+ Code + Text

```
# Colab - https://colab.research.google.com/drive/1pZOgDCi7C2lLereMNt_kCxqyukGZ_4K-
```

```
import numpy as np
```

```
# fitbit
```

fit.txt

```
#date   step_count      mood    calories_burned hours_of_sleep  active
06-10-2017      5464    Neutral 181     5       Inactive
07-10-2017      6041    Sad     197     8       Inactive
08-10-2017      25      Sad     0       5       Inactive
09-10-2017      5461    Sad     174     4       Inactive
10-10-2017      6915    Neutral 223     5       Active
11-10-2017      4545    Sad     149     6       Inactive
12-10-2017      4340    Sad     140     6       Inactive
13-10-2017      1230    Sad     38      7       Inactive
14-10-2017      61      Sad     1       5       Inactive
15-10-2017      1258    Sad     40      6       Inactive
16-10-2017      3148    Sad     101     8       Inactive
17-10-2017      4687    Sad     152     5       Inactive
18-10-2017      4732    Happy   150     6       Active
19-10-2017      3519    Sad     113     7       Inactive
20-10-2017      1580    Sad     49      5       Inactive
21-10-2017      2822    Sad     86      6       Inactive
22-10-2017      181     Sad     6       8       Inactive
23-10-2017      3158    Neutral 99      5       Inactive
24-10-2017      4383    Neutral 143     4       Inactive
25-10-2017      3881    Neutral 125     5       Inactive
26-10-2017      4037    Neutral 129     6       Inactive
```

```
m1 = np.array([[1, 2, 3], [4, 5, 6]])
m1
```

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

```
m1.ndim
```

```
2
```

```
m1.shape
```

```
(2, 3)
```

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

```
b = len(a)
print(b)
```

```
3
```

```
a.shape
```

```
(3, 3)
```

```
m2 = np.arange(1, 13)
m2
```

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

```
m2.shape
```

```
(12,)
```

```
m2.reshape(4,3)
```

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

```
m2.reshape(4,4)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-12-b0f1cc8c31b9> in <module>
----> 1 m2.reshape(4,4)

ValueError: cannot reshape array of size 12 into shape (4,4)
```

SEARCH STACK OVERFLOW

```
m3 = m2.reshape(4,3)
m3.shape
```

```
(4, 3)
```

```
m2.reshape(12, 1) # 2D array
```

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

```
       [11],
       [12]])
```

```
m2.reshape(1, 12)
```

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

```
a = np.array([[1,2,3],[0,1,4]])
print(a.ndim, a.shape)
```

```
2 (2, 3)
```

```
# Transpose
```

```
np.arange(3).T # not applicable
```

```
array([0, 1, 2])
```

```
np.arange(3).reshape(1, 3)
```

```
array([[0, 1, 2]])
```

```
np.arange(3).reshape(1, 3).T
```

```
array([[0],
       [1],
       [2]])
```

```
A = np.arange(12).reshape(3, 4)
A
```

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

```
A.flatten()
```

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

```
A.reshape(12)
```

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

```
m1 = np.arange(1,10).reshape((3,3))
m1
```

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

```
m1[1][2]
```

```
6
```

```
m1[1, 2] # indexing a 2D array
```

```
6
```

```
m1 = np.array([100,200,300,400,500,600])
m1[[2,3,4,1,2,2]] # accessing multipl element
```

```
array([300, 400, 500, 200, 300, 300])
```

```
m1 = np.arange(9).reshape((3,3))
m1
```

```
array([[0, 1, 2],
       [3, 4, 5],
       [6, 7, 8]])
```

```
m1[[0,1,2],[0,1,2]]
```

```
array([0, 4, 8])
```

```
m1 = np.arange(12).reshape(3,4)
m1
```

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

```
m1[:2]
```

```
array([[0, 1, 2, 3],
       [4, 5, 6, 7]])
```

```
m1[:, :2]
```

```
array([[0, 1],
       [4, 5],
       [8, 9]])
```

```
m1
```

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

```
m1[1:3, 1:3] # m1[1:, 1:3]
```

```
array([[ 5,  6],
       [ 9, 10]])
```

```
m1[:, 1::2]
```

```
array([[ 1,  3],
       [ 5,  7],
       [ 9, 11]])
```

```
m1[:, [1,3]] # mix of slicing and indexing
```

```
array([[ 1,  3],
       [ 5,  7],
       [ 9, 11]])
```

```
m1 = np.array(
      [[0, 1, 2, 3],
      [4, 5, 6, 7],
      [8, 9, 10, 11]])

print(m1[[0,2], 0:2])
```

```
[[0 1]
 [8 9]]
```

```
m1 = np.arange(12).reshape(3, 4)
m1 < 6
```

```
array([[ True,  True,  True,  True],
       [ True,  True, False, False],
       [False, False, False, False]])
```

```
m1[m1<6] # notice its a 1D array output
```

```
array([0, 1, 2, 3, 4, 5])
```

```
arr = np.arange(3)
arr.sum()
```

```
3
```

```
np.sum(arr)
```

```
3
```

```
a = np.arange(12).reshape(3, 4)
a
```

```
array([[ 0,  1,  2,  3],
       [ 4,  5,  6,  7],
```

```
       [ 8,  9, 10, 11]])
```

```
np.sum(a)
```

```
66
```

```
np.sum(a, axis=1)
```

```
array([ 6, 22, 38])
```

```
np.sum(a, axis=0)
```

```
array([12, 15, 18, 21])
```

```
np.mean(a, axis=1)
```

```
array([1.5, 5.5, 9.5])
```

```
np.min(a)
```

```
0
```

```
arr = np.arange(10,0,-1)
arr = arr.reshape(5,2)
arr
```

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

```
arr.sum(axis=-1) # arr.sum(axis=1)
```

```
array([19, 15, 11,  7,  3])
```

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

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

```
np.any(a < b)
```

```
True
```

```
np.all(a < b)
```

```
False
```

```
a = np.array([1,2,3,4])
b = np.array([4,3,2,1])
a < b
```

```
array([ True,  True, False, False])
```

```
np.all(a<b)
```

```
False
```

```
np.any(~(a<b)) # wont be the same
```

```
True
```

```
arr = np.array([-3,4,27,34,-2, 0, -45,-11,4,0])
arr
```

```
array([ -3,   4,  27,  34,  -2,   0, -45, -11,   4,   0])
```

```
arr[arr > 0] = 1
arr[arr < 0] = -1
arr
```

```
array([-1,  1,  1,  1, -1,  0, -1, -1,  1,  0])
```

```
np.where(arr > 0, +1, -1)
```

```
array([-1,  1,  1,  1, -1, -1, -1, -1,  1, -1])
```

```
a = np.array([2,30,41,7,17,52])
a
```

```
array([ 2, 30, 41,  7, 17, 52])
```

```
np.sort(a)
```

```
array([ 2,  7, 17, 30, 41, 52])
```

```
a = np.arange(9,0,-1).reshape(3,3)
a
```

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

```
np.sort(a, axis=0)
```

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

```
np.sort(a, axis=1)
```

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

```
np.sort(a) # same as np.sort(a, axis=-1), pls remember this
# usually, by default, for most of functions, axis=0, axis=None
```

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

```
a = np.array([2,30,41,7,17,52])
np.argsort(a)
```

```
array([0, 3, 4, 1, 2, 5])
```

```
!gdown 1vk1Pu0djiYcrdc85yUXZ_Rqq2oZNcohd
```

```
Downloading...
From: https://drive.google.com/uc?id=1vk1Pu0djiYcrdc85yUXZ_Rqq2oZNcohd
To: /content/fit.txt
100% 3.43k/3.43k [00:00<00:00, 5.26MB/s]
```

```
data = np.loadtxt('fit.txt', dtype='str')
```

```
data[:5] # first 5 rows, or first five columns
```

```
array([['06-10-2017', '5464', 'Neutral', '181', '5', 'Inactive'],
       ['07-10-2017', '6041', 'Sad', '197', '8', 'Inactive'],
       ['08-10-2017', '25', 'Sad', '0', '5', 'Inactive'],
       ['09-10-2017', '5461', 'Sad', '174', '4', 'Inactive'],
       ['10-10-2017', '6915', 'Neutral', '223', '5', 'Active']],
      dtype='<U10')
```

```
data.shape
```

```
(96, 6)
```

```
data[0]
```

```
array(['06-10-2017', '5464', 'Neutral', '181', '5', 'Inactive'],
      dtype='<U10')
```

```
data.T[0]
```

```
array(['06-10-2017', '07-10-2017', '08-10-2017', '09-10-2017',
       '10-10-2017', '11-10-2017', '12-10-2017', '13-10-2017',
       '14-10-2017', '15-10-2017', '16-10-2017', '17-10-2017',
```

```
       '18-10-2017', '19-10-2017', '20-10-2017', '21-10-2017',
       '22-10-2017', '23-10-2017', '24-10-2017', '25-10-2017',
       '26-10-2017', '27-10-2017', '28-10-2017', '29-10-2017',
       '30-10-2017', '31-10-2017', '01-11-2017', '02-11-2017',
       '03-11-2017', '04-11-2017', '05-11-2017', '06-11-2017',
       '07-11-2017', '08-11-2017', '09-11-2017', '10-11-2017',
       '11-11-2017', '12-11-2017', '13-11-2017', '14-11-2017',
       '15-11-2017', '16-11-2017', '17-11-2017', '18-11-2017',
       '19-11-2017', '20-11-2017', '21-11-2017', '22-11-2017',
       '23-11-2017', '24-11-2017', '25-11-2017', '26-11-2017',
       '27-11-2017', '28-11-2017', '29-11-2017', '30-11-2017',
       '01-12-2017', '02-12-2017', '03-12-2017', '04-12-2017',
       '05-12-2017', '06-12-2017', '07-12-2017', '08-12-2017',
       '09-12-2017', '10-12-2017', '11-12-2017', '12-12-2017',
       '13-12-2017', '14-12-2017', '15-12-2017', '16-12-2017',
       '17-12-2017', '18-12-2017', '19-12-2017', '20-12-2017',
       '21-12-2017', '22-12-2017', '23-12-2017', '24-12-2017',
       '25-12-2017', '26-12-2017', '27-12-2017', '28-12-2017',
       '29-12-2017', '30-12-2017', '31-12-2017', '01-01-2018',
       '02-01-2018', '03-01-2018', '04-01-2018', '05-01-2018',
       '06-01-2018', '07-01-2018', '08-01-2018', '09-01-2018'],
      dtype='<U10')
```

```
data.T[1]
```

```
array(['5464', '6041', '25', '5461', '6915', '4545', '4340', '1230', '61',
       '1258', '3148', '4687', '4732', '3519', '1580', '2822', '181',
       '3158', '4383', '3881', '4037', '202', '292', '330', '2209',
       '4550', '4435', '4779', '1831', '2255', '539', '5464', '6041',
       '4068', '4683', '4033', '6314', '614', '3149', '4005', '4880',
       '4136', '705', '570', '269', '4275', '5999', '4421', '6930',
       '5195', '546', '493', '995', '1163', '6676', '3608', '774', '1421',
       '4064', '2725', '5934', '1867', '3721', '2374', '2909', '1648',
       '799', '7102', '3941', '7422', '437', '1231', '1696', '4921',
       '221', '6500', '3575', '4061', '651', '753', '518', '5537', '4108',
       '5376', '3066', '177', '36', '299', '1447', '2599', '702', '133',
       '153', '500', '2127', '2203'], dtype='<U10')
```

```
data.T # array of columns
```

```
date, step_count, mood, calories_burned, hours_of_sleep, activity_status = data.T
# unpacking
```

```
#learner doubt
arr = np.array([-3,4,27,34,-2, 0, -45,-11,4,0])
```

```
np.where(arr<0, -1, 0)
```

```
array([-1,  0,  0,  0, -1,  0, -1, -1,  0,  0])
```

```
np.where(arr > 0, 1, np.where(arr<0, -1, 0))
```

```
array([-1,  1,  1,  1, -1,  0, -1, -1,  1,  0])
```

```
a = np.array([1,2,3,4])
b = np.array([4,3,2,1])
~np.any(a < b)
```

```
False
```