

Python Lab 1: Basic Python

In this lab we practice all that we have learned on variables (lack of types), naming conventions, numeric types and coercion, strings, Booleans, and operator grouping, as well as writing our first program in a file.

If you get confused, look at these resources available to you:

1. Lecture slides
2. Other students
3. Python documentation
4. Google
5. Instructors

Exercise 0: Survey

If you haven't filled out the course survey, please do so now:

<http://www.surveymonkey.com/s/3BPMKHF>

Exercise 1: Python Variables

Write your answers on paper or in a document on your computer.

1. Variable Names

Which of these variable names are valid? (If you aren't sure, you can test from the Python shell. Start the shell by typing `python` in the Terminal)

1. `x`
2. `56thnumber`
3. `length`
4. `!Tayo!`
5. `NUMBER`
6. `Android_phone1`
7. `this variable`

2. Types

Write down the data type for each of the following variables. Pay special attention to punctuation: values are not always the type they seem!

1. `a = False`
2. `b = 3.7`
3. `c = 'Alex'`
4. `d = 7`
5. `e = 'True'`
6. `g = '17`
7. `i = '3.14159'`
8. `j = "---add---`

Check your answers with the Python shell

Exercise 2: Boolean operators

1. Boolean operators can seem tricky at first, and it takes practice to evaluate them correctly. Write the value (True or False) produced by each expression below, using the assigned values of the variables a, b, and c. Remember to work from the inside out, starting with the innermost expressions.

a = False

b = True

c = False

1. b and c
2. b or c
3. not a and b
4. (a and b) or not c
5. not ((not b or not a) and c) or a

2. What do the following pieces of code return?

x = 3

1. 1 < x
2. x >= 4
3. x == 3
4. x == 3.0
5. "Hello" == "Hello"
6. 4.0/2.0 == 2
7. 3 < 7 and 4 > 5

Exercise 3: Data Types in Arithmetic

What are the outputs of each of the following equations?

1. 5/2
2. 5/2.0
3. 5.0/2
4. 7*(1/2)
5. 7* (1/2.0)
6. 5 **2
7. 5.0 **2
8. 5 **2.0
9. 1/3
10. 1/3.0

Exercise 4: Using Terminal + Hello World program

For this exercise, you will write a python program in a text editor and run it from the Terminal.

First open a text editor (gedit) by clicking this icon:



Type the lines:

```
print "Hello World"  
print "I'm a Python program"  
Save the file as helloworld.py in the folder "Documents".
```

Now open the Terminal by clicking this icon:



1. In the terminal, type `cd ~/Documents`
This will move you into the Documents folder. `cd` stands for "change directory."
2. Type `pwd`
This will print the directory you are in. `pwd` stands for "print working directory."
3. Type `ls`
This will list the files and subdirectories in your current directory. `ls` stands for "list." You should see your file `helloworld.py`
4. Type `python helloworld.py`
This will run your python program. You should see the output:
Hello World
I'm a Python program

You did it! Feel free to add a few more commands to your program and run it again.

Exercise 5: User input

You can request input from the user in the following way:

To get a string from the user: `var = raw_input("prompt here")`

To get an integer from the user: `var = int(raw_input("prompt here"))`

In this exercise, we will ask the user for his or her first and last name and date of birth and print them out formatted. Create a new file and call it `birthdays.py` and save it in Documents. Here is an example of what the program would do (the text in italics is what the user inputs):

```
Enter your first name: Anne  
Enter your last name: Williams  
Enter your date of birth:
```

Month? *July*

Day? *31*

Year? *1980*

Anne Williams was born on July 31, 1980

Write the program and run it with the command `python birthdays.py`

Exercise 6: Zeller's Algorithm - Extra Credit

Note: Each lab will have extra exercises to do if you finish early. You are also welcome to help other students who are having trouble. If you don't finish the extra credit, you are welcome to work on it at home or on a later date.

Zeller's algorithm computes the day of the week on which a given date will fall (or fell). In this exercise, you will write a program to run Zeller's algorithm on a specific date. Create a new file for this program, `zellers.py`. The program should use the algorithm outlined below to compute the day of the week on which the user's birthday fell in the year you were born and print the result to the screen.

Start with the program in Exercise 2.6, but **ask for the month as a number between 1-12 where March is 1 and February is 12**. If born in Jan or Feb, enter previous year (see the notes below). Review how to get user input of different types. In the end, print out the name of the user and the day of the week they were born.

Zeller's algorithm is defined as follows:

Let A, B, C, and D denote integer variables that have the following values:

A = the month of the year, with March having the value 1, April the value 2, ...
December the value 10, and January and February being counted as months 11 and 12 of the preceding year (in which case, subtract 1 from C)
B = the day of the month (1, 2, 3, ..., 30, 31)
C = the year of the century (e.g. C = 89 for the year 1989)
D = the century (e.g. D = 19 for the year 1989)

Note: if the month is January or February, then the preceding year is used for computation. This is because there was a period in history when March 1st, not January 1st, was the beginning of the year.

Let W, X, Y, Z, R also denote integer variables. Compute their values in the following order using integer arithmetic:

$$W = (13 * A - 1) / 5$$

$$X = C / 4$$

$$Y = D / 4$$

$$Z = W + X + Y + B + C - 2 * D$$

R = the remainder when Z is divided by 7

The value of R is the day of the week, where 0 represents Sunday, 1 is Monday, ... , 6 is Saturday. If the computed value of R is a negative number, add 7 to get a non-negative number between 0 and 6.

Print out R. You can check to be sure your code is working by looking at <http://www.timeanddate.com/calendar/>.

Run some test cases- try today's date, your birth date, any other dates you like.