General Points

- Use the course material located at:
  - Python Data Science Handbook
- Assignment 09 can be completed using previously covered material and content from the following chapters:
  - 00.00-Preface *through* 02.05-Computation on Arrays Broadcasting
- After completing the requirements, test to ensure all cells run correctly in the .ipynb file.
- Include appropriate markdown cells to identify the requirements below by number. See this example.
- Produce an .html file that shows the .ipynb after a *successful test run*.
  - by File | Download as | HTML (.html) .
- Test the .html file by opening it in a browser and ensure the content is produced correctly from the run in Jupyter Notebook.
- Submit BOTH the .ipynb and .html files to the appropriate link in Blackboard | Assignments. Submit the .html file as a .zip file to pass security settings. Submit other files individually.
- Submit any additional files required to complete the assignment.

Requirements

(*Ensure that all Requirements are complete*)

1. Using Jupyter Notebook (or similar tool), create a file named:
   - assignment-09.ipynb
2. Add an H1 markup: “This is Assignment 09 - <yournamehere>”
3. Include appropriate markdown cells to identify the requirements below by number.
4. Explain how NumPy arrays are like Python lists and how they are different. See multiple chapters for this information.

5. Explain how variable typing is performed in Python compared to how it is performed in a language like C.
6. Explain how integer storage is performed in Python compared to how it is performed in a language like C.
7. Create a list (L1) of numbers 1-20. Convert the numbers in L1 to strings and store them in a new list (L2).
8. Demonstrate a heterogeneous list with at least **5** different data types.
9. Create a Python array using the built-in array module. Demonstrate the following operations on the array:
    - append
    - insert
    - index
    - attempt to insert an element of a different type
    - clear
10. Create a NumPy array of 10 floats named one_d_array. Demonstrate the following operations on the array:
    - implicit casting (the author calls this up-casting)
    - explicit casting
11. Use list comprehension to create a two-dimensional NumPy array named two_d_array with 5 rows and 3 columns with the following content:

```
[[1 2 3]
 [2 3 4]
 [3 4 5]
 [4 5 6]
 [5 6 7]]
```

12. Create a 3-dimensional NumPy array named three_d_array with random numbers between 0-99. The *shape* of the array should be (3, 5, 5). Also display the size in bytes of the array.

13. Demonstrate accessing 4 different individual elements of the three_d_array.
14. Demonstrate modifying 4 different individual elements of the three_d_array.
15. Using one_d_array, demonstrate slicing:
    - the first 7 elements
    - all after index 7
    - from index 3 to index 8
    - every other element
    - elements reversed
    - reverse every other element from index 7
16. Using two_d_array, demonstrate slicing:
    - first 2 rows and first 2 columns
    - all rows and every other column
    - all columns and every other row
    - extracting a subarray
17. Use NumPy's arrange and reshape methods to create a 10x10 array with values as shown:

```
[[  1   2   3   4   5   6   7   8   9  10]
 [ 11  12  13  14  15  16  17  18  19  20]
 [ 21  22  23  24  25  26  27  28  29  30]
 [ 31  32  33  34  35  36  37  38  39  40]
 [ 41  42  43  44  45  46  47  48  49  50]
 [ 51  52  53  54  55  56  57  58  59  60]
 [ 61  62  63  64  65  66  67  68  69  70]
 [ 71  72  73  74  75  76  77  78  79  80]
 [ 81  82  83  84  85  86  87  88  89  90]
 [ 91  92  93  94  95  96  97  98  99 100]]
```

18. Using one_d_array, create a row vector and a column vector.
19. Explain what a NumPy *universal function* is and why they are useful.

20. Demonstrate the use of the *out* argument.
21. Use aggregates to show the product of all elements of one_d_array.
22. Create the NumPy arrays x and y below and use the multiply() and outer() methods to produce the output as shown.

```
x = [1 2 3 4 5]
y = [2 2 2 2 2]

array([[ 2,  2,  2,  2,  2],
       [ 4,  4,  4,  4,  4],
       [ 6,  6,  6,  6,  6],
       [ 8,  8,  8,  8,  8],
       [10, 10, 10, 10, 10]])
```

23. Demonstrate NumPy's sum, min, and max functions. Use the %timeit magic function to compare the performance to Python's built-in sum, min, and max functions.
24. Use the file president_heights.csv to produce a histogram of U.S. President height information.
25. Use broadcasting and Matplotlib to plot produce a plot of a two-dimensional array.
26. Use markdown to include a statement at the end of assignment-09.ipynb explaining your experiences with Assignment 09. Make this authentic (minimum of 2-3 sentences).

TEST – TEST – TEST your .ipynb file to ensure all requirements are met. Produce an .html file from a *successful test run* of the .ipynb file. Ensure that the .html is produced correctly by opening it in a browser.

- Use the list above as a confirmation checklist.
- Not meeting all requirements = 0 points for the assignment.