

# CS 8 Lab: pa04

November 5, 2014

## 1 Overview

This week we will explore Python's built-in random and turtle libraries. This pdf has suggested steps for completing the assignment.

## 2 Getting Set Up

Pull up the assignment description from <http://cs.ucsb.edu/~koc/cs8/hwexpa/pa04.html> and read through it.

Open up a new python file called `pa04.py` (use past handouts for help if needed). As the first three lines, type in comments containing your full name, lab section time, UCSB UMail email address, and perm number:

```
# Your name, Lab time
# Your UCSB email address
# Your perm number
```

For each source file in this programming assignment, you have to write two functions. Make sure you define them exactly as described on the assignment description. You can start with this:

```
import turtle
import random

#this prepares us for the random walk
def RandomWalkStart(myturtle,x,y):
    #clears the window
    #places the turtle object named myturtle in postion (x,y)

#this moves the turtle one step in a random direction
#when you make your images, you'll have to pick a value for step
#when you make your images, you keep calling this function so your turtle keeps moving
def RandomWalkStep(myturtle, step):
    angle = 45*random.randint(0,7)
    myturtle.right(angle)
    myturtle.forward(step)
```

Write the code needed in `RandomWalkStart` that will clear the window and place the turtle object in the correct position. When the turtle moves to the new position, it should not draw a line, but it should be ready to draw at the end of the `RandomWalkStart` function. Save your file.

Make a new file called `RandomWalk1.py`. In this file, we will experiment with our `RandomWalkStep` function. In order to use the `RandomWalkStep` function in `pa04`, we first have to import the `pa04` file. Do the following in `RandomWalk1.py`:

- `import pa04`
- import any relevant built-in python libraries
- create a new turtle (named whatever you want)
- move your turtle to some coordinate other than  $(0,0)$  by calling the `RandomWalkStart` function you wrote in `pa04`
  - for example, if your turtle is named `alice` and you want to move it to location  $(50,50)$ , type `pa04.RandomWalkStart(alice, 50, 50)`
- use a for loop to have your turtle move randomly by calling the `RandomWalkStep` function in `pa04` multiple times using the same step size

Experiment with the numbers you use for the starting coordinate, the step size you use in your `RandomWalk` (that's passed in to `RandomWalkStep`), and the number of times you call `RandomWalkStep`. Once you've done this for a while, and you find some numbers that will make an interesting image, add the following to the bottom of your file:

```
ts = alice.getscreen()
ts.getcanvas().postscript(file="RandomWalk1.eps")
```

Be sure to change `alice` to the name of your turtle. Now when you run the file, it will save the output turtle image to a file called `RandomWalk1.eps`. This image file will be saved in the same folder that `RandomWalk1.py` is currently saved in.

### 3 Write the other files and make more pictures!

Now you'll do a similar thing in `RandomWalk2.py` and `RandomWalk4.py`. In both of these files, import `pa04` and use the `RandomWalkStart` and `RandomWalkStep` functions from `pa04`. For both `RandomWalk2.py` and `RandomWalk4.py`, we want you to save an image outputting one of the random walks. Thus, at the very end you will have **seven files: `pa04`, `RandomWalk1.py`, `RandomWalk2.py`, `RandomWalk4.py`, `RandomWalk1.eps`, `RandomWalk2.eps` and `RandomWalk4.eps`**. Here are some tips to help you with these two python files:

- In the first quadrant,  $x$  and  $y$  are both positive.
- In the second quadrant,  $x$  is negative and  $y$  is positive.
- In the third quadrant,  $x$  and  $y$  are both negative.
- in the fourth quadrant,  $x$  is positive and  $y$  is negative.
- When you have more than one turtle, they should take turns drawing rather than having one turtle do their own drawing before another starts theirs.
- `turtle.window_width()` returns the width of the window in pixels.
- `turtle.window_height()` returns the height of the window in pixels.
- When you have more than one turtle, you only need call `getscreen` on one of them when you're writing the code to create the image eps file.

## 4 Turnin

Make you have all seven files to turn in, including three eps files of turtles making random walks.

Ready to submit? Make sure you move all your files over to CSIL first. Then, in a Terminal, navigate the the directory containing your files (they all have to be in the same directory). To turn in, type the following command:

```
turnin pa04@cs8 pa04.py RandomWalk1.py RandomWalk2.py  
RandomWalk4.py RandomWalk1.eps RandomWalk2.eps RandomWalk4.eps
```

and follow the on-screen directions. **Please turn in all seven files at the same time.** Remember, I will grade the last submission turned in before the deadline if you turn in multiple versions. **The deadline for this project is Friday, November 7th, 2014 at 11pm. We will not be accepting late submissions like we did last week, so make sure you give yourself enough time to complete and turn in your project.**