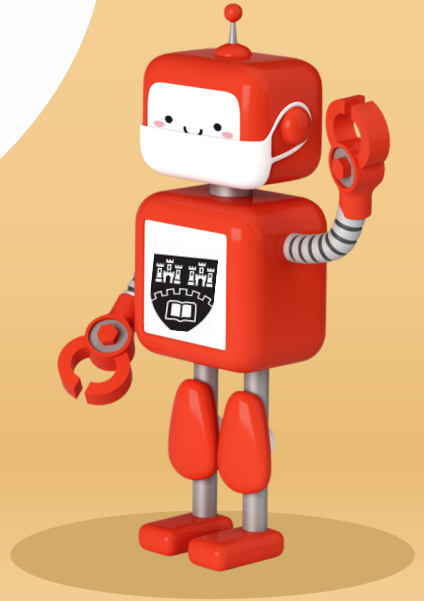
Northumbria University NEWCASTLE

Software Sustainability Institute

Software Carpentry in (and out of) Lockdown



Lucy Whalley

lucydot.github.io

< Outline >

1. Introduction to Software Carpentry
2. Case Study: Python for physicists
3. Some observations in and out of Lockdown

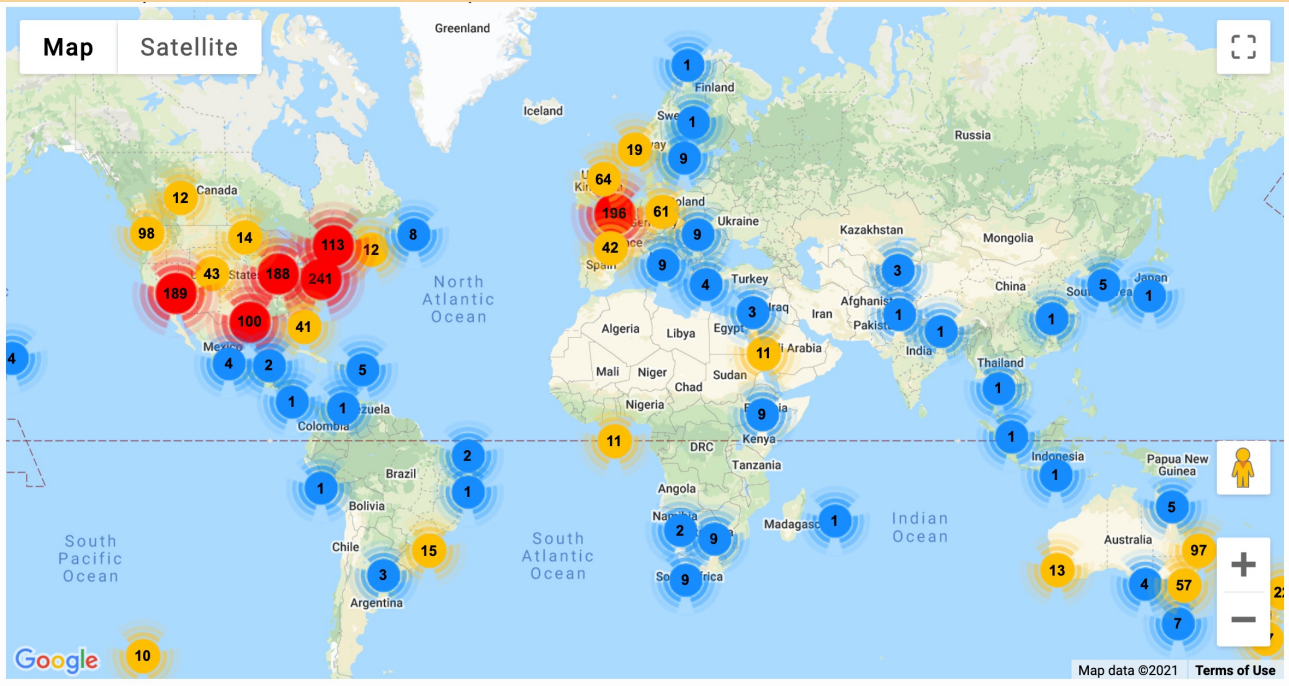
< Outline >

1. **Introduction to Software Carpentry**
2. Case Study: Python for physicists
3. Some observations in and out of Lockdown

“Since 1998, Software Carpentry has been teaching researchers **the computing skills they need to get more done in less time and with less pain**. Our volunteer instructors have run hundreds of events for more than **34,000 researchers since 2012**. All of our lesson materials are freely reusable under the Creative Commons – Attribution License.”

From https://software-carpentry.org/about/

34,000 researchers since 2012



Lesson materials are freely reusable

swcarpentry / git-novice

Sponsor | Watch 98 | Star 229 | Fork

Code | Issues 77 | Pull requests 40 | Actions | Projects | Security | Insights

gh-pages | 6 branches | 2 tags | Go to file | Add file | Code

munkm Merge pull request #790 from karenword/patch-1 ... 8d0edab 4 days ago 2,237 commits

.github	Update ISSUE_TEMPLATE.md	2 years ago
_episodes	Correct and add alt text	4 days ago
_extras	Merge pull request #680 from markmatney/gh-pages	5 months ago
_includes	aio: Update image links	2 years ago
_layouts	use ndash as separator	2 years ago
assets	Use darker purple for code blocks	2 years ago
bin	lesson-check: exclude aio.md, fix read_references	2 years ago
fig	Merge pull request #722 from juholehtonen/update-figures	10 months ago
.editorconfig	.editorconfig: code style guidelines for text editors	3 years ago
.gitignore	gitIgnore for R	3 years ago
.mailmap	Updating .mailmap and AUTHORS before release	4 years ago

About

Version Control with Git

swcarpentry.github.io/git-novice/

git | programming | version-control | versioning | english | stable | lesson | software-carpentry | carpentries

Readme

View license

Releases 2

Software Carpentry: Versi... Latest
on 1 Jul 2019

+ 1 release

Sponsor this project

Software Carpentry teaches the computing skills needed to get more done in less time and with less pain

Day 1		Day 2	
10:00	Automating tasks with the Unix shell	10:00	Plotting and Programming with Python 1
11:30	Coffee	11:30	Coffee
13:00	Lunch break	13:00	Lunch break
14:00	Version control with Git	14:00	Plotting and Programming with Python 2
15:30	Coffee	15:30	Coffee
17:00	END	17:00	END

"[The time is takes] to produce a new result is frequently dominated by how long it takes to write, test, debug, install, and maintain software."

Software Carpentry: lessons learned, Greg Wilson

< Outline >

1. Introduction to Software Carpentry
2. **Case Study: Python for physicists**
3. Some observations in and out of Lockdown

< Case Study >

Python for Physicists

- One day workshop closely modelled on the Software Carpentry Python course
- Adapted for the physical sciences, using UV-Vis data as the motivating example
- Graduates in the physical sciences: physics, chemistry, engineering
- `lucydot.github.io/python_novice`

< Case Study >

Python for Physicists

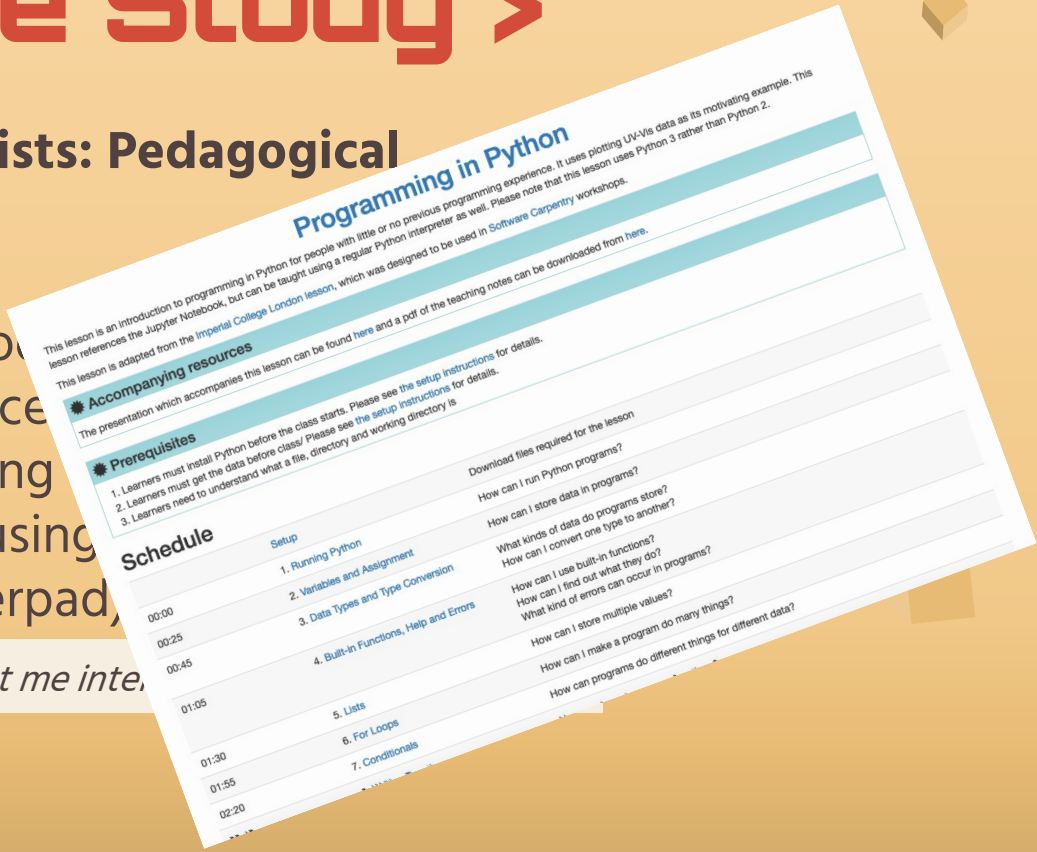
- Delivered in 2019 to ~20 students in the Centre for Doctoral Training in New and Sustainable Photovoltaics [in person]
- Delivered in 2020 to ~20 students in the Centre for Doctoral Training in Renewable Energy Northeast Universities [online]

< Case Study >

Python for Physicists: Pedagogical approaches

- Live coding: I co
- PPT for key conce
- Pair programming
- Open lessons (using
  Notebook, Etherpad,

"the interactiveness kept me inte



Programming in Python

This lesson is an introduction to programming in Python for people with little or no previous programming experience. It uses plotting UV-Vis data as its motivating example. This lesson references the Jupyter Notebook, but can be taught using a regular Python interpreter as well. Please note that this lesson uses Python 3 rather than Python 2.

This lesson is adapted from the Imperial College London lesson, which was designed to be used in Software Carpentry workshops.

Accompanying resources

The presentation which accompanies this lesson can be found here and a pdf of the teaching notes can be downloaded from here.

Prerequisites

1. Learners must install Python before the class starts. Please see the setup instructions for details.
2. Learners must get the data before class/ Please see the setup instructions for details.
3. Learners need to understand what a file, directory and working directory is

Schedule

	Setup	Download files required for the lesson
00:00	1. Running Python	How can I run Python programs?
00:25	2. Variables and Assignment	How can I store data in programs?
00:45	3. Data Types and Type Conversion	What kinds of data do programs store? How can I convert one type to another?
01:05	4. Built-in Functions, Help and Errors	How can I use built-in functions? How can I find out what they do? What kind of errors can occur in programs?
01:30	5. Lists	How can I store multiple values?
01:55	6. For Loops	How can I make a program do many things?
02:20	7. Conditionals	How can programs do different things for different data?

< Outline >

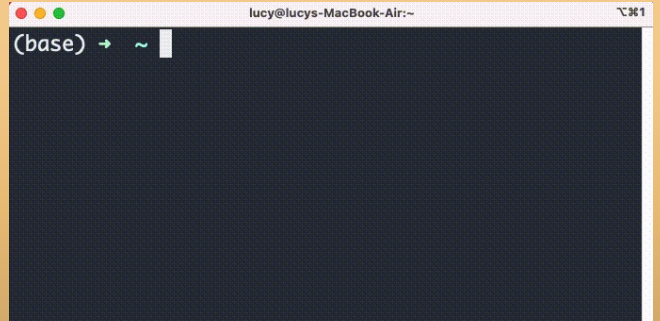
1. Introduction to Software Carpentry
2. Case Study: Python for physicists
3. **Some observations in and out of Lockdown**

Live coding works well both in and out of lockdown.

- Live coding:
  - Slows the instructor down
  - Is a flexible teaching tool (“what if…” questions)
  - Live coding helps to normalise making mistakes

“thought it was helpful to give examples of code breaking”

““good that there was no such thing as stupid questions”



Managing a diversity of skill levels is a work in progress.

- Students arrive with different skill levels.
- I send an email out in advance of the workshop, emphasising that this is a course for those with little or no programming experience

“felt it was generally a very good pace for my level of coding. “

- I invite those with experience to be helpers in the workshop (two birds, one stone as recruiting helpers through other means can be difficult).

“could have had different “difficulties“ for tasks so that people more familiar with coding/Python were still challenged”

One-to-one support was more difficult in lockdown

- Support is available:
  - before the workshop (for installation issues)
  - during the workshop (via sticky notes, helpers and pair programmer)
  - during the break (I have to initiate the conversation!)
  - one month after the course (with little uptake – one student).

> *"very enjoyable and easy to follow, feel quite confident to try some more involved programming now"*

- In lockdown it was more difficult to tailor one-to-one support
  - A dedicated break out room was available for 1:1 support from helpers.

For time-pressed students motivation was key.

"Why can't we just use excel?"

- Dedicate time at the start of the workshop to discuss:
  - The limitations of spreadsheets
  - Reproducibility and repeatability
- Use tools that have quick reward:
  - Pandas and matplotlib for Python analysis and plotting

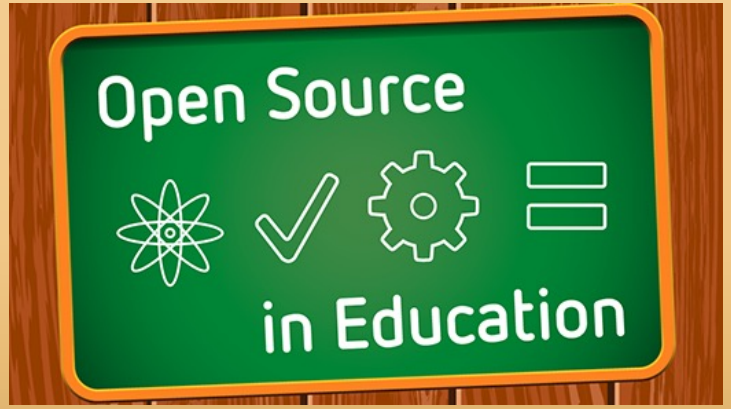
Summary

- Software Carpentry teaches the computing skills that underlie scientific research and computational project work
- One-to-one support is more challenging in-lockdown
- The “open lesson” approach has meant that pivoting to online delivery has been relatively painless
- Live coding works well in and out of lockdown, encouraging participation and normalising mistakes
- For Software Carpentry course materials, and to request a workshop at your institution, visit `software-carpentry.org`

Final note...

Why don’t we work together more?

Software Carpentry has benefited from the input of many different educators and researchers from across the world. Could we deliver better quality education by collectively developing more of our university-level courses?



Thank-you

Slides available at my website

Email: l.whalley@northumbria.ac.uk

Twitter: @lucydotwhalley

Github: @lucydot

Web: lucydot.github.io