

Configuring Ubuntu for Python Development

Note: the following instructions assume that you are connected to the Internet and that you have both the `main` and `universe` package repositories enabled. All unix shell commands are assumed to be running from your home directory (\$HOME). Finally, any command that begins with `sudo` assumes that you have administrative rights on your machine. If you do not — please ask your system administrator about installing the software you need.

What follows are instructions for setting up an Ubuntu 9.10 (Karmic) home environment for use with this book. I use Ubuntu GNU/Linux for both development and testing of the book, so it is the only system about which I can personally answer setup and configuration questions.

In the spirit of software freedom and open collaboration, please contact me if you would like to maintain a similar appendix for your own favorite system. I'd be more than happy to link to it or put it on the Open Book Project site, provided you agree to answer user feedback concerning it.

Thanks!

[Jeffrey Elkner](#)

Governor's Career and Technical Academy in Arlington
Arlington, Virginia

Vim

[Vim](#) can be used very effectively for Python development, but Ubuntu only comes with the `vim-tiny` package installed by default, so it doesn't support color syntax highlighting or auto-indenting.

To use Vim, do the following:

1. From the unix command prompt, run:

```
$ sudo apt-get install vim-gnome
```

2. Create a file in your home directory named `.vimrc` that contains the following:

```
syntax enable
filetype indent on
set et
set sw=4
set smarttab
map <f2> :w\!python %
```

When you edit a file with a `.py` extension, you should now have color syntax highlighting and auto-indenting. Pressing the key should run your program, and bring you back to the editor when the program completes.

To learn to use vim, run the following command at a unix command prompt:

```
$ vimtutor
```

\$HOME environment

The following creates a useful environment in your home directory for adding your own Python libraries and executable scripts:

1. From the command prompt in your home directory, create `bin` and `lib/python` subdirectories by running the following commands:

```
$ mkdir bin lib
$ mkdir lib/python
```

2. Add the following lines to the bottom of your `.bashrc` in your home directory:

```
PYTHONPATH=$HOME/lib/python
EDITOR=vim

export PYTHONPATH EDITOR
```

This will set your preferred editor to Vim, add your own *lib/python* subdirectory for your Python libraries to your Python path, and add your own *bin* directory as a place to put executable scripts. You need to logout and log back in before your local *bin* directory will be in your [search path](#).

Making a Python script executable and runnable from anywhere

On unix systems, Python scripts can be made *executable* using the following process:

1. Add this line as the first line in the script:

```
#!/usr/bin/env python3
```

2. At the unix command prompt, type the following to make *myscript.py* executable:

```
$ chmod +x myscript.py
```

3. Move *myscript.py* into your *bin* directory, and it will be runnable from anywhere.