

CS 1110, LAB 1: EXPRESSIONS AND ASSIGNMENTS
<http://www.cs.cornell.edu/courses/cs1110/2014sp/labs/lab01.pdf>

First Name: _____ **Last Name:** _____ **NetID:** _____

This lab serves two purposes. First, it is designed to get you started with Python immediately, particularly with the command shell. Second, it gives you hands-on experience with Python expressions, which we talked on the first day of class. Learning a computer language is a lot like learning a new human language, and this lab essentially works as a *grammar drill*.

1. GETTING STARTED WITH PYTHON

If your primary computer is a laptop, bring it to the lab to work on, as lab is an *excellent* opportunity to get started with Python on your machine. You should follow the instructions on the course website:

<http://www.cs.cornell.edu/courses/cs1110/2014sp/materials/python.php>

Ask a consultant for help if you have problems with your installation; that is why they are there.

Next, you need to start up the command shell on the computer you are working on. This would be the Command Prompt on Windows, or the Terminal on OS X. If you are unsure of what these mean, look at the tutorial on the course website and read *just* the first one or two paragraphs for your operating system

<http://www.cs.cornell.edu/courses/cs1110/2014sp/materials/command.php>

You do not need to read the sections on navigating directories; we will use that in a later lab.

After you have started the command shell, type `python` at the prompt. On an ACCEL-lab computer, you should see output that looks something like this:

```
ActivePython 2.7.2.5 (ActiveState Software Inc.) based on
Python 2.7.2 (default, Jun 24 2011, 12:22:14)
[MSC v. 1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

To make Python evaluate an expression, type the expression in after the `>>>`. When you hit “return”, Python will give you the evaluation, and then show with another `>>>` prompt.

1.1. Useful Shortcuts. If you press the *up-arrow key*, you obtain the previous expression that you typed in. Pressing up-arrow repeatedly scrolls through all the expressions that you have typed earlier in the session; if you go too far back, you can press the *down-arrow key* to get to a later expression. The *left and right-arrow keys* move the text cursor on a single line. Using all of these keys together, you can take a previously-used expression, modify it, and try it again.

2. LAB INSTRUCTIONS

The following pages have a list of expressions. For each expression, first *compute the expression in your head, without Python*. Write down what you think the value is in the second column of the table. If you have no idea, write “?”.

Next, use Python to compute the same expression. You may find it easier to cut-and-paste from the online version of these instructions; a clickable link is just under the title of this document. Write down Python’s result in the third column. **You should always fill in the second and third column of a row before moving on to the next row.** You want to learn from earlier examples before moving on to the next one.

If the two values are different, you should try to figure out why Python gave the answer that it did. Come up with a reasonable explanation and put it in the final column. You are not really being graded on complete correctness in these labs (see below) so make your best guess at what is happening; your answer will help the staff understand how to better aid you.

This lab is just supposed to be practice; do not waste too much time trying to figure things out yourself. If you do not understand something, ask a staff member immediately. It is very important that you understand *how* each expression is evaluated, so if an answer doesn’t make sense, ask someone. The lab instructors and consultants are in the lab to help. They will look over your shoulder from time to time and give you advice.

2.1. Getting Credit for the Lab. Labs are graded on effort, not correctness. When you are finished, you should show your written answers to this lab to your lab instructor. The instructor will ask you a few questions to make sure you understand the material and then swipe your ID card to record your success. This physical piece of paper is yours to keep.

If you do not finish during the lab, you have **until the beginning of lab next week to finish it**. You should show it to your lab instructor at the *beginning* of that next lab.

3. LAB PROBLEMS

3.1. int and float Expressions.

Expression	Expected Value	Calculated Value	Reason for Calculated Value
$2 * 3$			
$2 ** 3$			
$5 + 2 * 5$			
$(5 + 2) * 5$			
$-4 - -4 - -4$			
$2 ** 2 ** 0$			
$(2 ** 2) ** 0$			
$6 / 2$			
$6 / 4$			
$6.0 / 4$			
$6 / 4.0$			
$9.0 * 0.5$			
$9.0 ** 0.5$			
$6 \% 2$			
$7 \% 2$			
$6.2 \% 4$			

3.2. Types and Casting.

Expression	Expected Value	Calculated Value	Reason for Calculated Value
float(4)			
int(4)			
int(5.3)			
float(int(5.3))			
int(-5.3)			
int(-5.7)			
float(7) / 4			
7 / float(4)			
float(7 / 4)			
type(4)			
type(7 / 4.0)			

3.3. Comparisons and boolean Expressions.

Expression	Expected Value	Calculated Value	Reason for Calculated Value
<code>3 < 5</code>			
<code>3 < 5 and 5 < 3</code>			
<code>True</code>			
<code>true</code>			
<code>True and False</code>			
<code>True and True</code>			
<code>True or False</code>			
<code>False or False</code>			
<code>not True</code>			
<code>not not False</code>			
<code>not False and True</code>			
<code>not (False or True)</code>			
<code>True and False and True</code>			
<code>True or (False and True)</code>			
<code>(5 / 0 == 1) and False</code>			
<code>False and (5 / 0 == 1)</code>			

Why does the last expression in the table above “work” but the one above it doesn’t?

3.4. String Expressions.

Pay close attention to spaces and to the different types of quotation marks being used; we use both ' (single quote) and " (double quote). Also, note the difference between " (a double quote) and "(two consecutive single quotes).

Expression	Expected Value	Calculated Value	Reason for Value
'Truth ' + 'is ' + 'best'			
'Truth' + 'is' + 'best'			
"Truth " + "is " + "best"			
"Truth " + ('is ' + "best")			
'A double quote: "'			
"A single quote: '"			
'A single quote: "'			
" + 'ok'			
" + '4 / 2'			
" + 4 / 2			
" + str(4 / 2)			

3.5. Variables and Assignment Statements.

The last part of this lab involves assignment statements. You need to know the difference between expressions, which you've been working with so far, and assignment statements. An assignment statement like

$$b = 3 < 5$$

is a command to do something. In particular, this command

- (1) evaluates the expression on the right-hand side of the = (in this case, $3 < 5$), and
- (2) stores its value in the variable on the left-hand side of the =, in this case, b .

Because it is not an expression, Python will not actually output a result when you type it in; it will just perform the command silently.

In the table below, the first column contains *either* an expression or a command. If it is an expression, write the value. If it is a command, you should just write "None" (we have done the first one for you). Because some of the entries are commands, it is important that you *enter the expressions or commands in exactly the order they are given*.

Statement or Expression	Expected Value	Calculated Value	Reason for Calculated Value
$i = 2$	None		
i			
j			
$j = 1$			
j			
$j = j + i$			
j			
i			
$w = \text{'Hello'}$			
$i + w$			