Practical Computing for Scientists

Armin Sobhani

CSCI 2000U

UOIT – Fall 2015

UNIVERSITY OF ONTARIO
INSTITUTE OF TECHNOLOGY

Python

NumPy (Continue)

UNIVERSITY OF ONTARIO
INSTITUTE OF TECHNOLOGY

High Speed Recap

High Speed Recap

```
>>> import numpy as np
```

High Speed Recap

```
>>> import numpy as np
>>> a = np.array([1, 4, 5, 8], float)
```

High Speed Recap

```
>>> import numpy as np
>>> a = np.array([1, 4, 5, 8], float)
>>> a[:2]
array([ 1., 4.])
```

High Speed Recap

```
>>> import numpy as np
>>> a = np.array([1, 4, 5, 8], float)
>>> a[:2]
array([ 1., 4.])
>>> a[3]
8.0
```

High Speed Recap

```
>>> import numpy as np
>>> a = np.array([1, 4, 5, 8], float)
>>> a[:2]
array([ 1., 4.])
>>> a[3]
8.0
>>> a[0] = 5.
```

High Speed Recap

```
>>> import numpy as np
>>> a = np.array([1, 4, 5, 8], float)
>>> a[:2]
array([ 1., 4.])
>>> a[3]
8.0
>>> a[0] = 5.
>>> a
array([ 5., 4., 5., 8.])
```

High Speed Recap

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

High Speed Recap

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

High Speed Recap

```
>>> a = np.array([[1, 2, 3], [4, 5, 6]], float)
>>> a
array([[ 1., 2., 3.],
       [ 4., 5., 6.]])
>>> a[0,0]
1.0
```

High Speed Recap

```
>>> a = np.array([[1, 2, 3], [4, 5, 6]], float)
>>> a
array([[ 1., 2., 3.],
       [ 4., 5., 6.]])
>>> a[0,0]
1.0
>>> a[0:2,0:2]
array([[ 1.,  2.],
       [ 4.,  5.]])
```

High Speed Recap

```
>>> a = np.array([[1, 2, 3], [4, 5, 6]], float)
>>> a
array([[ 1., 2., 3.],
       [ 4., 5., 6.]])
>>> a[0,0]
1.0
>>> a[0:,0:2]
array([[ 1.,  2.],
       [ 4.,  5.]])
```

High Speed Recap

```
>>> a = np.array([[1, 2, 3], [4, 5, 6]], float)
>>> a
array([[ 1., 2., 3.],
       [ 4., 5., 6.]])
>>> a[0,0]
1.0
>>> a[:,0:2]
array([[ 1.,  2.],
       [ 4.,  5.]])
```

`in` for Testing Existence

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

`in` for Testing Existence

```
>>> a = np.array([[1, 2, 3], [4, 5, 6]], float)
>>> 2 in a
True
```

`in` for Testing Existence

```
>>> a = np.array([[1, 2, 3], [4, 5, 6]], float)
>>> 2 in a
True
>>> 0 in a
False
```

Reshaping Arrays

```
>>> a = np.array(range(10), float)
```

Reshaping Arrays

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

Reshaping Arrays

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

Reshaping Arrays

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

Reshaping Arrays

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

creates a new array

Reshaping Arrays

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

creates a new array

does not itself modify the original array

Reshaping Arrays

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

Reshaping Arrays

```
>>> a = np.array(range(10), float)
>>> a
array([ 0., 1., 2., 3., 4., 5., 6., 7., 8., 9.])
>>> a = a.reshape((5, 2))
>>> a
array([[ 0., 1.],
       [ 2., 3.],
       [ 4., 5.],
       [ 6., 7.],
       [ 8., 9.]])
>>> a.shape
(5, 2)
```

Creating a New Copy of an Array

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

Creating a New Copy of an Array

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

Creating a New Copy of an Array

```
>>> a = np.array([1, 2, 3], float)
>>> b = a
>>> c = a.copy()
```

Creating a New Copy of an Array

```
>>> a = np.array([1, 2, 3], float)
>>> b = a
>>> c = a.copy()
```

Creating a New Copy of an Array

```
>>> a = np.array([1, 2, 3], float)
>>> b = a
>>> c = a.copy()
>>> a[0] = 0
```

Creating a New Copy of an Array

```
>>> a = np.array([1, 2, 3], float)
>>> b = a
>>> c = a.copy()
>>> a[0] = 0
>>> a
array([0., 2., 3.])
```

Creating a New Copy of an Array

```
>>> a = np.array([1, 2, 3], float)
>>> b = a
>>> c = a.copy()
>>> a[0] = 0
>>> a
array([0., 2., 3.])
>>> b
array([0., 2., 3.])
```

Creating a New Copy of an Array

```
>>> a = np.array([1, 2, 3], float)
>>> b = a
>>> c = a.copy()
>>> a[0] = 0
>>> a
array([0., 2., 3.])
>>> b
array([0., 2., 3.])
>>> c
array([1., 2., 3.])
```

Creating a List from an Array

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

Creating a List from an Array

```
>>> a = np.array([1, 2, 3], float)
>>> a.tolist()
[1.0, 2.0, 3.0]
```

Creating a List from an Array

```
>>> a = np.array([1, 2, 3], float)
>>> a.tolist()
[1.0, 2.0, 3.0]
```

Creating a List from an Array

```
>>> a = np.array([1, 2, 3], float)
>>> a.tolist()
[1.0, 2.0, 3.0]
>>> list(a)
[1.0, 2.0, 3.0]
```

Creating a List from an Array

```
>>> a = np.array([1, 2, 3], float)
>>> a.tolist()
[1.0, 2.0, 3.0]
>>> list(a)
[1.0, 2.0, 3.0]
```

Converting to Binary String

```
>>> a = array([1, 2, 3], float)
```

Converting to Binary String

```
>>> a = array([1, 2, 3], float)
>>> s = a.tostring()
```

Converting to Binary String

```
>>> a = array([1, 2, 3], float)
>>> s = a.tostring()
```

Converting to Binary String

```
>>> a = array([1, 2, 3], float)
>>> s = a.tostring()
>>> s
'\x00\x00\x00\x00\x00\x00\xf0?\x00\x00\x00\x00\x00\x
00\x00@\x00\x00\x00\x00
\x00\x00\x08@'
```

Converting to Binary String

```
>>> a = array([1, 2, 3], float)
>>> s = a.tostring()
>>> s
'\x00\x00\x00\x00\x00\x00\xf0?\x00\x00\x00\x00\x00\x
00\x00@\x00\x00\x00\x00
\x00\x00\x08@'
>>> np.fromstring(s)
array([ 1., 2., 3.])
```

Converting to Binary String

```
>>> a = array([1, 2, 3], float)
>>> s = a.tostring()
>>> s
'\x00\x00\x00\x00\x00\x00\xf0?\x00\x00\x00\x00\x00\x
00\x00@\x00\x00\x00\x00
\x00\x00\x08@'
>>> np.fromstring(s)
array([ 1., 2., 3.])
```

Filling an Array with a Single Value

```
>>> a = array([1, 2, 3], float)
```

Filling an Array with a Single Value

```
>>> a = array([1, 2, 3], float)
>>> a
array([ 1., 2., 3.])
```

Filling an Array with a Single Value

```
>>> a = array([1, 2, 3], float)
>>> a
array([ 1., 2., 3.])
>>> a.fill(0)
```

Filling an Array with a Single Value

```
>>> a = array([1, 2, 3], float)
>>> a
array([ 1., 2., 3.])
>>> a.fill(0)
```

Filling an Array with a Single Value

```
>>> a = array([1, 2, 3], float)
>>> a
array([ 1., 2., 3.])
>>> a.fill(0)
>>> a
array([ 0., 0., 0.])
```

Transposing an Array

```
>>> a = np.array(range(9), float).reshape((3, 3))
```

Transposing an Array

```
>>> a = np.array(range(9), float).reshape((3, 3))
>>> a
array([[ 0.,  1.,  2.],
       [ 3.,  4.,  5.],
       [ 6.,  7.,  8.]])
```

Transposing an Array

```
>>> a = np.array(range(9), float).reshape((3, 3))
>>> a
array([[ 0.,  1.,  2.],
       [ 3.,  4.,  5.],
       [ 6.,  7.,  8.]])
>>> a.transpose()
array([[ 0.,  3.,  6.],
       [ 1.,  4.,  7.],
       [ 2.,  5.,  8.]])
```

Flattening a Multi-dimensional Array

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

Flattening a Multi-dimensional Array

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

Flattening a Multi-dimensional Array

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

Concatenating Arrays

```
>>> a = np.array([1,2], float)
```

Concatenating Arrays

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

Concatenating Arrays

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

Concatenating Arrays

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

Concatenating Arrays by Axis

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

Concatenating Arrays by Axis

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

Concatenating Arrays by Axis

```
>>> a = np.array([[1, 2], [3, 4]], float)
>>> b = np.array([[5, 6], [7,8]], float)
>>> np.concatenate((a,b))
array([[ 1., 2.],
       [ 3., 4.],
       [ 5., 6.],
       [ 7., 8.]])
```

Concatenating Arrays by Axis

```
>>> a = np.array([[1, 2], [3, 4]], float)
>>> b = np.array([[5, 6], [7,8]], float)
>>> np.concatenate((a,b))
array([[ 1., 2.],
       [ 3., 4.],
       [ 5., 6.],
       [ 7., 8.]])
>>> np.concatenate((a,b), axis=0)
array([[ 1., 2.],
       [ 3., 4.],
       [ 5., 6.],
       [ 7., 8.]])
```

Concatenating Arrays by Axis

```
>>> a = np.array([[1, 2], [3, 4]], float)
>>> b = np.array([[5, 6], [7,8]], float)
>>> np.concatenate((a,b))
array([[ 1., 2.],
       [ 3., 4.],
       [ 5., 6.],
       [ 7., 8.]])
>>> np.concatenate((a,b), axis=0)
array([[ 1., 2.],
       [ 3., 4.],
       [ 5., 6.],
       [ 7., 8.]])
>>> np.concatenate((a,b), axis=1)
array([[ 1., 2., 5., 6.],
       [ 3., 4., 7., 8.]])
```

Random Numbers

```
>>> np.random.seed(0)
```

Random Numbers

```
>>> np.random.seed(0)
>>> np.random.random() # [0.0, 1.0)
0.5488135039273248
```

Random Numbers

```
>>> np.random.seed(0)
>>> np.random.random() # [0.0, 1.0)
0.5488135039273248
>>> np.random.randint(5, 10) # [5, 10)
9
```

Random Numbers

```
>>> np.random.seed(0)
>>> np.random.random() # [0.0, 1.0)
0.5488135039273248
>>> np.random.randint(5, 10) # [5, 10)
9
>>> np.random.randint(5, 10)
5
```

Mathematical Constants

```
>>> np.pi
3.1415926535897931
```

Mathematical Constants

```
>>> np.pi
3.141592653589793
>>> np.e
2.718281828459045
```