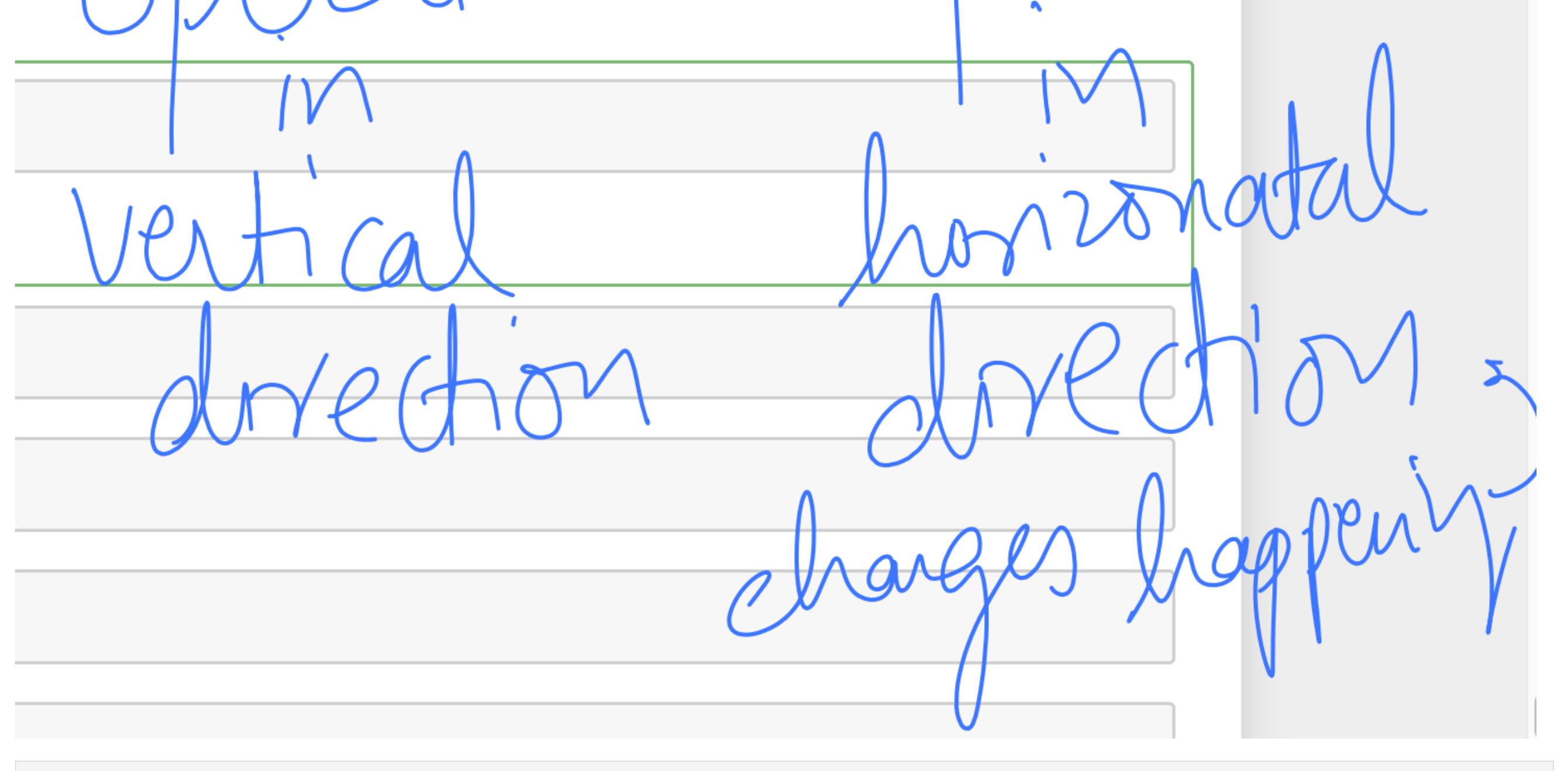
```
In [1]: print("Welcome to Numpy-3")

Welcome to Numpy-3

In [4]: import numpy as np

In [2]: !gdown 1c0ClC8SrPwJq5rrkyMKyPn80nyHcFikK

Downloading...
From: https://drive.google.com/uc?id=1c0ClC8SrPwJq5rrkyMKyPn80nyHcFikK
To: /Users/nikhilsanghi/Downloads/01_dsml-course-main-live/batches/Dec22_Adv_Dec22_Inter/Numpy_3/survey.txt
100%|████████████████████████████████████████| 2.55k/2.55k [00:00<00:00, 2.65MB/s]

In [5]: data=np.loadtxt("/Users/nikhilsanghi/Downloads/01_dsml-course-main-live/batches/Dec22_Adv_Dec22_Inter/Numpy_3/survey.txt",dtype="int")
        data

Out[5]: array([ 7, 10,  5, ...,  5,  9, 10])

In [9]: data.shape

Out[9]: (1167,)

In [11]: data

Out[11]: array([ 7, 10,  5, ...,  5,  9, 10])

In [8]: arr=np.empty(shape=data.shape,dtype="U21")
        arr

Out[8]: array(['', '', '', ..., '', '', ''], dtype='<U21')

In [10]: arr.shape

Out[10]: (1167,)

In [12]: data<7

Out[12]: array([False, False,  True, ...,  True, False, False])

In [14]: arr[data<7]="detractors"

In [17]: data

Out[17]: array([ 7, 10,  5, ...,  5,  9, 10])

In [15]: arr

Out[15]: array(['', '', 'detractors', ..., 'detractors', '', ''], dtype='<U21')

In [18]: arr[data>8]="promotors"

In [19]: arr

Out[19]: array(['', 'promotors', 'detractors', ..., 'detractors', 'promotors',
                'promotors'], dtype='<U21')

In [20]: arr[(data==8)|(data==7)]="passive"

In [21]: arr

Out[21]: array(['passive', 'promotors', 'detractors', ..., 'detractors',
                'promotors', 'promotors'], dtype='<U21')

In [22]: arr

Out[22]: array(['passive', 'promotors', 'detractors', ..., 'detractors',
                'promotors', 'promotors'], dtype='<U21')

In [23]: np.unique(arr)

Out[23]: array(['detractors', 'passive', 'promotors'], dtype='<U21')

In [24]: np.unique(arr,return_counts=True)

Out[24]: (array(['detractors', 'passive', 'promotors'], dtype='<U21'),
          array([332, 226, 609]))

In [ ]:

In [25]: a=np.array([1,2,3,4,5,6,7,8])

In [26]: a

Out[26]: array([1, 2, 3, 4, 5, 6, 7, 8])

In [27]: a%2==0

Out[27]: array([False,  True, False,  True, False,  True, False,  True])

In [28]: a[a%2==0]

Out[28]: array([2, 4, 6, 8])

In [29]: a[a%2==0] =10

In [30]: a

Out[30]: array([ 1, 10,  3, 10,  5, 10,  7, 10])

In [31]: #Q1
         a = np.array([1,2,3,4,5,6,7,8])
         print(a.ndim, a.shape)

1 (8,)

In [33]: #Q2
         a = np.array([1,2,3,4,5])
         b = np.array([8,7,6])
         # a[2 : ] = b[ : : -1]

In [34]: a[2 : ]

Out[34]: array([3, 4, 5])

In [35]: b[ : : -1]

Out[35]: array([6, 7, 8])

In [36]: a[2 : ] = b[ : : -1]

In [37]: a

Out[37]: array([1, 2, 6, 7, 8])

In [ ]:

In [38]: #2 d arrays

         a=np.array([[1,2,3],[4,5,6]])
         a

Out[38]: array([[1, 2, 3],
                [4, 5, 6]])

In [41]: # b=np.array([[1,2,3],[4,5,]])
         # b

In [42]: a

Out[42]: array([[1, 2, 3],
                [4, 5, 6]])

In [43]: a.ndim

Out[43]: 2

In [44]: a.shape

Out[44]: (2, 3)

In [45]: a.size

Out[45]: 6

In [46]: a.shape[0]*a.shape[1]

Out[46]: 6

In [47]: a

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

In [48]: len(a)

Out[48]: 2

In [49]: a

Out[49]: array([[1, 2, 3],
                [4, 5, 6]])

In [50]: a.reshape((3,2))

Out[50]: array([[1, 2],
                [3, 4],
                [5, 6]])

In [51]: a.reshape((5,1))

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_65494/3071720392.py in <module>
----> 1 a.reshape((5,1))

ValueError: cannot reshape array of size 6 into shape (5,1)

In [52]: a.reshape((6,1))

Out[52]: array([[1],
                [2],
                [3],
                [4],
                [5],
                [6]])

In [53]: a.reshape((6,0))

---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_65494/4019210259.py in <module>
----> 1 a.reshape((6,0))

ValueError: cannot reshape array of size 6 into shape (6,0)

In [54]: a.reshape((6,))

Out[54]: array([1, 2, 3, 4, 5, 6])

In [ ]:

In [55]: a=np.arange(1,13)
         a

Out[55]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])

In [56]: a.shape

Out[56]: (12,)

In [57]: a.reshape((1,12))

Out[57]: array([[ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12]])

In [58]: a.reshape((12,1))

Out[58]: array([[ 1],
                [ 2],
                [ 3],
                [ 4],
                [ 5],
                [ 6],
                [ 7],
                [ 8],
                [ 9],
                [10],
                [11],
                [12]])

In [59]: a.reshape((2,6))

Out[59]: array([[ 1,  2,  3,  4,  5,  6],
                [ 7,  8,  9, 10, 11, 12]])

In [60]: a.reshape((3,4))

Out[60]: array([[ 1,  2,  3,  4],
                [ 5,  6,  7,  8],
                [ 9, 10, 11, 12]])

In [61]: a.reshape((4,3))

Out[61]: array([[ 1,  2,  3],
                [ 4,  5,  6],
                [ 7,  8,  9],
                [10, 11, 12]])

In [63]: a.reshape((6,2))

Out[63]: array([[ 1,  2],
                [ 3,  4],
                [ 5,  6],
                [ 7,  8],
                [ 9, 10],
                [11, 12]])

In [64]: a.reshape((12,1))

Out[64]: array([[ 1],
                [ 2],
                [ 3],
                [ 4],
                [ 5],
                [ 6],
                [ 7],
                [ 8],
                [ 9],
                [10],
                [11],
                [12]])

In [66]: a.reshape((12,))

Out[66]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])

In [67]: a=np.arange(1,13).reshape((4,3))
         a

Out[67]: array([[ 1,  2,  3],
                [ 4,  5,  6],
                [ 7,  8,  9],
                [10, 11, 12]])

In [68]: a.reshape((-1,6))

Out[68]: array([[ 1,  2,  3,  4,  5,  6],
                [ 7,  8,  9, 10, 11, 12]])

In [69]: a.reshape((6,-1))

Out[69]: array([[ 1,  2],
                [ 3,  4],
                [ 5,  6],
                [ 7,  8],
                [ 9, 10],
                [11, 12]])

In [71]: # a.reshape((5,-1))

In [73]: # a.reshape((-1,-1))

In [74]: a

Out[74]: array([[ 1,  2,  3],
                [ 4,  5,  6],
                [ 7,  8,  9],
                [10, 11, 12]])

In [75]: a.T

Out[75]: array([[ 1,  4,  7, 10],
                [ 2,  5,  8, 11],
                [ 3,  6,  9, 12]])

In [76]: b=np.arange(12)

In [77]: b

Out[77]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11])

In [78]: b.T

Out[78]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11])

In [79]: c=b.reshape((1,12))
         c

Out[79]: array([[ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11]])

In [80]: c.T

Out[80]: array([[ 0],
                [ 1],
                [ 2],
                [ 3],
                [ 4],
                [ 5],
                [ 6],
                [ 7],
                [ 8],
                [ 9],
                [10],
                [11]])

In [81]: a

Out[81]: array([[ 1,  2,  3],
                [ 4,  5,  6],
                [ 7,  8,  9],
                [10, 11, 12]])

In [83]: a.reshape((a.size,))

Out[83]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])

In [84]: a.flatten()

Out[84]: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])

In [ ]:

In [ ]: # indexing and slicing

In [86]: a=np.arange(1,13).reshape((3,4))
         a

Out[86]: array([[ 1,  2,  3,  4],
                [ 5,  6,  7,  8],
                [ 9, 10, 11, 12]])

In [87]: # a[1][1]  #2 step process Method 1 NOT RECOMMENDED

Out[87]: 6

In [88]: a[ 1 ,  1 ]  # recommended and 1 step process # a[row,column]

Out[88]: 6

In [89]: a

Out[89]: array([[ 1,  2,  3,  4],
                [ 5,  6,  7,  8],
                [ 9, 10, 11, 12]])

In [90]: a[2,2]

Out[90]: 11

In [91]: a[-1,-2]

Out[91]: 11

In [92]: a[[1,2],[1,3]]

Out[92]: array([ 6, 12])

In [93]: a[[1,2,0],[1,3,3]]

Out[93]: array([ 6, 12,  4])

In [94]: a[(1,2,0),(1,3,3)]

Out[94]: array([ 6, 12,  4])

In [ ]:

In [95]: a[:,:]

Out[95]: array([[ 1,  2,  3,  4],
                [ 5,  6,  7,  8],
                [ 9, 10, 11, 12]])

In [96]: a[1:,:]

Out[96]: array([[ 5,  6,  7,  8],
                [ 9, 10, 11, 12]])

In [97]: a[1:,1:]

Out[97]: array([[ 6,  7,  8],
                [10, 11, 12]])

In [98]: a[1::,1::2]

Out[98]: array([[ 6,  8],
                [10, 12]])

In [105…: a[1::][1::2] # Homework

Out[105…: array([[ 9, 10, 11, 12]])

In [99]: a

Out[99]: array([[ 1,  2,  3,  4],
                [ 5,  6,  7,  8],
                [ 9, 10, 11, 12]])

In [100…: a[1::,::-1]

Out[100…: array([[ 8,  7,  6,  5],
                 [12, 11, 10,  9]])

In [ ]:

In [101…: a[::2,::-2]

Out[101…: array([[ 4,  2],
                 [12, 10]])

In [102…: a

Out[102…: array([[ 1,  2,  3,  4],
                 [ 5,  6,  7,  8],
                 [ 9, 10, 11, 12]])

In [103…: a[::-1,::-2]

Out[103…: array([[12, 10],
                 [ 8,  6],
                 [ 4,  2]])

In [104…: a[::-1][::-2]# Homework

Out[104…: array([[ 1,  2,  3,  4],
                 [ 9, 10, 11, 12]])

In [ ]:

In [106…: #Fancy indexing and masking
          a

Out[106…: array([[ 1,  2,  3,  4],
                 [ 5,  6,  7,  8],
                 [ 9, 10, 11, 12]])

In [107…: a<7

Out[107…: array([[ True,  True,  True,  True],
                 [ True,  True, False, False],
                 [False, False, False, False]])

In [108…: a[a<7]

Out[108…: array([1, 2, 3, 4, 5, 6])

In [110…: a[a%3==0]

Out[110…: array([ 3,  6,  9, 12])

In [111…: a[a>100]

Out[111…: array([], dtype=int64)

In [113…: a[a%1==0]

Out[113…: array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12])

In [ ]:

In [ ]: #universal functions

In [114…: a

Out[114…: array([[ 1,  2,  3,  4],
                 [ 5,  6,  7,  8],
                 [ 9, 10, 11, 12]])

In [116…: np.sum(a)

Out[116…: 78

In [ ]: np.sum()
```




```
In [117…: a

Out[117…: array([[ 1,  2,  3,  4],
                 [ 5,  6,  7,  8],
                 [ 9, 10, 11, 12]])

In [118…: np.sum(a)

Out[118…: 78

In [119…: np.sum(a,axis=0)

Out[119…: array([15, 18, 21, 24])

In [120…: np.sum(a,axis=1)

Out[120…: array([10, 26, 42])

In [121…: a

Out[121…: array([[ 1,  2,  3,  4],
                 [ 5,  6,  7,  8],
                 [ 9, 10, 11, 12]])

In [122…: np.min(a)

Out[122…: 1

In [123…: np.min(a,axis=0)

Out[123…: array([1, 2, 3, 4])

In [124…: np.min(a,axis=1)

Out[124…: array([1, 5, 9])

In [125…: a

Out[125…: array([[ 1,  2,  3,  4],
                 [ 5,  6,  7,  8],
                 [ 9, 10, 11, 12]])

In [ ]: np.sort(a,axis=0)

In [126…: a=np.array([[3,8,12,5],[91,1,3,4],[8,6,10,18]])

In [127…: a

Out[127…: array([[ 3,  8, 12,  5],
                 [91,  1,  3,  4],
                 [ 8,  6, 10, 18]])

In [128…: np.sort(a,axis=0)

Out[128…: array([[ 3,  1,  3,  4],
                 [ 8,  6, 10,  5],
                 [91,  8, 12, 18]])

In [129…: np.sort(a,axis=1)

Out[129…: array([[ 3,  5,  8, 12],
                 [ 1,  3,  4, 91],
                 [ 6,  8, 10, 18]])

In [130…: np.sort(a)

Out[130…: array([[ 3,  5,  8, 12],
                 [ 1,  3,  4, 91],
                 [ 6,  8, 10, 18]])

In [131…: np.sort(a,axis=-2)

Out[131…: array([[ 3,  1,  3,  4],
                 [ 8,  6, 10,  5],
                 [91,  8, 12, 18]])

In [ ]:

In [132…: a

Out[132…: array([[ 3,  8, 12,  5],
                 [91,  1,  3,  4],
                 [ 8,  6, 10, 18]])

In [134…: a.reshape(4,3)

Out[134…: array([[ 3,  8, 12],
                 [ 5, 91,  1],
                 [ 3,  4,  8],
                 [ 6, 10, 18]])

In [136…: a.reshape((4,3))

Out[136…: array([[ 3,  8, 12],
                 [ 5, 91,  1],
                 [ 3,  4,  8],
                 [ 6, 10, 18]])

In [137…: a

Out[137…: array([[ 3,  8, 12,  5],
                 [91,  1,  3,  4],
                 [ 8,  6, 10, 18]])

In [138…: a[::-2,::-1]

Out[138…: array([[18, 10,  6,  8],
                 [ 5, 12,  8,  3]])

In [ ]:
```