General Points

- Use the course material located at:
  - Whirlwind Tour of Python
- Assignment 05 can be completed using previously covered material and content from the following chapters:
  - 00-Introduction *through* 05-Built-in Scalar Types
    - Note: Scalar types are also known as Simple types
- After completing 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-05.ipynb
2. Add an H1 markdown cell: "This is Assignment 05 - <yournamehere>"
3. Include appropriate markdown cells to identify the requirements below by number.
4. Demonstrate the following arithmetic operations in a cell:

- addition
- subtraction
- multiplication
- true division
- floor division
- exponentiation

5. Demonstrate the following comparison operations in a cell:
   - ==
   - !=
   - < , >
   - <= , >=
6. Use a text editor like Notepad++ (or an IDE of your choice) to create a file named operators.py with the same arithmetic and comparison operations as Requirements 4 & 5. Include appropriate print statements. In assignment-05.ipynb, include a cell running operators.py.

   Hint: Use one of these to execute operators.py in Jupyter Notebooks.

```
In [1]:  1  import operators

         This is the operators module.

In [2]:  1  !python operators.py

         This is the operators module.
```

   The second version is a little easier to work with due to the server caching of the import operation. More on both techniques as we progress through the course.

7. In a cell, use Python to:
   - Include appropriate comments in your code.
   - Create two lists, list_1 and list_2.

- Populate each list with 10 arbitrary numbers, some even and some odd in each list.
- Create two more lists, list_even and list_odd.
- Programmatically populate list_even with the even numbers from list_1 and list_2.
- Programmatically populate list_odd with the odd numbers from list_1 and list_2.
- Print list_even and list_odd.

8. Demonstrate the use of the method is_integer() to test if numbers qualify as integers.
9. Demonstrate the following Boolean, Identity, and Membership operations in a cell:
    - and
    - or
    - not
    - is
    - is not
    - in
    - not in
    - Object Identity
10. Demonstrate *variable precision* in a cell.
11. Demonstrate the following string operations in a cell:
    - len
    - upper()
    - capitalize()
    - concatenation
    - multi-concatenation
    - access of individual characters
12. Use markdown to include a statement at the end of assignment-05.ipynb explaining your experiences with Assignment 05. 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 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.