

Lesson 1: Hunting the Snake!

Let's see how programmers charm the snake into doing their bidding. We'll track down our very own Python, and make it do math for us. Now that's a useful trick!

Hands-on labs

In this lesson you're going to learn how to navigate through the environment, and then call up the Python program to run some simple exercises.

Exercise 1-1: Start (and restart) engines!

Let's learn how to start the computer from the special USB key provided by the teacher. Most computers are set to *boot*, or load their operating system, from the hard disk storage device that's inside the computer. Most computers can be configured to look at USB slots first, and try booting from a device there. You have to hit a special key to get into the *BIOS setup*, and then set up the boot ordering correctly.

Fortunately the computer lab is already set up to do this, so we don't have to bother with any of that. Just follow these steps:

1. Insert the key in a USB slot on the computer.
2. Power the computer on, then wait while the system boots.

Yikes, where am I?

You'll notice things look a little different from your usual computer. That's because you're using a different *operating system*. The operating system, or "OS," is the stuff you usually don't notice – unless something goes wrong. The OS helps your software communicate with and use the physical parts of the computer.

The OS we're using is called *Linux* (pronounced "LIN-ucks") and believe it or not, **it's free**. That's right, it doesn't cost a dime. It's also free as in freedom, meaning you can **copy and share** it with other people, legally. No kidding!

How do I find and start programs?

Look for the *Activities* label at the top left of the screen. Just move your mouse cursor up to that corner, and the screen changes into an *overview*

mode. Move the mouse to the upper left corner again to get back to the normal screen.

There's a *lot* of software that comes free with a Linux system like Fedora. Believe it or not, all that software's free, too! We're only going to use a few tools specifically in this class, though.

Where are my tools?

The stuff we want to use is in the *Dash*, which is the area on the left. You'll see some icons there. Each icon represents an *application* or program you can run.



This is the *Text Editor*. You'll use this to write a program soon.



This is the *Terminal*. We'll interact with Python here.



This is the *Files* tool. We'll use this to find and open things.

Exercise 1-2: Running the Text Editor

These tools all start in a window, just like the OS you normally use. You'll find that Linux isn't so different from other operating systems in how it displays applications. Let's run a couple:

1. If you're not already looking at the Overview mode, move your mouse to the *Activities* hot corner to display it.
2. Locate the Text Editor application in the Dash, and click on it to start it up. The editor window appears.
3. Type some text – anything you want – into the tool.
4. Click the *Save* button in the Text Editor's menu. When you save a file with just simple, random text created in the Text Editor, it's a good idea to put **.txt** at the end of the name. Type the name **random.txt** in the file name entry bar, and then click *Save*.

5. To exit the Text Editor, you can either choose *File > Quit* from the menu, or you can click the grey X button at the upper right corner of the window. If you haven't saved your text yet, the Text Editor will ask you, so you don't lose your work.

How do we get our file back, though? It turns out that our OS remembers its name for us:

1. Open the Overview.
2. Instead of clicking anything, just type **random** and notice that your file comes up on the display, with a highlight around it.
3. Just hit the Enter key, and your file opens up automatically in the Text Editor.
4. Feel free to make changes to your file, save, and quit as many times as you like.

Tips and Hints

Keep a few things in mind as you do the remaining labs:

1. The computer is finicky about spelling and upper/lower case. If you get a weird response you didn't expect, make sure you typed everything exactly right.
2. Don't be afraid to try things on your own. It's really difficult to break a Linux OS or Python. Experiment and see what happens! If you have a question about why something happened, grab the teacher and ask.
3. Share what you learn with your classmates.

Exercise 1-3: Running Python

Let's learn how to start Python in a *console* so you can interact with it.

1. Open the Overview.
2. Click the Terminal application icon in the Dash.

The terminal will open with a line of text that looks like this:

```
[liveuser@localhost ~]$
```

You're actually running a program right now, called a *shell*. It's called a shell because, like a real shell, it's an outer layer of the OS. This particular layer lets you interact directly with the OS.

See the flashing cursor? That's a *prompt* – meaning the shell awaits your input. One of the things you can do at the shell prompt is to start a program. To do that, you type its name and hit Enter.

HINT: Any commands you're supposed to type, starting from this point, will be in **monospace bold text**. Unless the instructions say otherwise, hit the Enter key at the end of the line.

To start Python, enter the program name:

Python3

The computer responds:

```
bash: Python3: command not found...  
Similar command is: 'python3'
```

Aha! Remember the earlier warning that the computer is finicky about case? It's possible to have two different programs or commands that only have different case. For instance, **Python3**, **pyThon3**, and **PyThOn3** all represent different things to the computer. Fortunately the computer gives a hint on how to correct the error.

Try typing this instead:

python3

Now you should see some output from the Python program. The bottom line is – guess what? -- another prompt:

```
>>>
```

Once again, the computer awaits your input. Now you can start having fun with Python directly.

Exercise 1-4: Stupid snake tricks

For this lab exercise, **follow the instructions of the teacher** to make Python do some simple math problems for you.

Once you're comfortable with that material, try giving Python some math problems of your own. Here are some tips for the symbols that Python uses for math:

+ means addition

- means subtraction

* means multiplication (not the letter x)

/ means division (there's no ÷ key)

^ means exponentiation

...and you can use parentheses () too.

Bonus question!

If Python were to read the following math problem from left to right, what would it give as an answer? What does it actually answer? And what does that tell you about how Python does math?

```
10 + 8 * 2
```