

# CSCA20 Lab 4

## 1 Objectives

- Practice using loops.
- Practice writing functions that involve loops.

## 2 Driver and navigator

As always, this lab requires you and your partner to take on distinct roles: **driver** (the person typing at the keyboard) and **navigator** (the person watching for mistakes, and thinking ahead).

The rest of these instructions call you **s1** and **s2**. Pick which one is which. **s1** should log in, start up Wing, and be the first driver.

## 3 Basic Iteration

Working in the Python shell, begin by assigning the string “superconductivity” to a variable **s**. (If your fingers are up to it, you may use the string “supercalifragilisticexpialidocious”.)

Now do these exercises (alternating who is driving and navigating):

1. Write a loop that prints each character in **s** on a separate line.
2. Write a loop that prints each character in **s** on the same line with a space after each character. Notice where the `>>>` prompt appears. Be prepared to explain why to your TA.
3. Write a loop that prints each character in **s** on the same line, with a space and a comma after each character: **s , u , p , e , r , c ,** etc.
4. Write a loop similar to the previous but without a space between each character and the comma that follows it: **s, u, p, e, r, c,** etc. Notice that there is a comma after the last character. We won’t worry about this for now.

## 4 Loops

This section asks you to write the code for the functions specified below. There are some important aspects to these exercises:

- Your code must pass the stylechecker.
- For each of these functions, you will need to complete the full docstring first, including the `doctest` examples! Run the `doctests` for each function, once you’ve implemented it.

The functions:

- `count_letter(str, str)`: Given a string and a single-character string, count all occurrences of the second string in the first and return the count.
- `remove_digits(str)`: Return a new string that is the same as the given string, but with digits removed.
- `repeat_character(str, str)`: Given a string and a single-character string, return a string consisting of the second, single character string repeated as many times as it appears in the first string.

- `every_other(str)`: Return a string built from all characters of `s` at even indices, i.e. first, third, fifth, etc. characters.

Submit your functions in a file called `lab4.py` in assignment Lab 4.