IntroToNumPy

February 9, 2021

Let’s create some arrays...

```
[12]: import numpy as np
```

First, let’s make an array that is all floats. Tell me about numpy arrays...can they have variable types?

```
[108]: floats = np.array([1,2.5,6,8,5.25], dtype='float32')
       print(floats)
```

```
[1.   2.5  6.   8.   5.25]
```

Is the dtype=‘float32’ necessary?

Can auto-fill arrays with a starting value...0’s, 1’s, or decimal number. Can also create 2D arrays.

```
[15]: np.ones(15,dtype=int)
```

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

```
[16]: np.full((3,5),2.5)
```

```
[16]: array([[2.5, 2.5, 2.5, 2.5, 2.5],
             [2.5, 2.5, 2.5, 2.5, 2.5],
             [2.5, 2.5, 2.5, 2.5, 2.5]])
```

**np.arange is just like the range() function. Please create an array of all values divisible by 5 from 0 to 100

```
[109]: myArray = np.arange(0,105,5)
       print(myArray)
```

```
[  0   5  10  15  20  25  30  35  40  45  50  55  60  65  70  75  80  85
  90  95 100]
```

We mentioned creating an array of random numbers...let’s create an array of 10 random numbers.

```
[111]: np.random.rand(10)
```

```
[111]: array([0.61959393, 0.66844492, 0.75732316, 0.87499288, 0.87216972,
              0.93280896, 0.41726071, 0.31225399, 0.73367183, 0.5950818 ])
```

And an array of 10 random integers…

[113]:
```
np.random.randint(10,50,size=10)
```

[113]:
```
array([12, 31, 14, 43, 10, 23, 42, 18, 15, 25])
```

What's the difference between int16 and int32?

**Create a 1D array with 8 random numbers between 0 and 10

[123]:
```
x1 = np.random.randint(0,10,size=8)
x1
```

[123]:
```
array([7, 5, 7, 7, 9, 7, 0, 5])
```

**Create a 2D array (5x8) with random numbers between 0 and 10

[115]:
```
x2 = np.random.randint(0,10,size=(5,8))
x2
```

[115]:
```
array([[2, 9, 2, 7, 4, 5, 9, 1],
       [7, 6, 9, 7, 0, 7, 6, 4],
       [2, 1, 5, 7, 4, 5, 5, 1],
       [0, 2, 0, 0, 3, 4, 5, 5],
       [1, 7, 5, 4, 2, 6, 6, 9]])
```

**Create a 3D array (3x5x8) with random numbers between 0 and 15

[116]:
```
x3 = np.random.randint(15, size=(3,5,8))
x3
```

[116]:
```
array([[[12, 14, 14, 10,  6, 10, 12,  4],
        [13,  3,  0,  2,  2,  5,  7, 14],
        [ 9,  8, 11, 11, 13, 12, 11,  1],
        [ 4, 11, 14,  0,  5, 14,  5,  8],
        [ 5, 14,  3, 10,  6,  4, 14,  5]],

       [[ 1,  4,  0,  8, 14,  8,  3, 12],
        [ 5,  5,  3, 12,  2, 14, 14,  0],
        [ 9,  7, 10, 14,  0,  1,  3,  7],
        [ 2,  5, 11, 14,  2,  4,  1,  8],
        [ 2,  0, 13, 13, 10,  3, 14, 11]],

       [[ 1, 11, 10, 12,  8, 14,  2, 10],
        [ 4,  4, 12,  2,  4, 14,  2, 12],
        [13,  9,  8, 11, 11,  7, 12, 11],
        [ 8,  2,  1,  4,  5,  9, 13,  9],
        [ 8,  3,  2,  0, 13, 11, 11,  2]]])
```

**Using the 2D array from above, concatenate it with a = np.array[1,2,3,4,5,6,7,8]

```
[133]: a = np.array([1,2,3,4,5,6,7,8])
       y = np.vstack([x2,a])
       y
```

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

Let's remove the 6th row

```
[136]: y,extra = np.split(y,[5])
```

**and now, concatenate it to the end of your 3D array

```
[138]: x2 = x2.reshape((1,5,8))
       newX3 = np.vstack([x3,x2])
```

Let's try a horizontal stack on test, a 2d array

```
[144]: h1 = np.array([5,5,5,5,5])
       test = np.random.randint(10,size=(5,7))
       h1 = h1.reshape((5,1))
       newTest = np.hstack([test,h1])
       newTest
```

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

```
[ ]:
```