General Points

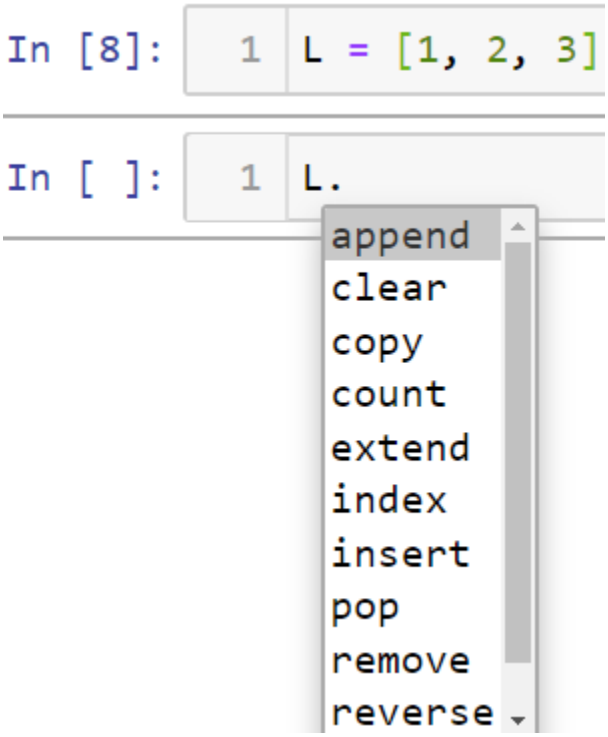
- Use the course material located at:
  - Python Data Science Handbook
- Assignment 08 can be completed using previously covered material and content from the following chapters:
  - 00.00-Preface *through* 01.08-More IPython Resources
- After completing the requirements, test to ensure all cells run 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 files individually (via a multi-select). However, if your browser posts an error for the .html file, submit it as a .zip.
- 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-08.ipynb
2. Add an H1 markup: “This is Assignment 08 - <yournamehere>”
3. Include appropriate markdown cells to identify the requirements below by number.

4. Access three sources for definitions of Data Science. Include those definitions in a markdown cell. Include appropriate citations for the sources.
5. Demonstrate accessing the help system for five different topics.
6. Write a function *cube()* that raises an argument to the third power and returns that argument. Include a *docstring* in the function. Include a call to cube().
7. Display the docstring and source code of cube() using the shortcuts shown in the online textbook.
8. Define a list L. Include L. in a code cell below L and press <TAB> to observe code completion. This displays the methods are available to Python list objects.



9. Demonstrate the use of tab completion with the import statement.
10. Demonstrate the use of wildcard matching.
11. Create a file myscript.py that includes the definition of cube() you wrote earlier and a for loop to call cube() multiple times. Use a magic command to

run myscript.py from a code cell. Include myscript.py in your Bb submission so your code works when run.

12. Use a magic command to time an operation.
13. Demonstrate silently displaying output.
14. Demonstrate five different command console shell commands.
    - Due to the differences between Windows and MacOS/Linux, the UNIX based shell commands do not operate correctly in Windows. Given this, simply use '!' with five Windows commands.
    - If your installation is still problematic, make comments about the requirements in a markdown cell explaining the commands you attempted and results.
15. With automagic set to off, demonstrate five different shell-related magic commands.
16. With automagic set to on, demonstrate five different shell-related magic commands. Different from those in the previous requirement.
17. Demonstrate the use %xmode magic function.
18. Use *Error? to show the list of Python system exceptions. Demonstrate *throwing* three of these exception types (in separate code segments).
19. Demonstrate the use of the debugger with the exceptions thrown in the previous requirement.
20. Use at least 4 magic commands to time a function(s) that you have written.
21. Use markdown to include a statement at the end of assignment-08.ipynb explaining your experiences with Assignment 08. 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.