LARSON—MATH 255–CLASSROOM WORKSHEET 05
Strings, Booleans, and More.

1. Create a Cocalc/Sage Cloud account.

   (a) Start the Chrome browser.

   (b) Go to `http://cocalc.com`

   (c) "Create new account" using **your VCU email address** .

   (d) You should see an existing Project for our class. Click on that.

   (e) Click "New", then "Worksheets", then call it **c05**.

   (f) For each problem number, label it in the Sage cell where the work is. So for Problem 2, the first line of the cell should be `#Problem 2`.

   In order to do sophisticated calculations, or to allow for multiple inputs, you will need to write *programs*. Our "hello world!" program was the first example. It included a `print` statement. Other program features, in almost any language, include *conditional statements* (if..then..) and *loops*.

2. Write your own function `triple_product` that takes *three* inputs (call them anything, or x, y, z is fine) and *returns* their product.

   String formatting.

   A *string* is a sequence of *characters* (letters, numerals, symbols, etc). If you put a sequence of characters between quotes, you are telling Sage to treat what's between the quotes as a string (instead of as a *keyword*). Strings can be manipulated, and have places that can be filled in.

3. Type and evaluate `print('This string has {}'.format('17 characters'))`. Now try replacing '17 characters' with any other string.

4. Type and evaluate the following program.

```
def superstring(x):
    print('This string has {}'.format(x))
```

5. Now test your function. Type and evaluate `superstring('black letters')`.

   More graphing and calculating basics.

6. Make a point at $(4, 4)$ Evaluate `point((4,4))`.

7. Make it bigger by adjusting the "size" parameter; evaluate `point((4,4),size=200)`. Try other values for `size`.

8. Draw a line from $(-1, 1)$ to $(4, 4)$ by evaluating `line([(-1,1),(4,4)])`. Try drawing a line with 3 points.

9. Make the line thicker by adjusting the "thickness" parameter: evaluate `line([(-1,1),(4,4)],thickness = 4)`. Try other values of `thickness`.

10. Make the line dashed by adjusting the "linestyle" parameter: evaluate `line([(-1,1),(4,4)],linestyle="dotted")`. Try another value for "linestyle" by reading the options from the help command `line2d?`.

11. Now make the line red.

12. Draw a triangle between $(1,1)$, $(1,2)$, and $(2,1)$ using the line command.

13. Now draw a triangle between $(1,1)$, $(1,2)$, and $(2,1)$ using the `polygon` command; find examples of how this command works with `help(polygon)`. What's the difference?

Boolean Expressions in Sage

A *boolean expression* is one that evaluates to True or False.

14. Evaluate `3==4`.

15. Evaluate `3==3`.

16. Evaluate `3>3`.

17. Evaluate `3>=-3`.

18. Evaluate `13%2==1`.

19. Evaluate `13%2==0`.

While "==" is used as a claim of equality of expressions (the left-hand-side and the right-hand-sides of the "==") the symbol "!=" is used to express in-equality.

20. Evaluate `5!=7`.

21. Evaluate `5!=5`.

22. We will *assign* a value to a variable "a". Then we will use that variable in a boolean expression. (These two lines can be typed in one cell, or each in its own cell). Type and evaluate:

```
a=5
a>2
```

Boolean expressions can be combined with *boolean operators* like "and" and "or".

23. Evaluate: `3==3 and 3==4`.

24. Evaluate: `3==3 or 3==4`.

Getting your classwork recorded

When you are done, before you leave class...

(a) Click the "Make pdf" (Adobe symbol) icon and make a pdf of this worksheet. (If Cocalc hangs, click the printer icon, then "Open", then print or make a pdf using your browser).

(b) Send me an email with an informative header like "Math 255 - c05 worksheet attached" (so that it will be properly recorded).

(c) Remove the worksheet from the [illegible]