NUMPY Cheat Sheet:

- What is the purpose of the numpy library? NumPy is short for numerical python and is used to perform mathematical operations on arrays. It adds data structures that allow for efficient calculations with arrays and matrices and there is a large library of mathematical functions.
- What are numpy arrays and what can they do? NumPy is a fundamental package for scientific computing in python. It provides a multidimensional array object, various derived objects, and an assortment of routines. NumPy allows mathematical, logical, shape manipulation, sorting, selecting, linear algebra, statistics, random simulation, etc. It acts a bit like a calculator for arrays and matrices. Numpy is notable to convert anything that is a string. NumPy can work with homogeneous data including integers, booleans, floats, etc. However, it cannot work with heterogenous data or mixed data types.
- Approaches for slicing a numpy array:
  - → Lets say you have a list of values in a 1-D array: [6, 10, 7, 2, 16, 5, 8, 13, 12]
    You only want the 3rd - 7th elements: array = [6, 10, 7, 2, 16, 5, 8, 13, 12]
    print (array[2:7])
    ↳ [7, 2, 16, 5, 8]
    #Note: number of elements starts at zero and ending element is uninclusive.
  - → Lets say you have a multidimensional array: array = [[1, 3, 5, 7, 9]
    [2, 4, 6, 8, 10]]
    You want the 3rd element in the 2nd row
    print (array[2, 1])
    ↳ [6]
    OR
    You want the 1st - 3rd element in the 1st row
    print (array[0:3, 0]
    ↳ [1, 3, 5]
- How to create a numpy array from scratch: There are many different ways to do this but one is by using np.arange:
  - Syntax: np.arange(start, stop, step)
    ex. np.arange(10)
    ↳ array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
    np.arange(1, 5)
    ↳ array([1, 2, 3, 4])
    np.arang(1, 2, 0.2)
    ↳ array([1., 1.2, 1.4, 1.6, 1.8])

Another way to do it is np.linspace:

- Syntax: np.linspace(start, stop, # of elements)
  ex. np.linspace(1., 4., 6)
  ↳ array([1., 1.6, 2.2, 2.8, 3.4, 4.])

- Helpful Numpy functions:

1. np.zeros((3,3)): Creates an array of zeros by giving it the number of rows and columns.

   Output: [[0.,0.,0.]
   [0.,0.,0.]
   [0.,0.,0.]]

2. np.reshape: modifies/rearranges arrays

   ex. array = np.array([1,3,5,7,9,11])
   array1 = np.reshape(array,(2,3))
   Output: [[1,3,5]
   [7,9,11]]

3. np.add: Adds the contents of two arrays.

   ex. array1 = np.array([1,2,3,4,5])
   array2 = np.array([4,9,16,25,36])
   np.add(array1,array2)
   Output: [5,11,19,29,41]

4. np.mean: Computes the statistical average of the elements in the array.

   ex. array = np.array([76,78,81,66,85])
   np.mean(array)
   Output: 77.2

5. np.min: Computes the statistical minimum of the elements in an array.

   ex. np.min(array)
   Output: 66