

# Purpose

The main purpose of this lab is to give you some experience navigating the Linux operating system and command-line tools, the Python programming environment, and to write your very first Python program. Even if you have experience with some of these things already, you should make sure you complete this lab, since from now on, we're going to assume that you know all of this background stuff.

# Preparation

Find Dearborn 119. Log into a computer with your engineering account.

# Assignment

1. Work your way through some of [these tutorials \(Links to an external site.\)](#). In particular, you're going to want to be familiar with the [introduction \(Links to an external site.\)](#), and tutorials [one \(Links to an external site.\)](#), [two \(Links to an external site.\)](#), [four \(Links to an external site.\)](#), [five \(Links to an external site.\)](#), and [eight \(Links to an external site.\)](#). You can take a look at the other ones, too, if you're interested (although seven is not going to be relevant to this class at all). Keep in mind that these tutorials were written for a slightly different system, so the details of how to open terminals and things like that are going to be a bit different in the Dearborn lab. If you can't figure it out for yourselves, then grab a TA and ask.
2. Now that you've seen a bit of linux, create a directory for this class, called `me499` or `me599`. Create a subdirectory in here called `lab0`, and go into this directory. All of the code you write for this lab should go in here.
3. Start up `idle` the Python development environment by typing `idle27` on the command line. This will bring up a Python interpreter where you can type things. Try typing some things in it. Once you've convinced yourself that it works as expected, it's time to write your first Python program. Click on "File", then "New Window" to open an editing window. In this window, type in the Hello, World! program

```
#!/usr/bin/env python
```

```
print 'Hello, World!'
```

Once you've typed it in, hit the F5 key to execute the file. The first time you do this, you will be prompted to save the program to a file. Call the file `hello.py`. The `.py` is a convention that indicates that the file is a Python program. It's not required, but you should still do it, since it will make your life easier in the long run.

4. Take a look at your directory with the `ls -l` command. You should see that `hello.py` does not have execute permissions set, meaning you can't run it as a stand-alone program. You can, however, run it with the interpreter. Do this

```
python hello.py
```

Then, change the permissions to allow you to run it directly.

```
chmod u+x hello.py
```

```
./hello.py
```

5. Write some python code that calculates the volume of a cylinder of height 5 and radius 3. Put this in a file called `cylinder.py` that prints out the answer when you run it, like this:

```
./cylinder.py
```

```
342.3423
```

(342.3423 is not the right answer, by the way).

6. Write some python code that calculates the volume of a torus with an inner radius (the radius of the hole) of 3 and an outer radius (the radius from the center of the shape to the outer edge of the shape) of 4, and prints it out. Put this code in a file called `torus.py`.

## Grading

You will be graded for showing the TA the following things.

1. Running your Hello, World! code from `idle`, using the interpreter, and directly as a stand-alone program. 1 point each for running the code. [2 points total]
2. Running the code for the cylinder volume calculation. 1 point for code that runs. 1 point for getting the correct answer. [2 points total]
3. Running the code for the torus volume calculation. 1 point for code that runs and prints the correct answer.

## What to Hand In

Submit a tarball of your code to Canvas. What's a tarball? It's the Linux equivalent of a zip file. Use this command to build your tarball:

```
tar -czf username.tgz hello.py cylinder.py torus.py
```

Where you replace `username` with your username. If you want to find out more about the `tar` command, then `man tar` will help you out.

## Thoughts

1. While they're not actually the same thing, Unix and Linux have very similar interfaces and tools. Generally, when you see a tutorial or resource for Unix, it will apply to a Linux environment, too. Also, there are many different distributions of Linux. Again, at the level we are going to be using them, they are all largely the same.

2. Linux is case-sensitive. When we say "create a directory called `lab0`", this means that the directory should be called `lab0`. Not `Lab0`, `Lab 0`, `LAB 0`, `My Lab 0`, or `LabZero`. This might not seem like a big deal, but writing code is all about precision; you should follow the instructions exactly, since we will treat everything else as being wrong.
3. If you're looking for more tutorials, you can find a list of them [here \(Links to an external site.\)](#).
4. There are a number of `idle` tutorials on the Web. We're going to leave it as an exercise for the reader to find one or more of these and work your way through them. Google, as always, is your friend.
5. There are two versions of idle on the lab computers. `idle27` runs the Python 2.7.x version, and this is the one you should use. `idle3` runs the Python 3 version.
6. Just to be very clear, you should write code that calculates the value for the last two questions, not code that simply prints out a number you have previously typed in. Yes, someone tries this in every introductory class like this.
7. No, we didn't tell you how to calculate the volumes of the cylinder or the torus. Yes, this was on purpose. Yes, the information is probably available on the Internet somewhere, in case you don't have it at your fingertips. Yes, it's possible to search for it. Yes, this will be recurring theme in the class.

## The Rules

Everything you do for this lab should be your own work. Don't look up the answers on the web, or copy them from any other source. You can look up general information about Python on the web, but no copying code you find there. Read the code, close the browser, then write your own code.