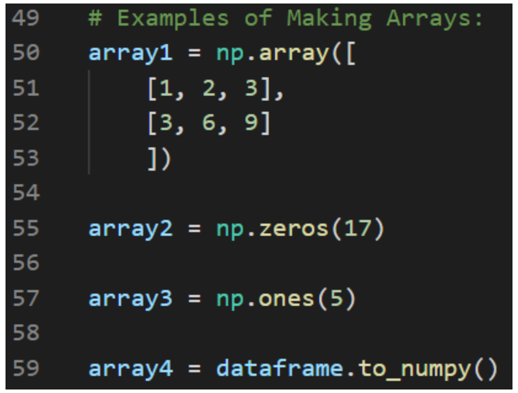
Numpy Cheat Sheet

- What are numpy arrays?
  - Store data as a grid/matrix
    - All data has to be the same data type
      - Will convert ints to floats automatically
  - Access with numpy package
    - import numpy as np
  - Rather than having elements removed or replaced like in lists, the entire array is deleted and recreated each time it is manipulated

- How to make numpy arrays:
  - See picture for multiple examples of numpy array types
    - **np.array** creates a general array. This is good for simple arrays when you want to put all the numbers in yourself.
    - **np.zeros** creates a 1-D array entirely of zeros of whatever size you put in the parentheses.
    - **np.ones** does the same as np.zeros just with ones.
    - **dataframe.to_numpy()** turns a dataframe into a numpy array
      - This can be done on regular dataframes that have been created by hand, or on dataframes that were generated by pulling data in from an outside file (this is the method we have used to pull data in for our forecast competition thus far)

- How to index and slice numpy arrays:
  - Indexing begins with 0 just like in lists
  - Slicing = indexing for numpy arrays
  - Use same format as lists [start:stop:step], have to have square brackets
    - Have to overshoot the stop because the stop value is not included in the output
    - Examples:
      - [2,4] = pulls value in third row, fifth column
      - [3:7, 3:4] = pulls values in rows 4-6, column 4
      - [:8, 6] = pulls all rows from 1st row to 9th row in 7th column
      - [: , 1] = all rows in the second column (can do it the other way for columns too)
      - array_name[5] = pulls the 6th row of data



```
# Examples of Making Arrays:
array1 = np.array([
    [1, 2, 3],
    [3, 6, 9]
    ])

array2 = np.zeros(17)

array3 = np.ones(5)

array4 = dataframe.to_numpy()
```

Important Functions:

- **np.mean** = finds the average of the array
- **np.median** = finds the median value in the array
- **np.std** = standard deviation of the array
- **np.sum** = the sum of all the elements in the array
- **np.round** = rounds the output of the input parameter to a specific number of decimals
- **np.min()** = minimum value in the array
- **np.max()** = maximum value in the array
  - For all the functions above, can choose which axis they apply to (i.e rows or columns)
    - axis = 0 : this summarizes across all rows
    - axis = 1: this summarizes across all columns
  - Or can use indexing to choose which part of the array it applies to

Key Methods:

- **.append()** = adds elements to the end of an array (creates a copy of an array)
- **.insert()** = inserts an element in a specific index spot (creates in a copy of an array)
- **.empty()** = deletes all the values in an array that has been assigned to a variable

Key Attributes:

- **.ndim** = dimensions
- **.shape** = rows and columns (if 2-D), # of elements/dimensions (if 1-D)
- **.dtype** = data types housed in the array