Fork me on GitHub
Software Carpentry banner

University of Queensland

Jul 13-14, 2015

9:00 am - 5:00 pm

Instructors: Damien Irving, Rob Johnson

Helpers: Scott Wales, Holger Wolff

General Information

The mission of the Software Carpentry project is to help scientists be more productive by teaching them basic computing skills. Scientists often spend much of their time wrestling with software, but most are self-taught programmers. As a result, they spend hours doing things that should take minutes, reinvent a lot of wheels, and still don't know if their results are reliable. To tackle this problem, Software Carpentry runs two-day workshops at hundreds of sites around the world. These hands-on workshops cover basic concepts and tools, including program design, version control, data management, and task automation. Participants are be encouraged to help one another and to apply what they have learned to their own research problems. To coincide with the Australian Meteorological and Oceanographic Society (AMOS) National Conference, AMOS and the ARC Centre of Excellence for Climate System Science are teaming up to deliver this workshop.

Who: The workshop is restricted to postgraduate students (including honours) and other research scientists working in the AMOS sciences (i.e. meteorology, climatoloogy, oceanography, glaciology) who are familiar with basic programming concepts like loops, conditionals, arrays, and functions, but need help to translate this knowledge into practical tools to help them work more productively.

Where: Planning Studio, Rooms 314 & 315, Level 3, Steele Building. Get directions with OpenStreetMap or Google Maps.

Requirements: Participants must bring a laptop with a few specific software packages installed (listed below). That laptop will need to be able to connect to the Eduroam wifi network, so please let the organisers know if you aren't affliated with an Australian tertiary instition and thus cannot access Eduroam. For help connecting to Eduroam, participants should search the IT website of their home institution, as opposed to the University of Queensland IT website. Participants are also required to abide by Software Carpentry's Code of Conduct.

Registration: Participants can register at the AMOS conference website at the same time as registering for the conference itself. The fee is $90 for students and $140 otherwise, which is used to pay an administratve fee to the Software Carpentry Foundation and to assist the volunteer instructors with their accommodation expenses. At the registration site it is possible to register for the workshop alone (i.e. and not the conference as well), however just note that the workshop is only open to people working/studying in the AMOS sciences. Registrations for the workshop will close on Friday 12 June or when the venue capacity (50 people) has been reached, whichever comes first. Note that it is not anticipated that people who attended the Software Carpentry workshop prior to the AMOS conference in Hobart (Feb 2014) will also attend this one, as the teaching content will be very similar.

Contact: Please mail irving.damien@gmail.com for more information.

Schedule

Day 1

09:00 - 10:30	Building programs with Python
10:30 - 11:00	Morning tea
11:00 - 12:30	Building programs with Python
12:30 - 01:30	Lunch break
01:30 - 03:00	Building programs with Python
03:00 - 03:30	Afternoon tea
03:30 - 05:00	Unix shell / command line programs

Day 2

09:00 - 10:30	Version control with Git
10:30 - 11:00	Morning tea
11:00 - 12:30	Data management in the AMOS sciences
12:30 - 01:30	Lunch break
01:30 - 03:00	Data management in the AMOS sciences
03:00 - 03:30	Afternoon tea
03:30 - 05:00	Workflow automation with Make

Food: In order to keep the registration fees down, morning tea, afternoon tea and lunch are not provided (there are many places to find food on the UQ campus).

Etherpad: https://etherpad.mozilla.org/2015-07-13-amos.
We will use this Etherpad for chatting, taking notes, and sharing URLs and bits of code.

Syllabus

Programming in Python

- Using libraries
- Working with arrays
- Reading and plotting data
- Creating and using functions
- Loops and conditionals
- Defensive programming
- Using Python from the command line
- Lesson notes...

The Unix Shell

- Files and directories
- History and tab completion
- Pipes and redirection
- Looping over files
- Creating and running shell scripts
- Finding things
- Lesson notes...

Version Control with Git

- Creating a repository
- Recording changes to files: `add`, `commit`, ...
- Viewing changes: `status`, `diff`, ...
- Working on the web: `clone`, `pull`, `push`, ...
- Resolving conflicts
- Lesson notes...

Data Management in AMOS Sciences

- Creating a Data Reference Syntax
- netCDF files and their attributes
- Data provanence and metadata
- Commonly used Python libraries in the AMOS sciences
- Lesson notes...

Workflow automation with Make

- Makefiles
- Automatic variables
- Dependencies
- Pattern rules
- Variables
- Functions
- Lesson notes...

Setup

1. Text Editor

When you're writing code, it's nice to have a text editor that is optimized for writing code, with features like automatic color-coding of key words. The default text editor on Mac OS X and Linux is usually set to Vim, which is not famous for being intuitive. if you accidentally find yourself stuck in it, try typing the escape key, followed by ':q!' (colon, lower-case 'q', exclamation mark), then hitting Return to return to the shell.

Windows

`nano` is the editor installed by the Software Carpentry Installer, it is a basic editor integrated into the lesson material.

Notepad++ is a popular free code editor for Windows. Be aware that you must add its installation directory to your system path in order to launch it from the command line (or have other tools like Git launch it for you). Please ask your instructor to help you do this.

Mac OS X

We recommend Text Wrangler or Sublime Text. In a pinch, you can use `nano`, which should be pre-installed.

Linux

Kate is one option for Linux users. In a pinch, you can use `nano`, which should be pre-installed.

2. The Bash Shell

Bash is a commonly-used shell that gives you the power to do simple tasks more quickly.

Windows

Install Git for Windows by download and running the installer. This will provide you with both Git and Bash in the Git Bash program.

Software Carpentry Installer

This installer requires an active internet connection.

After installing Python and Git Bash:

- Download the installer.
- If the file opens directly in the browser select **File→Save Page As** to download it to your computer.
- Double click on the file to run it.

Mac OS X

The default shell in all versions of Mac OS X is bash, so no need to install anything. You access bash from the Terminal (found in `/Applications/Utilities`). You may want to keep Terminal in your dock for this workshop.

Linux

The default shell is usually `bash`, but if your machine is set up differently you can run it by opening a terminal and typing `bash`. There is no need to install anything.

3. Git

Git is a version control system that lets you track who made changes to what when and has options for easily updating a shared or public version of your code on github.com.

Windows

Git should be installed on your computer as part of your Bash install (described above).

Mac OS X

For OS X 10.8 and higher, install Git for Mac by downloading and running the installer. After installing Git, there will not be anything in your `/Applications` folder, as Git is a command line program. **For older versions of OS X (10.5-10.7)** use the most recent available installer for your OS available here. Use the Leopard installer for 10.5 and the Snow Leopard installer for 10.6-10.7.

Linux

If Git is not already available on your machine you can try to install it via your distro's package manager. For Debian/Ubuntu run `sudo apt-get install git` and for Fedora run `sudo yum install git`.

4. Python

Python is a popular language for scientific computing, and great for general-purpose programming as well. Installing all of its scientific packages individually can be a bit difficult, so we recommend an all-in-one installer.

Windows

- Download and install Anaconda.
- Use all of the defaults for installation *except* make sure to check **Make Anaconda the default Python**.

Mac OS X

- Download and install Anaconda.
- Use all of the defaults for installation *except* make sure to check **Make Anaconda the default Python**.

Linux

We recommend the all-in-one scientific Python installer Anaconda. (Installation requires using the shell and if you aren't comfortable doing the installation yourself just download the installer and we'll help you at the boot camp.)

1. Download the installer that matches your operating system and save it in your home folder.
2. Open a terminal window.
3. Type

   ```
   bash Anaconda-
   ```

   and then press tab. The name of the file you just downloaded should appear.
4. Press enter. You will follow the text-only prompts. When there is a colon at the bottom of the screen press the down arrow to move down through the text. Type `yes` and press enter to approve the license. Press enter to approve the default location for the files. Type `yes` and press enter to prepend Anaconda to your `PATH` (this makes the Anaconda distribution the default Python).

5. Make

Originally invented to manage compilation of programs written in languages like C, Make can be used to automatically update any set of files that depend on another set of files. This makes it a good solution for many data analysis and data management problems. While there are many build tools now in existence (e.g. ANT and CMake) they share the same fundamental concepts as Make.

Windows

If you used the "Software Carpentry installer" to install the Bash Shell (see above), then you already also have Make installed.

Otherwise, once you have installed Git Bash you can install Make by:

1. Download `make.exe` from here
2. Place it in the `bin` directory where you installed Git Bash e.g. `C:\Program Files (x86)\Git\bin`.
3. To test: open a Git Bash window, type `make`, and press Enter.
4. You should see the following message

   ```
   make: *** No targets specified and no makefile found. Stop.
   ```

   This means that Make was successfully installed. Otherwise, you'll see this error message:

   ```
   bash: make: command not found
   ```

Mac OS X

For OS X, version 10.9 (Mavericks) or above, download the Command Line Tools by doing:

```
xcode-select  --install
```

For more information, see the OSX Daily blog.

If you have an older OS X version and you do not already have access to `make` from within your shell, you will need to install XCode (which is free, but over a gigabyte to download).

- Go to the Apple app store
- Search for XCode
- Click **Free**
- Click **Install App**

Once XCode has installed:

- Click **Applications**
- Click **XCode**
- Select **XCode→Preferences...**
- Click **Downloads**
- Select **Command Line Tools**
- Click **Install**

You will now be able to run `make` within your shell.

Linux

Make is a standard tool on Linux systems and should already be available.