

# Lab 4

## Computer Science 111, Sections 3 and 4 Fall 2013

In this lab, you'll practice writing and using functions.

### 1 Using the interpreter

You can use the interpreter to write functions and test how they work. Try this, for example:

```
>>>def double (x) :
>>>     x = x * 2
>>>     return x
>>>                                     #blank line to finish the function
>>>a = 5
>>>double (a)
>>>a
>>>a = double ( double (a) )
>>>a
```

Feel free to use the interpreter to check your syntax and to try out little sections of code to see how they work. But since interpreter is a little bit clunky to use once our programs start getting bigger, we will move straight to downloading and running files in the File window from now on.

### 2 Writing and Using Functions

#### 2.1 Moving Code Into Functions

First, make copy of **lab3.py** – the one you worked on last week – which should be stored on your U drive. Call the copy **lab4from3.py**. Open the lab file in IDLE.

Your job last week was to practice writing if statements. Your first task this week is to re-write those if statements inside functions.

**Helpful Tip:** To move code inside a function you have to cut-and-paste it and also indent it. If you look under the **Format** menu, you will see a way to indent (and outdent) a highlighted block of code. There is also a command to comment-out a block of code so you can ignore it while working on other sections.

In Part 1 of last week's lab you had to write an if statement that prints different messages about election years. Change your code as follows (you can ignore my code with the three-way-if statement).

1. Write a function called **chooseResponse** that takes one argument representing the year. Move the if statement to be inside the function: based on the year, it selects one of the string messages to be **returned** by the function. There should be no printing inside this function.
2. Put your function definition at the **top of the program**, just after my introductory comments and above the dashed line.
3. You will have to change the “outside” part of the program to call this function. Make the outside code look like this.

```
year = int (input ("What year were you born? ")) # (already in the code)
response = chooseResponse(year)
print (response)
```

```
year = int (input ("What year is it now? ")) # again
response = chooseResponse (year)
print (response)
```

4. See how handy functions are? You did the computation twice but you didn't have to write the if statement twice!

Part 2, we pretty much did in class. **Changing Part 3 is optional**; try it if you want more practice before going on.

1. Write a function called **makeChange** that takes 3 arguments: an amount in cents, a coin denomination (like 10, 25), and a coin name like "dimes" or "quarters". The function should do three things:
  - (a) Figure out how many times the denomination value goes into the amount (for example, if amount is 87 and denomination is 25 (and the string is "quarters") the answer is 3.
  - (b) Print that number and the coin name, for example: 3 quarters
  - (c) Return, as its return value, the remainder from dividing the amount by the denomination: for example,  $87 - 2*25$  is 12 (but use the modulus operator as before).
2. Your outside code that calls this function should look something like this:

```
cents = int (input ("Enter an amount in cents ... and 100: "))
print ("Here is your change: " )
```

```
cents = makeChange ( cents, 25, "quarters")
cents = makeChange ( cents, 10, "dimes")
cents = makeChange ( cents, 5, "nickels")
cents = makeChange ( cents, 1, "pennies")
```

## 2.2 Fun with turtle functions

Open up your favorite browser and get a copy of `www.cs.amherst.edu/~ccm/cs111/lab4turtle.py`. Don't forget to save it on your U drive. Open the File in IDLE, and click **Run Module**.

Whoa! Turtle graphics! You can draw pictures! Take a look at the code; here are some comments.

1. Note that there is an **import turtle** statement at the top, needed to get this to work.
2. I wrote a **drawBox** function that takes four arguments: the (x,y) coordinates of the lower left corner of the box, the size of the box (in units of pixels), and the color of the pen.
3. The body of the *function* contains commands to the turtle. I think these should be fairly self-explanatory, but here are some explanations anyway:

The turtle lives on a square canvas that is (I think) about 700 x 700 pixels in dimension. A pixel is the smallest drawing unit on the screen. The center of the canvas is at (0,0): positive x direction is to the right, and positive y direction is up. The **setheading** and **left** commands refer to rotational angles in degrees. Heading 0 is due east; turning left 90 degrees points the turtle north.

4. The body of the *program* contains four calls to **drawBox**, to draw the four squares you seen on the canvas. You can
5. There is an assignment in the file that asks you to write a new function and try it out by calling it multiple times. Feel free to change the assignment according to one of the optional ideas.
6. Visit [docs.python.org/3.3/library/turtle.html](https://docs.python.org/3.3/library/turtle.html) to see a complete list of turtle functions and capabilities. Do not worry if you can't understand everything in this document – you are still a newbie. If you are not sure, it never hurts to ask. But don't be afraid to try things out – what is the worst that could happen?

## 2.3 Your assignment

For this week's lab, your assignment is to write an entire program from scratch, so there are no files to copy. Just go to the **File** window and select **New**. Call your program **lab4.py**. **You just have to turn in one** of the functions below; you can choose which one to work on.

1. Write a function called **maxDays (m, isleap)** that takes two arguments. The first is an integer in range **1..12** representing a month, and the second is a boolean value that is True if it is a leap year and False otherwise. The function figures out how many maximum days there are in the given month, and returns that value.

You remember the rule: Thirty days hath September, April, June, and November ....

In the main part of the program, write code to prompt the user for a month and ask the user whether it is a leap year or not. The program should call the function and print out the max number of days.

2. Write a function called **isLeapYear** (**year**, **isGregorian**) that takes two arguments. The first is an integer representing a year (negative values allowed) and the second is a boolean value that is True if the year is in the Gregorian calendar, and False if it is in the Julian calendar. The function returns True or False according to whether it is a leap year or a common year.

You remember the rule:

- In the Julian calendar, it is a leap year if divisible by 4, and otherwise it is not a leap year.
- In the Gregorian calendar, it is a leap year if divisible by 4, except years that are divisible by 100 are not leap years, except years divisible by 400 are leap years. Or, put another way, if not divisible by 4, or divisible by 4 and 100 but not by 400, it is a common year.

In the main part of the program, write code to prompt the user to enter a year and to say whether which calendar is being used. Call the function, and print the result returned by the function.

When you are finished with your assignment go to [www.cs.amherst.edu/Submit](http://www.cs.amherst.edu/Submit) to submit your lab.