Computational Fluency Workshop

Day 2: Beginning with structure, continued

https://github.com/brownritt/cfw2022

Jason Ritt
jason_ritt@brown.edu
Scientific Director of Quantitative Neuroscience

ROBERT J. & NANCY D. CARNEY
INSTITUTE FOR BRAIN SCIENCE
BROWN UNIVERSITY

Recall: What is a computer?

A central processor (**CPU**), and often auxillary processors (e.g. graphics processing units, **GPU**s)

Memory, for fast and transient work

Storage (hard drives, thumb drives, etc), organized as a ***File system***, for slow and permanent work

Devices (e.g. keyboard, screen, WiFi interface…) for interfacing with the world

Processes, many task-specific programs, interacting with each other

Every command invokes some set of the following questions:

Who am I? ***Accounts***
What am I allowed to do? ***Priviledges***
Where am I? ***Working directory***
Where is the file that I want to run or access? ***Paths***
What kind of thing is the file I want to run or access? ***File formats***

Helpful steps when starting a new project

For a "local only" `git` repo; see the reference doc for an order of steps to start a GitHub enabled project

Pick a good name: short, descriptive, and unique; no spaces or weird characters

Set up a folder skeleton (manually, or with a utility like `cookiecutter`) before adding any important files

Add a README file (in "markdown"), that explains the purpose of the project

Initialize `git`, make an initial commit

Make virtual environment (depends on language), capture to a `requirements.txt` or `environments.yml` file in the top project directory and commit

Copy any files you need from other locations, commit the changes

Do cool science

Some thoughts on different kinds of versioning

Updates: You are making the code "better", or more general, or otherwise altering in a way where you will not need the older versions except if you find bugs or other problems.

New contexts: Your analysis code works on Experiment 1, but now you have data from Experiment 2 that differs in important ways (e.g. you added stimulation to what was neural recording and behavior). You want to adapt to Expt 2 without disrupting the working pipeine for Expt 1.

The key distinction is whether older versions remain "active" or "valid"; think carefully about what you expect from the future.

Note that modularity and hierarchical organization in general helps reduce the complexity of versioning choices.

Live demos and exercises

Processes

You are here $\longrightarrow$ Shells, command line interfaces, the REPL framework

IDEs over editors

Exercise: pupillometry analysis, setting up a new project, beginnings of versioning