

Last name _____

First name _____

LARSON—MATH 356—LAB WORKSHEET 02
Graphs!

1. Log in to your Sage/CoCalc account.
 - (a) Start the Chrome browser.
 - (b) Go to `http://cocalc.com` and sign in.
 - (c) You should see an existing Project for our class. Click on that.
 - (d) Click “New”, call it **s02**, then click “Sage Worksheet”.
 - (e) For each problem number, label it in the Sage cell where the work is. So for Problem 1, the first line of the cell should be `#Problem 1`.
 - (f) When you are finished with the worksheet, click “make pdf”, email me the pdf (at `clarson@vcu.edu`, with a header that says **Math 356 s02 worksheet attached**).

Sage uses Python as its *interface language*: that is, the basic syntax and program control and organization are Python. And you can run pure Python in Sage Worksheets cells). This also means that the more Python you learn, the more power you will have to write interesting and useful code).

In order to do sophisticated calculations, or to allow for multiple inputs, you will need to define *procedures* (also called *functions*). 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. Type in the following procedure definition and evaluate.

```
# This function returns the absolute value of a number x
def absolute(x):
    if x>=0:
        return x
    else:
        return -x
```

3. Now test it. Evaluate `absolute(4)`, `absolute(-4)`. “#” is the *comment* symbol. Everything after “#” is ignored—and not evaluated.

```
def abs_plus_five(x):
    return absolute(x)+5
```

4. You don’t have to add five, you can add *any* number by adding a *parameter*.

```
def abs_plus(x,y):
    return absolute(x)+y
```

5. Now test it. Evaluate `abs_plus(4,5)`, `abs_plus(-4,5)`, `abs_plus(-4,23)`, etc.

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

6. Evaluate `3==4`.
7. Evaluate `3==3`.
8. Evaluate `3>3`.
9. Evaluate `3>=-3`.
10. A very useful arithmetic operator in Sage is the *modulo* operator (represented by `%`). `a%n` gives the remainder of dividing a by n . Evaluate `5%2`. Now evaluate `6%2`. Try `99%5`.
11. Evaluate `13%2==1`.
12. 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 does-not-equal.

13. Evaluate `5!=7`.
14. Evaluate `5!=5`.
15. 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”.

16. Evaluate `3==3 and 3==4`.
17. Evaluate `3==3 or 3==4`.

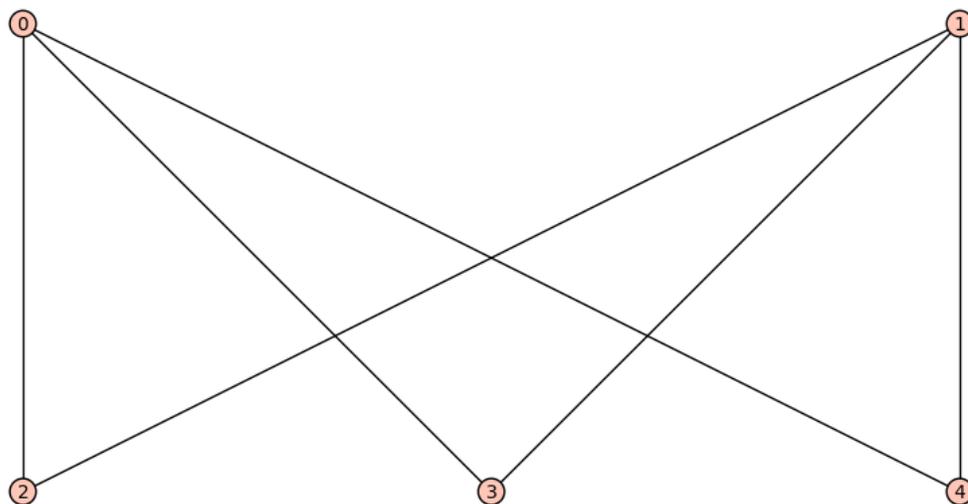
Lists in Sage

A *list* is a basic *data structure* in Python and Sage. They are represented by square brackets with comma separated numbers, strings, etc., between them (like `[2, 5, 9]` or `["red", "blue"]`).

18. Lists can be given names. Evaluate `L=[2,5,9]`. Then evaluate `L`.
19. Lists are indexed starting with 0. Evaluate each of `L[0]`, `L[1]`, `L[2]`, and `L[3]`.
20. Lists can be combined with "+". Evaluate `[2,5,9]+[3,4,5]`.
21. Let `M=[3,4,5]`. Evaluate `L+M`.
22. If you want all the integers from x to y you can use the shorthand notation `[x..y]`. Evaluate `[3..7]`.
23. You can have a list of lists. Evaluate `L=[[0,1],[2,3],[4,5]]`. Now evaluate `L[1]`. Then evaluate `L[1][0]`. What do you think the value of `L[0][1]` is?
24. You can use *list comprehension* to apply a function to all the elements of a list. Evaluate `[abs(x) for x in [-1,2,-3]]`.

Getting Graphs into Sage

25. How can we get the following graph into Sage/Cocalc?



26. Use Sage to find an incidence matrix for this graph.

27. Use Sage to find an adjacency matrix for this graph.

28. Many many graphs are already pre-coded (built in) to Sage/Cocalc. How can we access the built-in Petersen Graph?

29. Use Sage to find an incidence matrix for this graph.

30. Use Sage to find an adjacency matrix for this graph.