Intro. to Numpy Module

1. Creating Arrays

```
In [1]: import numpy as np
```

```
In [2]: a = [1, 2, 3, 4 ]
        a

Out[2]: [1, 2, 3, 4]
```

```
In [4]: arr1 = np.array( a )
        arr1

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

```
In [5]: b = [ 5, 6, 7, 8 ]
```

```
In [6]: arr2d = np.array([a, b])
        arr2d

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

```
In [7]: arr2d.shape

Out[7]: (2, 4)
```

```
In [8]: A = np.arange(10)
        A

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

```
In [9]: B = np.arange(5, 50, 3)
        B

Out[9]: array([ 5,  8, 11, 14, 17, 20, 23, 26, 29, 32, 35, 38, 41, 44, 4
        7])
```

```
In [10]: C = np.arange(25).reshape(5,5)
         C
```

```
Out[10]: 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]])
```

```
In [11]: C.sum()
```

```
Out[11]: 300
```

```
In [12]: x = np.zeros(10)
         x
```

```
Out[12]: array([ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.])
```

```
In [13]: x.dtype
```

```
Out[13]: dtype('float64')
```

```
In [14]: y = np.zeros(25).reshape(5,5)
         y
```

```
Out[14]: array([[ 0.,  0.,  0.,  0.,  0.],
                [ 0.,  0.,  0.,  0.,  0.],
                [ 0.,  0.,  0.,  0.,  0.],
                [ 0.,  0.,  0.,  0.,  0.],
                [ 0.,  0.,  0.,  0.,  0.]])
```

```
In [15]: z = np.ones([5,5])
         z
```

```
Out[15]: array([[ 1.,  1.,  1.,  1.,  1.],
                [ 1.,  1.,  1.,  1.,  1.],
                [ 1.,  1.,  1.,  1.,  1.],
                [ 1.,  1.,  1.,  1.,  1.],
                [ 1.,  1.,  1.,  1.,  1.]])
```

2. Arrays Arithmetic

```
In [16]: arr = np.array([[1,2,3,4], [5,6,7,8]])
         arr
```

```
Out[16]: array([[1, 2, 3, 4],
                [5, 6, 7, 8]])
```

```
In [17]: arr + 10

Out[17]: array([[11, 12, 13, 14],
                [15, 16, 17, 18]])
```

```
In [18]: arr

Out[18]: array([[1, 2, 3, 4],
                [5, 6, 7, 8]])
```

```
In [19]: arr * arr

Out[19]: array([[ 1,  4,  9, 16],
                [25, 36, 49, 64]])
```

```
In [24]: arr

Out[24]: array([[1, 2, 3, 4],
                [5, 6, 7, 8]])
```

```
In [31]: arr ** 2

Out[31]: array([[ 1,  4,  9, 16],
                [25, 36, 49, 64]])
```

```
In [32]: arr

Out[32]: array([[1, 2, 3, 4],
                [5, 6, 7, 8]])
```

```
In [33]: arr ** 5

Out[33]: array([[    1,    32,   243,  1024],
                [ 3125,  7776, 16807, 32768]])
```

```
In [34]: 8 ** 5

Out[34]: 32768
```

```
In [35]: arr - arr

Out[35]: array([[0, 0, 0, 0],
                [0, 0, 0, 0]])
```

```
In [36]: # Transpose

         A = np.arange(25).reshape(5,5)
         A
```

```
Out[36]: 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]])
```

```
In [37]: A.T
```

```
Out[37]: array([[ 0,  5, 10, 15, 20],
                [ 1,  6, 11, 16, 21],
                [ 2,  7, 12, 17, 22],
                [ 3,  8, 13, 18, 23],
                [ 4,  9, 14, 19, 24]])
```

```
In [38]: B = np.arange(6).reshape(2,3)
         B
```

```
Out[38]: array([[0, 1, 2],
                [3, 4, 5]])
```

```
In [39]: Bt = B.T
         Bt
```

```
Out[39]: array([[0, 3],
                [1, 4],
                [2, 5]])
```

```
In [40]: np.dot(B, Bt)
```

```
Out[40]: array([[ 5, 14],
                [14, 50]])
```

3. Array Slices are Views

```
In [41]: A = np.arange(20)
         A
```

```
Out[41]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14,
         15, 16,
                17, 18, 19])
```

```
In [42]: a1 = A[:10]
         a1
```

```
Out[42]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

In [43]:

```
a1[:] = 100
a1
```

```
Out[43]: array([100, 100, 100, 100, 100, 100, 100, 100, 100, 100])
```

In [44]:

```
A
```

```
Out[44]: array([100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  10,  11,
         12,
                 13,  14,  15,  16,  17,  18,  19])
```

In [45]:

```
x = list(range(20))
x
```

```
Out[45]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
         19]
```

In [46]:

```
y = x[:10]
y
```

```
Out[46]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [47]:

```
y[0] = 100
y
```

```
Out[47]: [100, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [48]:

```
x
```

```
Out[48]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
         19]
```

In [49]:

```
A
```

```
Out[49]: array([100, 100, 100, 100, 100, 100, 100, 100, 100, 100,  10,  11,
         12,
                 13,  14,  15,  16,  17,  18,  19])
```

In [50]:

```
A = np.arange(20)
A
```

```
Out[50]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14,
         15, 16,
                17, 18, 19])
```

In [51]:

```
B = A.copy()
B
```

```
Out[51]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14,
         15, 16,
                17, 18, 19])
```

```
In [52]: B[0] = 100
         B

Out[52]: array([100,   1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,
         12,
                 13,  14,  15,  16,  17,  18,  19])

In [53]: A

Out[53]: array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14,
         15, 16,
                17, 18, 19])

In [54]: A = np.arange(25).reshape(5,5)
         A

Out[54]: 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]])

In [55]: A[0]

Out[55]: array([0, 1, 2, 3, 4])

In [56]: A[2][4]

Out[56]: 14

In [57]: A[2:4, 1:4]

Out[57]: array([[11, 12, 13],
                [16, 17, 18]])

In [58]: A

Out[58]: 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]])

In [59]: A[[1, 2, 4]]

Out[59]: array([[ 5,  6,  7,  8,  9],
                [10, 11, 12, 13, 14],
                [20, 21, 22, 23, 24]])
```

```
In [60]: A.sum()

Out[60]: 300
```

```
In [61]: A[[1, 2, 4]].sum()

Out[61]: 205
```

```
In [62]: A[[1, 2, 4]].shape

Out[62]: (3, 5)
```

4. Array Functions -- Universal Functions

```
In [66]: x = np.random.randint(1, 10)
         x

Out[66]: 5
```

```
In [67]: x = []
         y = []
         for i in range(10):
             x.append( np.random.randint(1, 10) )
             y.append( np.random.randint(1, 10) )

         x, y

Out[67]: ([3, 5, 2, 3, 9, 3, 5, 2, 2, 8], [8, 1, 9, 6, 6, 1, 9, 9, 7, 5])
```

```
In [68]: x = np.array( x )
         y = np.array( y )
         x, y

Out[68]: (array([3, 5, 2, 3, 9, 3, 5, 2, 2, 8]), array([8, 1, 9, 6, 6, 1,
         9, 9, 7, 5]))
```

```
In [69]: np.add( x, y )

Out[69]: array([11,  6, 11,  9, 15,  4, 14, 11,  9, 13])
```

```
In [70]: x

Out[70]: array([3, 5, 2, 3, 9, 3, 5, 2, 2, 8])
```

```
In [71]: y

Out[71]: array([8, 1, 9, 6, 6, 1, 9, 9, 7, 5])
```

```
In [72]: np.maximum(x, y)

Out[72]: array([8, 5, 9, 6, 9, 3, 9, 9, 7, 8])
```

```
In [73]: np.minimum(x, y)

Out[73]: array([3, 1, 2, 3, 6, 1, 5, 2, 2, 5])
```

```
In [74]: np.sqrt( np.minimum(x, y) )

Out[74]: array([ 1.73205081,  1.        ,  1.41421356,  1.73205081,  2.4494
         8974,
                 1.        ,  2.23606798,  1.41421356,  1.41421356,  2.2360
         6798])
```

```
In [75]: x

Out[75]: array([3, 5, 2, 3, 9, 3, 5, 2, 2, 8])
```

```
In [76]: np.square(x)

Out[76]: array([ 9, 25,  4,  9, 81,  9, 25,  4,  4, 64])
```

```
In [77]: x

Out[77]: array([3, 5, 2, 3, 9, 3, 5, 2, 2, 8])
```

```
In [78]: y

Out[78]: array([8, 1, 9, 6, 6, 1, 9, 9, 7, 5])
```

```
In [79]: np.divide( x, y )

Out[79]: array([0, 5, 0, 0, 1, 3, 0, 0, 0, 1])
```

```
In [80]: np.power( x, y )

Out[80]: array([   6561,       5,     512,     729,  531441,       3, 19531
         25,
                    512,     128,   32768])
```

```
In [81]: 3 ** 8

Out[81]: 6561
```

```
In [82]: 8 ** 5

Out[82]: 32768
```

```
In [83]: x, y

Out[83]: (array([3, 5, 2, 3, 9, 3, 5, 2, 2, 8]), array([8, 1, 9, 6, 6, 1,
         9, 9, 7, 5]))

In [84]: np.mod( x, y )

Out[84]: array([3, 0, 2, 3, 3, 0, 5, 2, 2, 3])

In [85]: x

Out[85]: array([3, 5, 2, 3, 9, 3, 5, 2, 2, 8])

In [86]: x.sort()
         x

Out[86]: array([2, 2, 2, 3, 3, 3, 5, 5, 8, 9])

In [89]: y.sort()
         y

Out[89]: array([1, 1, 5, 6, 6, 7, 8, 9, 9, 9])

In [90]: np.unique( x )

Out[90]: array([2, 3, 5, 8, 9])

In [91]: x = []
         y = []
         for i in range(10):
             x.append( np.random.randint(1, 10) )
             y.append( np.random.randint(1, 10) )

         x, y

Out[91]: ([2, 7, 2, 9, 3, 9, 3, 2, 1, 4], [2, 7, 8, 9, 5, 4, 3, 2, 5, 9])

In [92]: x = np.array( x )
         y = np.array( y )
         x, y

Out[92]: (array([2, 7, 2, 9, 3, 9, 3, 2, 1, 4]), array([2, 7, 8, 9, 5, 4,
         3, 2, 5, 9]))

In [95]: x

Out[95]: array([2, 7, 2, 9, 3, 9, 3, 2, 1, 4])

In [96]: y

Out[96]: array([2, 7, 8, 9, 5, 4, 3, 2, 5, 9])
```

```
In [97]: np.greater(x,y)

Out[97]: array([False, False, False, False, False,  True, False, False, Fal
         se, False], dtype=bool)

In [98]: b = np.greater(x,y)
         b

Out[98]: array([False, False, False, False, False,  True, False, False, Fal
         se, False], dtype=bool)

In [99]: b.all()

Out[99]: False

In [100]: b.any()

Out[100]: True

In [101]: x

Out[101]: array([2, 7, 2, 9, 3, 9, 3, 2, 1, 4])

In [102]: y

Out[102]: array([2, 7, 8, 9, 5, 4, 3, 2, 5, 9])

In [103]: np.equal(x, y )

Out[103]: array([ True,  True, False,  True, False, False,  True,  True, Fal
          se, False], dtype=bool)
```

Numpy Functions Reference

website = 'http://docs.scipy.org/doc/numpy/reference/ufuncs.html#available-ufuncs (http://docs.scipy.org/doc/numpy/reference/ufuncs.html#available-ufuncs)'

```
In [ ]:
```