

```
In [26]: np.take(x1,z)
```

```
Out[26]: array([[2, 4, 6]])
```

```
In [27]: #concatenation
```

```
In [28]: x1=np.array([[1,2,3],[4,5,6]])  
x1
```

```
Out[28]: array([[1, 2, 3],  
               [4, 5, 6]])
```

```
In [29]: x2=np.array([[11,12,13],[14,15,16]])  
x2
```

```
Out[29]: array([[11, 12, 13],  
               [14, 15, 16]])
```

```
In [30]: np.concatenate([x1,x2],axis=0)#row
```

```
Out[30]: array([[ 1,  2,  3],  
               [ 4,  5,  6],  
               [11, 12, 13],  
               [14, 15, 16]])
```

```
In [31]: np.concatenate([x1,x2],axis=1)#col
```

```
Out[31]: array([[ 1,  2,  3, 11, 12, 13],  
               [ 4,  5,  6, 14, 15, 16]])
```

```
In [32]: np.hstack([x1,x2])
```

```
Out[32]: array([[ 1,  2,  3, 11, 12, 13],  
               [ 4,  5,  6, 14, 15, 16]])
```

```
In [33]: np.vstack([x1,x2])
```

```
Out[33]: array([[ 1,  2,  3],  
               [ 4,  5,  6],  
               [11, 12, 13],  
               [14, 15, 16]])
```

```
In [34]: np.r_[x1,x2]#row r is not a function
```

```
Out[34]: array([[ 1,  2,  3],
                [ 4,  5,  6],
                [11, 12, 13],
                [14, 15, 16]])
```

```
In [35]: np.c_[x1,x2]#col r is not a function
```

```
Out[35]: array([[ 1,  2,  3, 11, 12, 13],
                [ 4,  5,  6, 14, 15, 16]])
```

```
In [36]: #np.save np.savez
```

```
In [37]: x=np.arange(1,21).reshape(4,5)
x
```

```
Out[37]: array([[ 1,  2,  3,  4,  5],
                [ 6,  7,  8,  9, 10],
                [11, 12, 13, 14, 15],
                [16, 17, 18, 19, 20]])
```

```
In [38]: np.save()
```

TypeError

Traceback (most recent call last)

~\AppData\Local\Temp\ipykernel_6620\2929270562.py in <module>

----> 1 np.save()

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

TypeError: _save_dispatcher() missing 2 required positional arguments: 'file' and 'arr'

```
In [ ]: x=np.arange(1,21).reshape(4,5)
x1=np.arange(1,21).reshape(4,5)
x2=np.arange(1,21).reshape(4,5)
```

```
In [ ]: np.save(r"C:\Users\user\Pictures\Saved Pictures.npy",x)
```

```
In [ ]: np.load(r"C:\Users\user\Pictures\Saved Pictures.npy")
```

```
In [ ]: "C:\Users\user\Pictures\Saved Pictures.npy"
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [39]: q1=np.array([[3,2,6,9,1],[1,21,3,4,6],[1,2,3,4,6]])  
q1
```

```
Out[39]: array([[ 3,  2,  6,  9,  1],  
               [ 1, 21,  3,  4,  6],  
               [ 1,  2,  3,  4,  6]])
```

```
In [40]: np.sort(q1,axis=0)
```

```
Out[40]: array([[ 1,  2,  3,  4,  1],  
               [ 1,  2,  3,  4,  6],  
               [ 3, 21,  6,  9,  6]])
```

```
In [41]: np.sort(q1,axis=1)
```

```
Out[41]: array([[ 1,  2,  3,  6,  9],  
               [ 1,  3,  4,  6, 21],  
               [ 1,  2,  3,  4,  6]])
```



```
In [42]: q2=np.array([3,4,6,2])
```

```
In [43]: np.argsort(q2)
```

```
Out[43]: array([3, 0, 1, 2], dtype=int64)
```

```
In [44]: #broadcasting
```

```
In [45]: #np.asarray(L)
```

```
In [46]: l=([1,2,3,4],[4,5,6,7])
```

```
In [47]: np.asarray(l)
```

```
Out[47]: array([[1, 2, 3, 4],  
               [4, 5, 6, 7]])
```

```
In [48]: np.array(l).tolist()
```

```
Out[48]: [[1, 2, 3, 4], [4, 5, 6, 7]]
```

```
In [49]: q=np.array([1,2,4,6])  
         r=np.array([5,3,2,1])
```

```
In [50]: np.intersect1d(q,r)
```

```
Out[50]: array([1, 2])
```

```
In [51]: np.sin(q)
```

```
Out[51]: array([ 0.84147098,  0.90929743, -0.7568025 , -0.2794155 ])
```

```
In [52]: np.cos(q)
```

```
Out[52]: array([ 0.54030231, -0.41614684, -0.65364362,  0.96017029])
```

```
In [53]: np.tan(q)
```

```
Out[53]: array([ 1.55740772, -2.18503986,  1.15782128, -0.29100619])
```

```
In [54]: a=np.array([1,"a",2,"b",3+2j],dtype='O')  
a
```

```
Out[54]: array([1, 'a', 2, 'b', (3+2j)], dtype=object)
```

```
In [55]: #practice
```

```
In [56]: t=np.arange(1,10).reshape(3,3)  
t
```

```
Out[56]: array([[1, 2, 3],  
               [4, 5, 6],  
               [7, 8, 9]])
```

```
In [57]: np.where(t%2==0)
```

```
Out[57]: (array([0, 1, 1, 2], dtype=int64), array([1, 0, 2, 1], dtype=int64))
```

```
In [58]: x1=t.flatten()  
x1
```

```
Out[58]: array([1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [59]: y1=np.where(x1%2==0)  
y1
```

```
Out[59]: (array([1, 3, 5, 7], dtype=int64),)
```

```
In [60]: #1
```

```
In [61]: x=np.arange(1,21).reshape(5,4)
x
```

```
Out[61]: array([[ 1,  2,  3,  4],
               [ 5,  6,  7,  8],
               [ 9, 10, 11, 12],
               [13, 14, 15, 16],
               [17, 18, 19, 20]])
```

```
In [62]: np.where(x%2==0)
```

```
Out[62]: (array([0, 0, 1, 1, 2, 2, 3, 3, 4, 4], dtype=int64),
          array([1, 3, 1, 3, 1, 3, 1, 3, 1, 3], dtype=int64))
```

```
In [63]: #2
```

```
In [64]: t=np.arange(1,10).reshape(3,3)
t
```

```
Out[64]: array([[1, 2, 3],
               [4, 5, 6],
               [7, 8, 9]])
```

```
In [65]: u=np.where(t%2==0)
u
```

```
Out[65]: (array([0, 1, 1, 2], dtype=int64), array([1, 0, 2, 1], dtype=int64))
```

```
In [66]: u[0]
```

```
Out[66]: array([0, 1, 1, 2], dtype=int64)
```

```
In [67]: u[1]
```

```
Out[67]: array([1, 0, 2, 1], dtype=int64)
```

```
In [68]: t[u[0],u[1]]
```

```
Out[68]: array([2, 4, 6, 8])
```

```
In [69]: t[[0,1,1,2],[1,0,2,1]]
```

```
Out[69]: array([2, 4, 6, 8])
```

```
In [70]: np.where(t%2==0,"div by 2","not div by 2")
```

```
Out[70]: array(['not div by 2', 'div by 2', 'not div by 2'],  
              ['div by 2', 'not div by 2', 'div by 2'],  
              ['not div by 2', 'div by 2', 'not div by 2']], dtype='<U12')
```

```
In [71]: s=np.where(t%2==0,"div","not div")  
s
```

```
Out[71]: array(['not div', 'div', 'not div'],  
              ['div', 'not div', 'div'],  
              ['not div', 'div', 'not div']], dtype='<U7')
```

```
In [72]: t
```

```
Out[72]: array([[1, 2, 3],  
               [4, 5, 6],  
               [7, 8, 9]])
```

```
In [73]: p=np.arange(3,9).reshape(3,2)  
p
```

```
Out[73]: array([[3, 4],  
               [5, 6],  
               [7, 8]])
```

```
In [74]: q=np.where((p%2==0),np.nan,np.inf)  
q
```

```
Out[74]: array([[inf, nan],  
               [inf, nan],  
               [inf, nan]])
```

```
In [75]: p[np.isnan(q)]=12  
p
```

```
Out[75]: array([[ 3, 12],  
               [ 5, 12],  
               [ 7, 12]])
```

```
In [76]: np.where((x%3==0)&(x%5==0),np.nan,np.inf)
```

```
Out[76]: array([[inf, inf, inf, inf],  
               [inf, inf, inf, inf],  
               [inf, inf, inf, inf],  
               [inf, inf, nan, inf],  
               [inf, inf, inf, inf]])
```

```
In [77]: np.where((p%3==0)&(p%5==0),np.nan)
```

```
-----  
ValueError                                Traceback (most recent call last)  
~\AppData\Local\Temp\ipykernel_6620\512750963.py in <module>  
----> 1 np.where((p%3==0)&(p%5==0),np.nan)  
  
<__array_function__ internals> in where(*args, **kwargs)  
  
ValueError: either both or neither of x and y should be given
```

```
In [78]: p[np.isinf(q)]=2  
p
```

```
Out[78]: array([[ 2, 12],  
               [ 2, 12],  
               [ 2, 12]])
```

```
In [79]: #1
p=np.arange(1,21).reshape(5,4)
p#doubt
```

```
Out[79]: array([[ 1,  2,  3,  4],
               [ 5,  6,  7,  8],
               [ 9, 10, 11, 12],
               [13, 14, 15, 16],
               [17, 18, 19, 20]])
```

```
In [80]: q=np.where(x%2==0)
q
```

```
Out[80]: (array([0, 0, 1, 1, 2, 2, 3, 3, 4, 4], dtype=int64),
          array([1, 3, 1, 3, 1, 3, 1, 3, 1, 3], dtype=int64))
```

```
In [96]: x[np.where(x%2==0)]
```

```
Out[96]: array([ 2,  4,  6,  8, 10, 12, 14, 16, 18, 20])
```

```
In [97]: p.take(q)
```

```
Out[97]: array([[1, 1, 2, 2, 3, 3, 4, 4, 5, 5],
               [2, 4, 2, 4, 2, 4, 2, 4, 2, 4]])
```

```
In [98]: np.take(p,q)
```

```
Out[98]: array([[1, 1, 2, 2, 3, 3, 4, 4, 5, 5],
               [2, 4, 2, 4, 2, 4, 2, 4, 2, 4]])
```

```
In [83]: #2
x1=np.array([5,3,2,1,5,4,6])
x1
```

```
Out[83]: array([5, 3, 2, 1, 5, 4, 6])
```

```
In [84]: z=np.where(x1%2==0)
z
```

```
Out[84]: (array([2, 5, 6], dtype=int64),)
```

```
In [85]: x1.take(z)
```

```
Out[85]: array([[2, 4, 6]])
```

```
In [86]: np.take(x1,z)
```

```
Out[86]: array([[2, 4, 6]])
```

```
In [88]: m=np.array([[2,3,4,5],[1,3,5,7]])
m
```

```
Out[88]: array([[2, 3, 4, 5],
               [1, 3, 5, 7]])
```

```
In [90]: n=np.array([[1,2,3,4],[2,3,4,5]])
n
```

```
Out[90]: array([[1, 2, 3, 4],
               [2, 3, 4, 5]])
```

```
In [93]: np.concatenate([m,n])
```

```
Out[93]: array([[2, 3, 4, 5],
               [1, 3, 5, 7],
               [1, 2, 3, 4],
               [2, 3, 4, 5]])
```

```
In [99]: #      m              n
#([ [2, 3, 4, 5] * ([ [1, 2, 3, 4]
# [1, 3, 5, 7]])    [2, 3, 4, 5]])

np.concatenate([m*n])
```

```
Out[99]: array([[ 2,  6, 12, 20],
               [ 2,  9, 20, 35]])
```

```
In [101]: np.concatenate([m+n])
```

```
Out[101]: array([[ 3,  5,  7,  9],
                [ 3,  6,  9, 12]])
```

```
In [104]: np.concatenate([m+n],axis=0)
```

```
Out[104]: array([[ 3,  5,  7,  9],
                [ 3,  6,  9, 12]])
```

```
In [105]: np.concatenate([m+n],axis=1)
```

```
Out[105]: array([[ 3,  5,  7,  9],
                [ 3,  6,  9, 12]])
```

```
In [106]: m
```

```
Out[106]: array([[2, 3, 4, 5],
               [1, 3, 5, 7]])
```

```
In [107]: n
```

```
Out[107]: array([[1, 2, 3, 4],
               [2, 3, 4, 5]])
```

```
In [111]: np.concatenate(m*n)
```

```
Out[111]: array([ 2,  6, 12, 20,  2,  9, 20, 35])
```



```
In [108]: np.concatenate([m*n],axis=0)
```

```
Out[108]: array([[ 2,  6, 12, 20],  
                [ 2,  9, 20, 35]])
```

```
In [110]: np.concatenate([m*n],axis=1)
```

```
Out[110]: array([[ 2,  6, 12, 20],  
                [ 2,  9, 20, 35]])
```

```
In [112]: np.concatenate([m,n],axis=0)
```

```
Out[112]: array([[2, 3, 4, 5],  
                [1, 3, 5, 7],  
                [1, 2, 3, 4],  
                [2, 3, 4, 5]])
```

```
In [113]: np.concatenate([m,n],axis=1)
```

```
Out[113]: array([[2, 3, 4, 5, 1, 2, 3, 4],  
                [1, 3, 5, 7, 2, 3, 4, 5]])
```

```
In [134]: x=np.arange(1,21).reshape(4,5)  
          x1=np.arange(1,21).reshape(4,5)  
          x2=np.arange(1,21).reshape(4,5)
```

```
In [135]: np.save(r"C:\Users\user\Pictures\Saved Pictures.file1",x)
```

```
In [136]: np.load(r"C:\Users\user\Pictures\Saved Pictures.file1.npy")
```

```
Out[136]: array([[ 1,  2,  3,  4,  5],  
                [ 6,  7,  8,  9, 10],  
                [11, 12, 13, 14, 15],  
                [16, 17, 18, 19, 20]])
```

```
In [137]: np.savez(r"C:\Users\user\Pictures\Saved Pictures.file2",x,x1,x2)
```

```
In [139]: x=np.load(r"C:\Users\user\Pictures\Saved Pictures.file2.npz")
```

In [140]: x

Out[140]: <numpy.lib.npyio.NpzFile at 0x197c03fd8b0>

In [141]: x.files

Out[141]: ['arr_0', 'arr_1', 'arr_2']

In [145]: x['arr_0']

Out[145]: array([[1, 2, 3, 4, 5],
[6, 7, 8, 9, 10],
[11, 12, 13, 14, 15],
[16, 17, 18, 19, 20]])

In [146]: x['arr_1']

Out[146]: array([[1, 2, 3, 4, 5],
[6, 7, 8, 9, 10],
[11, 12, 13, 14, 15],
[16, 17, 18, 19, 20]])

In [147]: x['arr_2']

Out[147]: array([[1, 2, 3, 4, 5],
[6, 7, 8, 9, 10],
[11, 12, 13, 14, 15],
[16, 17, 18, 19, 20]])

In [155]: q=np.array([[6,5,7],[2,2,4],[6,7,3]])
q

Out[155]: array([[6, 5, 7],
[2, 2, 4],
[6, 7, 3]])

```
In [156]: np.sort(q)
```

```
Out[156]: array([[5, 6, 7],  
                [2, 2, 4],  
                [3, 6, 7]])
```

```
In [157]: np.sort(q,axis=0)
```

```
Out[157]: array([[2, 2, 3],  
                [6, 5, 4],  
                [6, 7, 7]])
```

```
In [158]: np.sort(q,axis=1)
```

```
Out[158]: array([[5, 6, 7],  
                [2, 2, 4],  
                [3, 6, 7]])
```

```
In [160]: np.argsort(q)
```

```
Out[160]: array([[1, 0, 2],  
                [0, 1, 2],  
                [2, 0, 1]], dtype=int64)
```

```
In [161]: np.argsort(q,axis=0)
```

```
Out[161]: array([[1, 1, 2],  
                [0, 0, 1],  
                [2, 2, 0]], dtype=int64)
```

```
In [162]: np.argsort(q,axis=1)
```

```
Out[162]: array([[1, 0, 2],  
                [0, 1, 2],  
                [2, 0, 1]], dtype=int64)
```

```
In [ ]:
```

