The Hacker Toolkit

Matthew Brett

April 16, 2018

The toolkit metaphor

Since 1998, Software Carpentry has been teaching researchers the computing skills they need to get more done in less time and with less pain – Software Carpentry website.

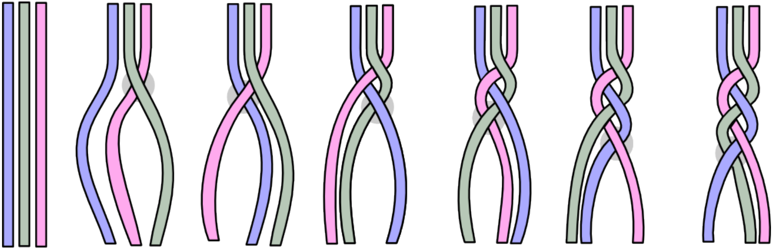
The journey is long

Teach Yourself Programming in Ten Years

Simple compared to easy

The roots of "simple" are "sim" and "plex", and means "one twist". The opposite, which would be complex, is "multiple twists" or "braided together" . . . The latin origin of "easy" is the root of "adjacent", which means "to lie near" and "to be nearby" – Rich Hickey "Simple Made Easy"

Simple compared to easy

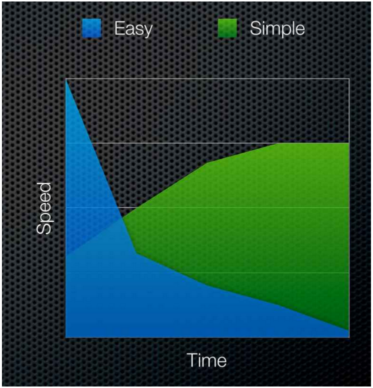


Simple: clarity, efficiency, basis for learning.

You will be tempted to keep doing it the easy way

A couple of months in the laboratory can frequently save a couple of hours in the library – Frank Westheimer (link)

But this will have terrible long term cost



From presentation Simple Made Easy

Choosing your tools

- Find efficient people;
- Learn from them!
- Read - e.g. Wilson *et al* (2017) [Good enough practices in scientific computing] (http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcb *PLoS Comp Bio*

Choosing tools - simple rather than easy

3. THE BASIC TOOLS
    14. The Power of Plain Text
    15. Shell Games
    16. Power Editing
    17. Source Code Control
    18. Debugging
    19. Text Manipulation
    20. Code Generators

The Pragmatic Programmer

Shell games

- Use the (probably Unix) command line for everything you can;

Power editing

Use a Single Editor Well

The editor should be an extension of your hand; make sure your editor is configurable, extensible, and programmable – The Pragmatic Programmer

Source code control

You may not know it yet, but you will need:

- git;
- (something like) Github.

“FINAL”.doc at
http://phdcomics.com/comics/archive.php?comicid=1531

A general purpose programming language

Prefer open: popular options are:

- Python
- R / R Studio

Standard programming tools

- Testing;
- Continuous integration;
- Process automation with `make` and shell scripting.

Notebooks

- The Jupyter Notebook;
- The R Notebook

For example, this presentation

- Uses plain text for everything;
- I wrote it using the Vim editor;
- Is stored with Git and uploaded to Github;
- Uses the Unix command line and `make` for building.

Your tool here

Over to you.