Python Package Index

pypi.org

```
pip install SomePackage
pip install SomePackage-1.0-py2.py3-none-any.whl
pip show --files SomePackage
pip list --outdated
pip install --upgrade SomePackage
pip uninstall SomePackage
```

Предварительно собранные пакеты для Windows:

http://www.lfd.uci.edu/~gohlke/pythonlibs/

NumPy

numpy.org

Numpy = ***Num**erical **Py**thon*

- Поддержка многомерных массивов
- Эффективная работа с ними
- Дополнительные типы данных
- Ориентированность на научное применение

Создание массивов

```
>>> import numpy as np
>>> a = np.array([0, 1, 2, 3])
>>> a
array([0, 1, 2, 3])

>>> type(a)
numpy.ndarray
```

```
In [7]: std = range(100)

In [8]: %timeit [i**2 for i in std]
10000 loops, best of 3: 65.1 µs per loop

In [9]: n = np.arange(100)

In [10]: %timeit n**2
100000 loops, best of 3: 1.9 µs per loop
```

Создание массивов

```
>>> a = np.array([0, 1, 2, 3])
>>> b = np.array([[0, 1, 2], [3, 4, 5]])
>>> c = np.array([[[1], [2]], [[3], [4]]])
>>> a.ndim
1
>>> a.shape
(4,)
>>> b.ndim
2
>>> b.shape
(2, 3)
>>> c.ndim
3
>>> c.shape
(2, 2, 1)
```

Создание массивов

```
>>> a = np.arange(10)
>>> a
array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
>>> b = np.arange(1, 9, 2)
>>> b
array([1, 3, 5, 7])
```

```
>>> c = np.linspace(0, 1, 6)
>>> c
array([ 0. ,  0.2,  0.4,  0.6,  0.8,  1. ])
>>> d = np.linspace(0, 1, 5, endpoint=False)
>>> d
array([ 0. ,  0.2,  0.4,  0.6,  0.8])
```

Создание массивов

```
>>> a = np.ones((3, 3))
>>> a
array([[ 1.,  1.,  1.],
       [ 1.,  1.,  1.],
       [ 1.,  1.,  1.]])
>>> b = np.zeros((2, 2))
>>> b
array([[ 0.,  0.],
       [ 0.,  0.]])
>>> c = np.eye(3)
>>> c
array([[ 1.,  0.,  0.],
       [ 0.,  1.,  0.],
       [ 0.,  0.,  1.]])
>>> d = np.diag(np.array([1, 2, 3, 4]))
>>> d
array([[1, 0, 0, 0],
       [0, 2, 0, 0],
       [0, 0, 3, 0],
       [0, 0, 0, 4]])
```

Создание массивов

```
>>> n.empty((3,3))
array([[  6.94731466e-310,   7.35508808e-317,   7.17217651e-299],
       [  7.17217651e-299,   7.17217651e-299,   7.17334309e-299],
       [  7.16867678e-299,   7.16867678e-299,   3.95252517e-322]])
```

numpy.array(object, dtype = None, copy = True, order = None, subok = False, ndmin = 0)

Типы данных в Numpy

int8	8 бит	float16	16 бит
int16	16 бит	float32	32 бит
int32	32 бит (int на 32-битных платфорах)	float64	64 бит (float)
int64	64 бит (int на 64-битных платформах)	float96	96 бит, платформозависимый тип (longdouble)
bool	True/False (int)	float128	128 бит, платформозависимый тип (longdouble)
uint8	8 бит		
uint16	16 бит	complex64	2 x float32
uint32	32 бит	complex128	2 x float64
uint64	64 бит	complex192	2 x float96, платформозависимый тип
Sn	строка длиной n символов	complex256	2 x float128, платформозависимый тип

Типы данных в Numpy

```
>>> a = np.array([1, 2, 3])
>>> a.dtype
dtype('int64')

>>> b = np.array([1., 2., 3.])
>>> b.dtype
dtype('float64')


>>> c = np.array([1, 2, 3], dtype=float)
>>> c.dtype
dtype('float64')



>>> np.array('123456789',dtype='S5')
array(b'12345',
      dtype='|S5')
```

Изменение размерности

```
>>> a = np.array([np.arange(6),np.arange(10,16)])
array([[ 0,  1,  2,  3,  4,  5],
       [10, 11, 12, 13, 14, 15]])

>>> a.shape
(2, 6)

>>> a.reshape((3,4))
array([[ 0,  1,  2,  3],
       [ 4,  5, 10, 11],
       [12, 13, 14, 15]])
```

“Нарезка” массива

```
>>> a[2,3]
15

>>> a[:,:]
array([[ 0,  1,  2,  3],
       [ 4,  5, 10, 11],
       [12, 13, 14, 15]])

>>> a[:,1]
array([ 1,  5, 13])

>>> a[1,:]
array([ 4,  5, 10, 11])

>>> a[:2,1:3]
array([[ 1,  2],
       [ 5, 10]])
```

“Нарезка” массива, копии

```
>>> b = a[:,1]
array([ 1,  4, 11, 14])

>>> c = a[:,1].copy()
array([ 1,  4, 11, 14])

>>> b[0]=9999
>>> c[0]=333

>>> a
array([[   0, 9999,    2],
       [   3,    4,    5],
       [  10,   11,   12],
       [  13,   14,   15]])

>>> c
array([333,   4,  11,  14])

>>> np.may_share_memory(a,b)
True

>>> np.may_share_memory(a,c)
False
```

Доступ по списку

```
>>> a[[0,2]]
array([[   0, 9999,    2],
       [  10,   11,   12]])
```

Транспонирование

```
>>> a
array([[   0, 9999,    2],
       [   3,    4,    5],
       [  10,   11,   12],
       [  13,   14,   15]])

>>> a.T
array([[   0,    3,   10,   13],
       [9999,    4,   11,   14],
       [   2,    5,   12,   15]])
```

Операции с массивами

```
>>> a = np.array( [20,30,40,50] )
>>> b = np.arange( 4 )
>>> b
array([0, 1, 2, 3])

>>> c = a-b
>>> c
array([20, 29, 38, 47])

>>> b**2
array([0, 1, 4, 9])

>>> 10*np.sin(a)
array([ 9.12945251, -9.88031624,  7.4511316 , -2.62374854])

>>> a<35
array([True, True, False, False], dtype=bool)
```

Операции с массивами

```
>>> A = array( [[1,1],
...             [0,1]] )
>>> B = array( [[2,0],
...             [3,4]] )
>>> A*B                    # поэлементное умножение!!
array([[2, 0],
       [0, 4]])
>>> dot(A,B)               # матричное умножение
array([[5, 4],
       [3, 4]])
```

```
>>> np.mat(a)
matrix([[   0, 9999,    2],
        [   3,    4,    5],
        [  10,   11,   12],
        [  13,   14,   15]])
```

Операции с массивами

```
>>> a.sum()
10088

>>> a.min()
0

>>> a.max()
9999

>>> np.median(a)
10.5
```

Трансляция (broadcasting)

```
>>> a = np.tile(np.arange(0, 40, 10), (3, 1)).T
>>> a
array([[ 0,  0,  0],
       [10, 10, 10],
       [20, 20, 20],
       [30, 30, 30]])
>>> b = np.array([0, 1, 2])
>>> a + b
array([[ 0,  1,  2],
       [10, 11, 12],
       [20, 21, 22],
       [30, 31, 32]])
```

Трансляция (broadcasting)

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

“Сплющивание” ,“вытягивание”...?

```
>>> a = np.floor(10*np.random.random((3,4)))
>>> a
array([[ 4.,  5.,  7.,  1.],
       [ 9.,  4.,  9.,  5.],
       [ 6.,  4.,  0.,  3.]])

>>>r= a.ravel()
array([ 4.,  5.,  7.,  1.,  9.,  4.,  9.,  5.,  6.,  4.,  0.,  3.])

>>>f= a.flatten() # Копия!
array([ 4.,  5.,  7.,  1.,  9.,  4.,  9.,  5.,  6.,  4.,  0.,  3.])

>>> r[0]=111
>>>a
array([[ 111.,    5.,    7.,    1.],
       [   9.,    4.,    9.,    5.],
       [   6.,    4.,    0.,    3.]])
```

Объединение массивов

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

>>> b=np.array([11,22,33,44]).reshape(2,2)
array([[11, 22],
       [33, 44]])

>>> np.vstack((a,b))
array([[ 1,  2],
       [ 3,  4],
       [11, 22],
       [33, 44]])

>>> np.hstack((a,b))
array([[ 1,  2, 11, 22],
       [ 3,  4, 33, 44]])
```

Разделение массивов

```
>>> np.concatenate((a,b),axis=1)
array([[ 1,  2, 11, 22],
       [ 3,  4, 33, 44]])

>>> np.split(c,2)
[array([[ 1,  2, 11, 22]]), array([[ 3,  4, 33, 44]])]

>>> np.split(c,2,axis=1)
[array([[1, 2],
        [3, 4]]), array([[11, 22],
        [33, 44]])]
```

Подбор размера (“autoreshape”)

```
>>> a = np.arange(30)
>>> a.shape = 2,-1,3  # -1 — подобрать размер автоматически
>>> a.shape
(2, 5, 3)
>>> a
array([[[ 0,  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]]])
```

Изменение размера

```
>>> a = np.arange(4)
>>> a.resize((8,))
>>> a
array([0, 1, 2, 3, 0, 0, 0, 0])
```

Создание массивов

arange, array, copy, empty, empty_like, eye, fromfile, fromfunction, identity, linspace, logspace, mgrid, ogrid, ones, ones_like, r, zeros, zeros_like

Преобразования

ndarray.astype, atleast_1d, atleast_2d, atleast_3d, mat

Минипуляции

array_split, column_stack, concatenate, diagonal, dsplit, dstack, hsplit, hstack, ndarray.item, newaxis, ravel, repeat, reshape, resize, squeeze, swapaxes, take, transpose, vsplit, vstack

Условия

all, any, nonzero, where

Сортировки и т.п.

argmax, argmin, argsort, max, min, ptp, searchsorted, sort

Операции

choose, compress, cumprod, cumsum, inner, ndarray.fill, imag, prod, put, putmask, real, sum

Статистика

cov, mean, std, var

Лин. алгебра

cross, dot, outer, vdot, eig ...

Ввод / вывод

```
>>> data = np.random.random((5,2))
array([[ 0.16472754,  0.19252193],
       [ 0.39780631,  0.98838826],
       [ 0.53898721,  0.68676646],
       [ 0.03572282,  0.18779152],
       [ 0.95166912,  0.75086286]])

>>> np.savetxt('/tmp/qqq.txt', data, fmt="%.6G")

>>> data2 = np.loadtxt('/tmp/qqq.txt')
array([[ 0.164728 ,  0.192522 ],
       [ 0.397806 ,  0.988388 ],
       [ 0.538987 ,  0.686766 ],
       [ 0.0357228,  0.187792 ],
       [ 0.951669 ,  0.750863 ]])

>>> data2 = np.genfromtxt('/tmp/qqq.txt', usecols=1)
array([ 0.192522,  0.988388,  0.686766,  0.187792,  0.750863])
```

Ввод / вывод

```
>>> ds = np.DataSource()
>>> ds.exists('http://site.com/data.txt')
False
>>> data = ds.open('http://site.com/data.txt', mode='r')
```

Ссылки

numpy.org/devdocs/user/quickstart.html

scipy-lectures.org