**Carpenter Lab Python Workshop, Summer 2020**

Delivered via Zoom link: [Python Workshop](https://odu.zoom.us/j/6601751038)

Times: Day time to time xm; 4 May to 21 August

**Facilitators:**

Jemelyn Baldisimo, John Whalen, and Ivan Lopez. Contact [ilopez@odu.edu](mailto:ilopez@odu.edu), cell: 757-748-7922. Office hours by appointment via [Zoom](https://odu.zoom.us/j/6601751038).

**Description:**

This workshop will provide hands-on training to show how to use general computing tools to work more effectively on Carpenter Lab projects. It pulls together a broad range of powerful and flexible tools used for working with large or complex data sets. We will go over text file manipulation with regular expressions, programming in Python, and interaction with remote devices. The workshop is organized into three parts. The first part will review regular expressions and provide extra sessions during the week for participants who request help with UNIX shell scripting and GitHub. The second part will review Python programing. In the third part we will collaborate on two projects that will deliver a benefit to our research efforts. There are no costs associated with this workshop.

**Goals:**

1. Participants will become comfortable using computer science tools and programing techniques to improve data manipulation and analysis.

2. Participants will be able to perform the following tasks:

1. Editing/modifying text files using regular expressions
2. Basic bash scripting
3. Python programming.

**Workshop materials:**

The websites for this workshop are:

[Carpenter Lab Python Git Repository](https://github.com/philippinespire/CarpenterLabPythonWorkshop). It has all the course content, instructions, links, the book and super-fun-try-on-your-own exercises. Create a directory named clpw/ in your home directory and download this material there.

[Software Carpentry](https://software-carpentry.org/lessons/). This website provides free lessons about many aspects of computing for scientists. The lessons can be performed as self-guided tutorials. We will go over the following lessons:

[Programming with Python](https://swcarpentry.github.io/python-novice-inflammation/)

[Plotting and Programming with Python](http://swcarpentry.github.io/python-novice-gapminder/)

These optional sessions will be delivered to participants on request for:

[UNIX Shell](http://swcarpentry.github.io/shell-novice/) Modules 1-4 (1:45 hours)

[Git](https://swcarpentry.github.io/git-novice/) Modules 1-9 (2:00 hours)

The book is: Haddock, S. H. D. and Dunn, C. W. (2010). [Practical Computing for Biologists](http://practicalcomputing.org/).

Frequently used commands are in the [appendices](http://practicalcomputing.org/files/PCfB_Appendices.pdf).

**Required software:**

Download and install the following software:

1. Windows only: UNIX Terminal window emulator. (Insert link for free Windows terminal emulator here. Is PuTTY ok? Or <https://gitforwindows.org/> ?)
2. Anaconda. Download and install Anaconda for your operating system. <https://www.anaconda.com/products/individual>

Run Anaconda and from Anaconda install SPYDER. SPYDER is a free software program that let’s you type Python scripts, provides error correction and troubleshooting recommendations as you type, and lets you run scripts line by line for testing and debugging, similar to R Studio. It also serves as a text editor for opening, editing and saving scripts.

1. Text editor. Download and install the appropriate text editor for your operating system.

Mac OS: [BBedit](https://www.barebones.com/products/bbedit/)

Microsoft Windows: [Notepad++](https://notepad-plus-plus.org/)

1. Python3. Download and install the latest version of Python for your operating system. <https://www.python.org/downloads/>
2. Git. Download and install the latest version of Git for your operating system.

<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

**Changes to the Workshop:**

We reserve the right to make appropriate changes to the workshop content and schedule. We will notify participants of changes in a manner that maintains the flow of the workshop.

**Conflict resolution:**

As workshop instructors, we will make every effort to resolve conflicts or difficulties in a timely and fair manner. Issues that cannot be resolved by discussions with us should be brought to the attention of Dr. Kent Carpenter, or his designee.

**Workshop schedule:**

Part 1 Regular expressions and Shell operations

*Week 1,* (parentheses indicate time in minutes)

Computer Setup, PCFB Chapter 2 Regular Expressions: Powerful Search & Replace (35 min), Chapter 3 Exploring the Flexibility of Regular Expressions (40)

Optional, UNIX Shell, Software Carpentry Modules 1-4

Optional, Git, Software Carpentry Modules 1-9

Part 2 Programming with Python

*Week 2*

SC PwP Module 1 [Python Fundamentals](https://swcarpentry.github.io/python-novice-inflammation/01-intro/index.html) (30), SC P&PinP Module 2 [Variables and Assignment](http://swcarpentry.github.io/python-novice-gapminder/02-variables/index.html) (20), Chapter 7 Components of Programming (25)

*Week 3*

Chapter 7 Components of Programming (30), SC P&PinP Module 3 [Data Types and Type Conversion](http://swcarpentry.github.io/python-novice-gapminder/03-types-conversion/index.html) (20), Chapter 8 Beginning Python Programming (35)

*Week 4*

SC P&PinP Module 4 [Built-in Functions and Help](http://swcarpentry.github.io/python-novice-gapminder/04-built-in/index.html) (25), SC P&PinP Module 6 [Libraries](http://swcarpentry.github.io/python-novice-gapminder/06-libraries/index.html) (20), SC PwP Module 4 [Repeating Actions with Loops](https://swcarpentry.github.io/python-novice-inflammation/04-loop/index.html) (30)

*Week 5*

SC P&PinP Module 12 [For Loops](http://swcarpentry.github.io/python-novice-gapminder/12-for-loops/index.html) (25), Chapter 9 Decisions and Loops (30), SC P&PinP Module 11 [Lists](http://swcarpentry.github.io/python-novice-gapminder/11-lists/index.html) (20)

*Week 6*

SC PwP Module 5 [Storing Multiple Values in Lists](https://swcarpentry.github.io/python-novice-inflammation/05-lists/index.html) (45), SC PwP Module 7 [Making Choices](https://swcarpentry.github.io/python-novice-inflammation/07-cond/index.html) (30)

*Week 7*

SC P&PinP Module 13 [Conditionals](http://swcarpentry.github.io/python-novice-gapminder/13-conditionals/index.html) (25), Chapter 10 Reading & Writing files to page 188 (50)

*Week 8*

SC PwP Module 8 [Creating Functions](https://swcarpentry.github.io/python-novice-inflammation/08-func/index.html) (30), SC P&PinP Module 16 [Writing Functions](http://swcarpentry.github.io/python-novice-gapminder/16-writing-functions/index.html) (25), Chapter 10 Reading & Writing files from page 188 (20)

*Week 9*

SC PwP Module 9 [Errors and Exceptions](https://swcarpentry.github.io/python-novice-inflammation/09-errors/index.html) (30), SC PwP Module 10 [Defensive Programming](https://swcarpentry.github.io/python-novice-inflammation/10-defensive/index.html) (40), SC P&PinP Module 17 [Variable Scope](http://swcarpentry.github.io/python-novice-gapminder/17-scope/index.html) (20)

*Week 10*

Chapter 11 Merging Files (40), Chapter 12 Modules and Libraries (35)

*Week 11*

SC PwP Module 11 [Debugging](https://swcarpentry.github.io/python-novice-inflammation/11-debugging/index.html) (50), Chapter 13 Debugging (25)

*Week 12*

Part 3 Group projects “Hackathon”

*Week 13*

Group project (75)

*Week 14*

Group project (75)

*Week 15*

Group project (75)

*Week 16*

Group project (75)