

```
In [2]: pip install numpy
```

Requirement already satisfied: numpy in /home/guido/anaconda3/lib/python3.8/site-packages (1.19.2)
Note: you may need to restart the kernel to use updated packages.

```
In [2]: import numpy as np  
print(np.__version__)
```

1.19.2

```
In [3]: arr = np.arange(10)
```

```
In [4]: arr
```

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

```
In [5]: np.full((3, 3), True, dtype=bool)
```

```
Out[5]: array([[ True,  True,  True],  
               [ True,  True,  True],  
               [ True,  True,  True]])
```

```
In [6]: np.ones((3,3), dtype=bool)
```

```
Out[6]: array([[ True,  True,  True],  
               [ True,  True,  True],  
               [ True,  True,  True]])
```

```
In [8]: arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [9]: arr
```

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

```
In [10]: arr[arr % 2 == 1]
```

```
Out[10]: array([1, 3, 5, 7, 9])
```

```
In [11]: arr[arr % 2 == 1] = -1
```

```
In [12]: arr
```

```
Out[12]: array([ 0, -1,  2, -1,  4, -1,  6, -1,  8, -1])
```

```
In [13]: arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [14]: arr
```

```
Out[14]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
In [16]: arr = np.arange(10)
```

```
In [17]: out = np.where(arr % 2 == 1, -1, arr)
```

```
In [19]: print(arr)  
out
```

```
[0 1 2 3 4 5 6 7 8 9]
```

```
Out[19]: array([ 0, -1,  2, -1,  4, -1,  6, -1,  8, -1])
```

```
In [21]: arr
arr.reshape(2, -1)
```

```
Out[21]: array([[0, 1, 2, 3, 4],
               [5, 6, 7, 8, 9]])
```

```
In [29]: a = np.arange(10).reshape(2, -1)
a
b = np.repeat(1, 10).reshape(2, -1)
b
out = np.r_[a, b]
out
```

```
Out[29]: array([[0, 1, 2, 3, 4],
               [5, 6, 7, 8, 9],
               [1, 1, 1, 1, 1],
               [1, 1, 1, 1, 1]])
```

```
In [30]: np.vstack([a, b])
```

```
Out[30]: array([[0, 1, 2, 3, 4],
               [5, 6, 7, 8, 9],
               [1, 1, 1, 1, 1],
               [1, 1, 1, 1, 1]])
```

```
In [31]: a = np.arange(10).reshape(2, -1)
b = np.repeat(1, 10).reshape(2, -1)
np.hstack([a, b])
```

```
Out[31]: array([[0, 1, 2, 3, 4, 1, 1, 1, 1, 1],
               [5, 6, 7, 8, 9, 1, 1, 1, 1, 1]])
```

```
In [32]: a = np.array([1,2,3])
np.r_[np.repeat(a, 3), np.tile(a, 3)]
```

```
Out[32]: array([1, 1, 1, 2, 2, 2, 3, 3, 3, 1, 2, 3, 1, 2, 3])
```

```
In [33]: a = np.array([1,2,3,2,3,4,3,4,5,6])
b = np.array([7,2,10,2,7,4,9,4,9,8])
np.intersect1d(a,b)
```

```
Out[33]: array([2, 4])
```

```
In [34]: a = np.array([1,2,3,4,5])
b = np.array([5,6,7,8,9])
np.setdiff1d(a,b)
```

```
Out[34]: array([1, 2, 3, 4])
```

```
In [36]: a = np.array([1,2,3,2,3,4,3,4,5,6])
b = np.array([7,2,10,2,7,4,9,4,9,8])
np.where(a == b)
```

```
Out[36]: (array([1, 3, 5, 7]),)
```

```
In [39]: a = np.array([2, 6, 1, 9, 10, 3, 27])
a[(a >= 5) & (a <= 10)]
index = np.where((a >= 5) & (a <= 10))
a[index]
```

```
Out[39]: array([ 6,  9, 10])
```

```
In [40]: a = np.array([5, 7, 9, 8, 6, 4, 5])
b = np.array([6, 3, 4, 8, 9, 7, 1])
def maxx(x, y):
    """Get the maximum of two items"""
    if x >= y:
        return x
    else:
        return y
```

```
In [43]: pair_max = np.vectorize(maxx, otypes=[float])
pair_max(a, b)
```

```
Out[43]: array([6., 7., 9., 8., 9., 7., 5.])
```

```
In [47]: arr = np.arange(9).reshape(3,3)
arr
arr[:, [1,0,2]]
```

```
Out[47]: array([[1, 0, 2],
               [4, 3, 5],
               [7, 6, 8]])
```

```
In [51]: arr = np.arange(9).reshape(3,3)
arr
arr[[1,0,2], :]
```

```
Out[51]: array([[3, 4, 5],
               [0, 1, 2],
               [6, 7, 8]])
```

```
In [53]: arr = np.arange(9).reshape(3,3)
arr
```

```
Out[53]: array([[0, 1, 2],
               [3, 4, 5],
               [6, 7, 8]])
```

```
In [54]: arr[::-1]
```

```
Out[54]: array([[6, 7, 8],
               [3, 4, 5],
               [0, 1, 2]])
```

```
In [55]: arr = np.arange(9).reshape(3,3)
arr[:, ::-1]
```

```
Out[55]: array([[2, 1, 0],
               [5, 4, 3],
               [8, 7, 6]])
```

```
In [56]: arr = np.arange(9).reshape(3,3)
rand_arr = np.random.randint(low=5, high=10, size=(5,3)) + np.random.random
print(rand_arr)

[[9.47919107 9.68878174 9.67588037]
```

```
[9.78908822  7.55705915  8.23426875]
[6.23241923  9.82161284  7.77926444]
[9.38555112  8.73608952  6.39849528]
[8.64468528  5.35023774  5.66907055]]
```

```
In [57]: rand_arr = np.random.random((5,3))
np.set_printoptions(precision=3)
rand_arr[:4]
```

```
Out[57]: array([[0.646, 0.773, 0.848],
               [0.309, 0.755, 0.506],
               [0.268, 0.084, 0.6   ],
               [0.295, 0.211, 0.478]])
```

```
In [58]: rand_arr
```

```
Out[58]: array([[0.646, 0.773, 0.848],
               [0.309, 0.755, 0.506],
               [0.268, 0.084, 0.6   ],
               [0.295, 0.211, 0.478],
               [0.948, 0.705, 0.029]])
```

```
In [59]: np.random.seed(100)
rand_arr = np.random.random([3,3])/1e3
rand_arr
```

```
Out[59]: array([[5.434e-04, 2.784e-04, 4.245e-04],
               [8.448e-04, 4.719e-06, 1.216e-04],
               [6.707e-04, 8.259e-04, 1.367e-04]])
```

```
In [61]: np.set_printoptions(suppress=False)
rand_arr
np.set_printoptions(suppress=True, precision=6)
rand_arr
```

```
Out[61]: array([[0.000543, 0.000278, 0.000425],
               [0.000845, 0.000005, 0.000122],
               [0.000671, 0.000826, 0.000137]])
```

```
In [63]: a = np.arange(15)
a
np.set_printoptions(threshold=6)
a
```

```
Out[63]: array([ 0,  1,  2, ..., 12, 13, 14])
```

```
In [65]: url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris'
iris = np.genfromtxt(url, delimiter=',', dtype='object')
names = ('sepalength', 'sepalwidth', 'petallength', 'petalwidth', 'species')
iris[:3]
```

```
Out[65]: array([[b'5.1', b'3.5', b'1.4', b'0.2', b'Iris-setosa'],
               [b'4.9', b'3.0', b'1.4', b'0.2', b'Iris-setosa'],
               [b'4.7', b'3.2', b'1.3', b'0.2', b'Iris-setosa']], dtype=object)
```