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

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

In [2]:
```
2**a
```

Out[2]:
```
array([ 2,  4,  8, 16])
```

In [3]:
```
a + 1
```

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

In [4]:
```
b = np.ones(4) + 1
a - b
```

Out[4]:
```
array([-1.,  0.,  1.,  2.])
```

In [5]:
```
a*b
```

Out[5]:
```
array([ 2.,  4.,  6.,  8.])
```

In [6]:
```
j = np.arange(5)
2**(j+1)-j
```

Out[6]:
```
array([ 2,  3,  6, 13, 28])
```

In [7]:
```
a = np.arange(10000)
%timeit[i+1 for i in a]
```

```
100 loops, best of 3: 7.17 ms per loop
```

In [8]:
```
c = np.ones((3,3))
c*c
```

Out[8]:
```
array([[ 1.,  1.,  1.],
       [ 1.,  1.,  1.],
       [ 1.,  1.,  1.]])
```

In [9]:
```
c.dot(c)
```

Out[9]:
```
array([[ 3.,  3.,  3.],
       [ 3.,  3.,  3.],
       [ 3.,  3.,  3.]])
```

In [10]:
```
a=np.array([1,2,3,4])
b=np.array([4,2,2,4])
a==b
```

Out[10]:
```
array([False,  True, False,  True], dtype=bool)
```

In [11]:
```
c.dot(1)
```

Out[11]:
```
array([[ 1.,  1.,  1.],
       [ 1.,  1.,  1.],
       [ 1.,  1.,  1.]])
```

In [12]:
```
a>b
```

Out[12]:
```
array([False, False,  True, False], dtype=bool)
```

In [13]:
```
a = np.array([1,1,0,0],dtype=bool)
b=np.array([1,0,1,0],dtype=bool)
np.logical_or(a,b)
np.logical_and(a,b)
```

Out[13]:
```
array([ True, False, False, False], dtype=bool)
```

In [14]:
```
np.logival_or(a,b)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-14-be32e9421610> in <module>()
----> 1 np.logival_or(a,b)

AttributeError: 'module' object has no attribute 'logival_or'
```

In [15]:
```
np.logical_or(a,b)
```

Out[15]:
```
array([ True,  True,  True, False], dtype=bool)
```

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

Out[16]:
```
array([ 0.        ,  0.84147098,  0.90929743,  0.14112001, -0.7568025 ,
       -0.95892427, -0.2794155 ,  0.6569866 ,  0.98935825,  0.41211849])
```

In [17]:
```
np.log(a)
```

```
-c:1: RuntimeWarning: divide by zero encountered in log
```

Out[17]:
```
array([       -inf,  0.        ,  0.69314718,  1.09861229,  1.38629436,
        1.60943791,  1.79175947,  1.94591015,  2.07944154,  2.19722458])
```

In [18]:
```
np.exp(a)
```

Out[18]:
```
array([  1.00000000e+00,   2.71828183e+00,   7.38905610e+00,
         2.00855369e+01,   5.45981500e+01,   1.48413159e+02,
         4.03428793e+02,   1.09663316e+03,   2.98095799e+03,
         8.10308393e+03])
```

In [19]:
```
a=np.arange(4)
a + np.array([1,2])
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-19-9b701d0001d7> in <module>()
      1 a=np.arange(4)
----> 2 a + np.array([1,2])

ValueError: operands could not be broadcast together with shapes (4,) (2,)
```

In [20]:
```
a = np.triu(np.ones((3,3)),1)
a
```

Out[20]:
```
array([[ 0.,  1.,  1.],
       [ 0.,  0.,  1.],
       [ 0.,  0.,  0.]])
```

In [21]:
```
a.
```

```
  File "<ipython-input-21-a0d310e2b5e6>", line 1
    a.
      ^
SyntaxError: invalid syntax
```

In [22]:
```
T
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-22-8f898b22d33b> in <module>()
----> 1 T

NameError: name 'T' is not defined
```

In [23]:
```
a.T
```

Out[23]:
```
array([[ 0.,  0.,  0.],
       [ 1.,  0.,  0.],
       [ 1.,  1.,  0.]])
```

In [24]:
```
a=np.array([1,2,3,4])
b=np.array([4.2.2,4])
c=np.array([1,2,3,4])
np.array_equal(a,b)
```

```
  File "<ipython-input-24-d0d50ed71ae5>", line 2
    b=np.array([4.2.2,4])
                      ^
SyntaxError: invalid syntax
```

In [25]:
```
a=np.array([1,2,3,4])
b=np.array([4.2,2,4])
c=np.array([1,2,3,4])
np.array_equal(a,b)
```

Out[25]:
```
False
```

In [26]:
```
np.array_equal(a,c)
```

Out[26]:
```
True
```

In [27]:
```
help(np.allclose)
```

```
Help on function allclose in module numpy.core.numeric:

allclose(a, b, rtol=1e-05, atol=1e-08)
    Returns True if two arrays are element-wise equal within a tolerance.

    The tolerance values are positive, typically very small numbers.  The
    relative difference (`rtol` * abs(`b`)) and the absolute difference
    `atol` are added together to compare against the absolute difference
    between `a` and `b`.

    If either array contains one or more NaNs, False is returned.
    Infs are treated as equal if they are in the same place and of the same
    sign in both arrays.

    Parameters
    ----------
    a, b : array_like
        Input arrays to compare.
    rtol : float
        The relative tolerance parameter (see Notes).
    atol : float
        The absolute tolerance parameter (see Notes).

    Returns
    -------
    allclose : bool
        Returns True if the two arrays are equal within the given
        tolerance; False otherwise.

    See Also
    --------
    isclose, all, any
```

```
Notes
-----
If the following equation is element-wise True, then allclose returns
True.

 absolute(`a` - `b`) <= (`atol` + `rtol` * absolute(`b`))

The above equation is not symmetric in `a` and `b`, so that
`allclose(a, b)` might be different from `allclose(b, a)` in
some rare cases.

Examples
--------
>>> np.allclose([1e10,1e-7], [1.00001e10,1e-8])
False
>>> np.allclose([1e10,1e-8], [1.00001e10,1e-9])
True
>>> np.allclose([1e10,1e-8], [1.0001e10,1e-9])
False
>>> np.allclose([1.0, np.nan], [1.0, np.nan])
False
```

In [28]:
```
help(np.triu)
```

```
Help on function triu in module numpy.lib.twodim_base:

triu(m, k=0)
    Upper triangle of an array.

    Return a copy of a matrix with the elements below the `k`-th diagonal
    zeroed.

    Please refer to the documentation for `tril` for further details.

    See Also
    --------
    tril : lower triangle of an array

    Examples
    --------
    >>> np.triu([[1,2,3],[4,5,6],[7,8,9],[10,11,12]], -1)
    array([[ 1,  2,  3],
           [ 4,  5,  6],
           [ 0,  8,  9],
           [ 0,  0, 12]])
```

In [29]:
```
x = np.array([1,2,3,4])
np.sum(x)
```

Out[29]: 10

In [30]:
```
x.sum()
```

Out[30]: 10

In [31]:
```
x = np.array([[1, 1], [2, 2]])
x
```

Out[31]:
```
array([[1, 1],
       [2, 2]])
```

In [32]:
```
x.sum(axis=0)
```

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

In [33]:
```
x[:, 0].sum(), x[:, 1].sum()
```

Out[33]:
```
(3, 3)
```

In [34]:
```
x.sum(axis=1)
```

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

In [35]:
```
x[0, :].sum(), x[1, :].sum()
```

Out[35]:
```
(2, 4)
```

In [36]:
```
x = np.random.rand(2, 2, 2)
x.sum(axis=2)[0, 1]
```

Out[36]:
```
1.35521805501988
```

In [37]:
```
x[0, 1, :].sum()
```

Out[37]:
```
1.35521805501988
```

In [38]:
```
x = np.array([1, 2, 3, 1])
y = np.array([[1, 2, 3], [5, 6, 1]])
x.mean()
```

Out[38]: 1.75

In [39]:
```
np.median(x)
```

Out[39]: 1.5

In [40]:
```
np.median(y, axis=-1)
```

Out[40]: array([2., 5.])

In [41]:
```
x.std()
```

Out[41]: 0.82915619758884995

In [42]:
```
x = np.array([1, 3, 2])
x.min()
```

Out[42]: 1

In [43]:
```
x.max()
```

Out[43]: 3

In [44]:
```
x.argmin()
```

Out[44]: 0

In [45]:
```
x.argmax()
```

Out[45]: 1

In [46]:
```
np.all([True, True, False])
```

Out[46]: False

In [47]:
```
np.any([True, True, False])
```

Out[47]: True

In [48]:
```
a = np.zeros((100, 100))
np.any(a != 0)
```

Out[48]: False

In [49]:
```
np.all(a == a)
```

Out[49]: True

In [50]:
```
a = np.array([1, 2, 3, 2])
b = np.array([2, 2, 3, 2])
c = np.array([6, 4, 4, 5])
((a <= b) & (b <= c)).all()
```

Out[50]: True

In [51]:
```
a = (1,2,3,4,5)
a.sum()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-51-3116cf88f76c> in <module>()
      1 a = (1,2,3,4,5)
----> 2 a.sum()

AttributeError: 'tuple' object has no attribute 'sum'
```

In [52]:
```
a=([1,2,3,4,5])
a.sum()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-52-3099d76d1301> in <module>()
      1 a=([1,2,3,4,5])
----> 2 a.sum()

AttributeError: 'list' object has no attribute 'sum'
```

In [53]:
```
sum(a)
```

Out[53]: 15

In [54]:
```
cumsum(a)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-54-e80a1d329382> in <module>()
----> 1 cumsum(a)

NameError: name 'cumsum' is not defined
```

In [55]:
```
help(cumsum)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-55-8833e574381d> in <module>()
----> 1 help(cumsum)

NameError: name 'cumsum' is not defined
```

In [56]:
```
help(cumsum())
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-56-b1b446e2a7cb> in <module>()
----> 1 help(cumsum())

NameError: name 'cumsum' is not defined
```

In [57]:
```
!cat data/populations.txt
```

```
cat: data/populations.txt: No such file or directory
```

In [58]:
```
!cat kpgnarvaez/populations.txt
```

```
cat: kpgnarvaez/populations.txt: No such file or directory
```

In [59]:
```
a=np.tile(np.arange(0,40,10), (3,1))
```

In [60]:
```
a
```

Out[60]:
```
array([[ 0, 10, 20, 30],
       [ 0, 10, 20, 30],
       [ 0, 10, 20, 30]])
```

In [61]:
```
a=np.tile(np.arange(0,40,10), (3.1)).T
a
```

Out[61]:
```
array([ 0, 10, 20, 30,  0, 10, 20, 30,  0, 10, 20, 30])
```

In [62]:
```
a = np.tile(np.arange(0, 40, 10), (3, 1)).T
a
```

Out[62]:
```
array([[ 0,  0,  0],
       [10, 10, 10],
       [20, 20, 20],
       [30, 30, 30]])
```

In [63]:
```
b = np.array([0, 1, 2])
a + b
```

Out[63]:
```
array([[ 0,  1,  2],
       [10, 11, 12],
       [20, 21, 22],
       [30, 31, 32]])
```

In [64]:
```
a = np.ones((4, 5))
a[0] = 2  # we assign an array of dimension 0 to an array of dimension 1
a
```

Out[64]:
```
array([[ 2.,  2.,  2.,  2.,  2.],
       [ 1.,  1.,  1.,  1.,  1.],
       [ 1.,  1.,  1.,  1.,  1.],
       [ 1.,  1.,  1.,  1.,  1.]])
```

In [65]:
```
a = np.arange(0, 40, 10)
a.shape
```

Out[65]:
```
(4,)
```

In [66]:
```
a = a[:, np.newaxis]  # adds a new axis -> 2D array
a.shape
```

Out[66]:
```
(4, 1)
```

In [67]:
```
a
```

Out[67]:
```
array([[ 0],
       [10],
       [20],
       [30]])
```

In [68]:
```
a + b
```

Out[68]:
```
array([[ 0,  1,  2],
       [10, 11, 12],
       [20, 21, 22],
       [30, 31, 32]])
```

In [69]:
```
mileposts = np.array([0, 198, 303, 736, 871, 1175, 1475, 1544,
       1913, 2448])
distance_array = np.abs(mileposts - mileposts[:, np.newaxis])
distance_array
```

Out[69]:
```
array([[   0,  198,  303,  736,  871, 1175, 1475, 1544, 1913, 2448],
       [ 198,    0,  105,  538,  673,  977, 1277, 1346, 1715, 2250],
       [ 303,  105,    0,  433,  568,  872, 1172, 1241, 1610, 2145],
       [ 736,  538,  433,    0,  135,  439,  739,  808, 1177, 1712],
       [ 871,  673,  568,  135,    0,  304,  604,  673, 1042, 1577],
       [1175,  977,  872,  439,  304,    0,  300,  369,  738, 1273],
       [1475, 1277, 1172,  739,  604,  300,    0,   69,  438,  973],
       [1544, 1346, 1241,  808,  673,  369,   69,    0,  369,  904],
       [1913, 1715, 1610, 1177, 1042,  738,  438,  369,    0,  535],
       [2448, 2250, 2145, 1712, 1577, 1273,  973,  904,  535,    0]])
```

In [70]:
```
x, y = np.arange(5), np.arange(5)[:, np.newaxis]
distance = np.sqrt(x ** 2 + y ** 2)
distance
```

Out[70]:
```
array([[ 0.        ,  1.        ,  2.        ,  3.        ,  4.        ],
       [ 1.        ,  1.41421356,  2.23606798,  3.16227766,  4.12310563],
       [ 2.        ,  2.23606798,  2.82842712,  3.60555128,  4.47213595],
       [ 3.        ,  3.16227766,  3.60555128,  4.24264069,  5.        ],
       [ 4.        ,  4.12310563,  4.47213595,  5.        ,  5.65685425]])
```

In [71]:
```
plt.pcolor(distance)
plt.colorbar()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-71-64b30c232fc1> in <module>()
----> 1 plt.pcolor(distance)
      2 plt.colorbar()

NameError: name 'plt' is not defined
```

In [72]:
```
x, y = np.ogrid[0:5, 0:5]
x, y
```

Out[72]:
```
(array([[0],
        [1],
        [2],
        [3],
        [4]]), array([[0, 1, 2, 3, 4]]))
```

In [73]:
```
x.shape, y.shape
```

Out[73]:
```
((5, 1), (1, 5))
```

In [74]:
```
distance = np.sqrt(x ** 2 + y ** 2)
```

In [75]:
```
distance
```

Out[75]:
```
array([[ 0.        ,  1.        ,  2.        ,  3.        ,  4.        ],
       [ 1.        ,  1.41421356,  2.23606798,  3.16227766,  4.12310563],
       [ 2.        ,  2.23606798,  2.82842712,  3.60555128,  4.47213595],
       [ 3.        ,  3.16227766,  3.60555128,  4.24264069,  5.        ],
       [ 4.        ,  4.12310563,  4.47213595,  5.        ,  5.65685425]])
```

In [76]:
```
x, y = np.mgrid[0:4, 0:4]
x
```

Out[76]:
```
array([[0, 0, 0, 0],
       [1, 1, 1, 1],
       [2, 2, 2, 2],
       [3, 3, 3, 3]])
```

In [77]:
```
y
```

Out[77]:
```
array([[0, 1, 2, 3],
       [0, 1, 2, 3],
       [0, 1, 2, 3],
       [0, 1, 2, 3]])
```

In [78]:
```
a = np.array([[1, 2, 3], [4, 5, 6]])
a.ravel()
```

Out[78]:
```
array([1, 2, 3, 4, 5, 6])
```

In [79]:
```
a.T
```

Out[79]:
```
array([[1, 4],
       [2, 5],
       [3, 6]])
```

In [80]:
```
a.T.ravel()
```

Out[80]:
```
array([1, 4, 2, 5, 3, 6])
```

In [81]:
```
a.shape
```

Out[81]:
```
(2, 3)
```

In [82]:
```
b = a.ravel()
b = b.reshape((2, 3))
b
```

Out[82]:
```
array([[1, 2, 3],
       [4, 5, 6]])
```

In [83]:
```
a.reshape((2, -1))
```

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

In [84]:
```
b[0, 0] = 99
a
```

Out[84]:
```
array([[99,  2,  3],
       [ 4,  5,  6]])
```

In [85]:
```
a = np.zeros((3, 2))
b = a.T.reshape(3*2)
b[0] = 9
a
```

Out[85]:
```
array([[ 0.,  0.],
       [ 0.,  0.],
       [ 0.,  0.]])
```

In [86]:
```
z = np.array([1, 2, 3])
z
```

Out[86]:
```
array([1, 2, 3])
```

In [87]:
```
z[:, np.newaxis]
```

Out[87]:
```
array([[1],
       [2],
       [3]])
```

In [88]:
```
z[np.newaxis, :]
```

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

In [89]:
```
a = np.arange(4*3*2).reshape(4, 3, 2)
a.shape
```

Out[89]: (4, 3, 2)

In [90]:
```
a[0, 2, 1]
```

Out[90]: 5

In [91]:
```
b = a.transpose(1, 2, 0)
b.shape
```

Out[91]: (3, 2, 4)

In [92]:
```
b[2, 1, 0]
```

Out[92]: 5

In [93]:
```
b[2, 1, 0] = -1
a[0, 2, 1]
```

Out[93]: -1

In [94]:
```
a = np.arange(4)
a.resize((8,))
a
```

Out[94]: array([0, 1, 2, 3, 0, 0, 0, 0])

In [95]:
```
b = a
a.resize((4,))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-95-59edd3107605> in <module>()
      1 b = a
----> 2 a.resize((4,))

ValueError: cannot resize an array references or is referenced
by another array in this way.  Use the resize function
```

In [96]:
```
a = np.array([[4, 3, 5], [1, 2, 1]])
b = np.sort(a, axis=1)
b
```

Out[96]:
```
array([[3, 4, 5],
       [1, 1, 2]])
```

In [97]:
```
a.sort(axis=1)
a
```

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

In [98]:
```
a = np.array([4, 3, 1, 2])
j = np.argsort(a)
j
```

Out[98]:
```
array([2, 3, 1, 0])
```

In [99]:
```
a[j]
```

Out[99]:
```
array([1, 2, 3, 4])
```

In [100]:
```
a = np.array([4, 3, 1, 2])
j_max = np.argmax(a)
j_min = np.argmin(a)
j_max, j_min
```

Out[100]:
```
(0, 2)
```

In [101]:
```
a[a < 0] = 0
```

In [102]:
```
a = (1,2,3,4,5)
b = (True, False, True, False)

a.sort()
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-102-ab7780d06157> in <module>()
      2 b = (True, False, True, False)
      3
----> 4 a.sort()

AttributeError: 'tuple' object has no attribute 'sort'
```

In [103]:
```
a = np.array([4, 3, 1, 2])
a.sort()
```

In []:
```

```