

CSC108H Week 3 Lab

To earn your lab marks, you must actively participate in the lab.

1 Objectives

1. Explore module `math` using `help` and `dir`
2. Write `if` statements to manipulate images
3. Mess around with strings
4. Practice submitting an assignment

2 Driver and Navigator

driver: The person typing at the keyboard.

navigator: The person watching for mistakes and thinking ahead.

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

3 math Module

Python provides a large number of useful functions, many of which are saved in separate *modules*. (Modules are used for organization; it's like having a tool chest where modules are drawers, instead of throwing all of your tools into a single, huge box.) In this lab, we will explore the `math` module. We have provided you with a few tasks to complete, and you should use functions from the `math` module to do so. Hint: Use `dir(math)` to find out which functions `math` contains and use `help(math.function)` (where *function* is replaced by the name of a function from `math`) to find out more about a particular function.

To begin, `import math`. In the shell, write code to do the following:

- Take the floor and ceiling of 56.4.
- Calculate the square root of the (natural) logarithm of 244.
- Calculate the base 2 logarithm of 256.
- Calculate the absolute value of -34.84.
- Calculate 2π .

4 if statements and image manipulation

In last week's lab, you wrote several functions to manipulate images. Those functions had something in common: they performed the same action on *every* pixel in the image. This week you will manipulate images, but you will not necessarily perform the same action (or any action) on each pixel. Hints:

- You will need to use `if` statements to complete this section.
- In the last two functions, use `media` functions `get_color` and `set_color` when appropriate.

Download `w3.py` from the course website, and add the following functions to it:

<code>reduce_blue(pic)</code>	Set the blue color component of each pixel in picture <code>pic</code> to zero if the current blue component of the pixel is greater than 100. Hint: read function <code>reduce_red</code> from <code>w3.py</code> .
<code>black_to_red(pic)</code>	For each pixel in <code>pic</code> , if it is currently black, set it to red.
<code>grey_posterize(pic)</code>	Set the color of each pixel in picture <code>pic</code> to black if the luminance is less than 128, and to white otherwise. Note: The luminance of a pixel is the average of its red, green, and blue color components.

`w3.py` includes a function `make_test_pic` that creates a *very small* picture for you to use during debugging. Don't try to debug a larger picture; you'll have to step through a lot of code!

Once you have completed all three functions, demonstrate them to your TA.

5 Practice submitting an assignment

The assignments in this course will be submitted electronically using the CDF submission system. Visit the following assignment submission information page:

<http://www.cdf.toronto.edu/students/submit.html>

Submit `w3.py` for assignment `LabPractice`. (This is the only time that you will need to submit your lab work electronically and you will not be marked on it; this is just for practice submitting).

Switch roles: s1 drives and s2 navigates: s1 should also practice submitting `w3.py`.

6 Type `str`

The Python type `str` stands for string. A string is a sequence of characters. We can “walk” through it in a loop and examine each character. In the Python shell, do the following (alternating who is driving and navigating):

1. Assign the string “superconductivity” to a variable `f`. (If your fingers are up to it, you may use the string “supercalifragilisticexpialidocious”.) Recall that “+” combines strings (concatenates them), and “,” can be used to separate items in a print statement.
 - (a) Write a loop which prints each character in `f` on a separate line.
 - (b) Write a loop which prints each character in `f` on the same line with a space after each character. Notice where the `>>>` prompt appears. Be prepared to explain why to your TA.
 - (c) Write a loop which prints each character in `f` on the same line, with a space and a comma after each character.
 - (d) Write a loop similar to the above, but so that it will put spaces only after the comma, and no spaces between a character and a following comma: `s, u, p, e, r, c,` etc. Notice that there is a comma at the end of the last character printed. We won't worry about this now.

7 Using `str` Methods

Methods act just like functions. The only difference is that methods are functions that belong to specific types and are designed to make use of values of those types. What methods a data value has depends on its type. Use `dir` and `help` to look up `str` methods that will help you complete the following tasks. Create strings in the Python shell and do the following (alternate driving and navigating):

1. Check whether or not a string ends with the letter “h”.
2. Create a version of a string with all letters converted to uppercase.
3. Check whether or not a string contains only numbers.

4. Check whether or not a string contains only lowercase letters.
5. Replace every instance of the character “l” in a string with a “1”.
6. Pad a string composed only of numbers with zeros so that it is exactly 10 characters long.
7. Remove all leading zeros from a string composed only of numbers.