

CS177 Fall 2013

Lab02: Python types and simple algorithms

Outcomes

- practicing Python types
- python variables and assignment statement
- getting input from the user (`input` function)
- use the `print` function
- use a library function

Prerequisites

- textbook (e-book) Chapter 1,2, 3.1
- Lesson 1 slides
- Recitation 1 slides
- Prelab2

Introduction

In today's lab you will practice some of the Python basic types. You will practice how to get an input from the user and how to the print the results of your program to the screen using the `print` function.

You will also practice the process of the software development. Using one example of the textbook, you will be guided by your lab TA through the phases of the Software development process and you will implement the described algorithm.

In so doing, you will be introduced to the Python library concept and to the use of a function of a library.

Set-up

Before starting, you must create a folder **lab02** under the folder **CS177** you created in lab01.

If you already created the folder lab02, then proceed to the Exercises.

If you did not have created the folder CS177 yet, ask your lab TA to help you to do that, and then create the folder lab02.

Exercise 1

For each of the following statements, find the ones that generate an error and explain why.

Run these statements in your IDLE Interactive Window. Write the answer in a new file named **ex1.txt**. Also copy in the file ex1.txt the error message you will get.

a)

```
value = 4 + 13.5
```

```
value = 23.5 + "21.2"
```

b)

```
my var = 100
```

```
my_var = 100
```

c)

```
mycar = 'Ferrari Daytona'
```

```
print (myCar)
```

```
print (mycar)
```

Exercise 2 - strings

You are given the following 4 strings of one character each:

```
'H'
```

```
'e'
```

```
'l'
```

```
'o'
```

Write a program that prints the strings:

'Hello'

'olleH'

'HelloHello'

'olleHolleH'

NOTES:

1. you must assign the characters to four different variables (you can call them c1, c2, c3, c4, but you can use other names too), and then build and print the strings above.
2. Save your program in a file named **ex2.py** under the folder lab02.

Exercise 3

The objective of this exercise is to go through the phases of the software development, using the example of the textbook, and finally to write the Python program that implements the algorithm.

The problem we want to solve is to determine how long a ball would fall from a given height before hitting the ground.

Let us go through the software development stages you saw in the lesson.

PHASE 1) Understanding the problem

The starting point is the problem you want to solve, i.e. know how long a ball would fall from a given height before hitting the ground.

You need to understand the equation that governs free-fall and note that it assumes neglecting air resistance. Then, the equation is given by the well-known formula:

$$d = \frac{1}{2}gt^2$$

where g is the gravitational constant (9.81 m/sec^2). Hence, once you know d , it is straightforward to calculate t .

PHASE 2) Design

Decompose the initial problem in a sequence of tasks (or steps) to be performed.

Let's identify these steps that we must perform to solve our problem:

1. Get from the user the height d of the fall;
2. Determine the time t of the fall; and
3. print t .

PHASE 3) Algorithm

Think about how to accomplish each of the tasks identified in the PHASE2. Basically we have to define a conceptual program written in English:

1. Request the user to input the height d
2. Compute $2 * d / 9.81$, and assign this value to a variable (you may use whichever name you want for this variable, but let's call it `time_square`)
3. Then compute the square root of `timesquare`, and assign its value to another variable. You may use whichever name you want, but let's call it `effective_time`)
4. Print `effective_time`

Note that the step 2 of the Design phase was decomposed in two different steps of the algorithm:

2. Compute $2 * d / 9.81$, and assign it to `timesquare`
3. Compute the square root `timesquare` and assign it to `effective_time`

PHASE 4) Implementation

Now we have to write the Python program that implements the algorithm.

In the prelab2 you saw how to tell Python to get an input value from the user and how to assign it to a variable.

You also saw the Python `print` function.

What's missing? Well, you need to calculate a square root. How to do that? Should you write by yourself the algorithm to calculate the square root? Not really. Python (and all programming languages) provide to you several functions. They are already implemented, and you can use (re-use actually) them! These functions are grouped in so-called *libraries*.

Specifically, there is a Python library, ***math***, that contains a number of useful mathematical functions, including `sqrt`.

But you have to tell Python that your program needs to use one (or more) functions of this library.

You tell that to Python using the `import` statement:

```
import math
```

This statement tells Python to make available to your program all the functions provided by the `math` library.

Now you have all the ingredients to write your program.

NOTES:

Save your program in a file named **ex3.py** under the lab02 folder.

Grading

This lab is worth 100 points.

Exercise	%	
Exercise 1	30%	
Exercise 2	30%	
Exercise 3	40%	

Turn-in instructions

You must turn-in the three files: ex1.txt, ex2.py and ex3.py you created under the folder lab02.

From the UNIX terminal window, login into your UNIX CS account.

Then launch the following command:

```
$ cd
```

```
$ cd CS177
```

```
$ turnin -v -c cs177=COMMON -p lab02 lab02
```