Lab 0: Setup

Ahmed Eldeeb

2023-08-21

Table of contents

Warning: This page is under construction for 2024.

Setting up your environment (08/25/23)

You will need to set up (or make sure you have access to) the following:

1. Unix shell
2. Git
3. Quarto
4. Python
5. A text editor or IDE of your choice (deeb recommends VS Code or sublime, but if you are already familiar with a specific editor, stick with it)

After making sure you have access to all these 5 tools, it may be a good idea to go through some of the following tutorials on unix bash and unix commands. You can do this on your own or in the Lab section on Friday 8/25.

Attending lab 0 is optional. If you successfully set up your environment to have all the listed tools, you don't need to attend. You are welcome to attend to ask for help with the setup or to help your classmates with setting up their environments.

You are also welcome to come and ask for help on using any of these 5 tools/systems (esp. bash and unix commands this week), but priority will be given to environment setup questions.

Reach out to deeb @ deeb@berkeley.edu with any unresolved problems or if you discover something that needs to be changed with our howtos and instructions.

Deeb's unsolicited advice on languages and tools:

1. Whichever editor you pick, make sure to spend some time every week learning a few keyboard shortcuts for it. The same goes for the bash shell (ctrl+a and ctrl+e are among my favorites) and for your OS of choice in general (and even your web browser!). Not only do keyboard shortcuts make you more efficient, but they dramatically reduce the cognitive load after a while, and so

make your life less painful in the long run. They can be the difference between hating computers and loving them.

2. Programming languages come and go, but Unix is forever! Well, maybe not forever, but close enough. Invest more of your time in getting familiar with durable and proven paradigms. Different programming languages are suitable in different situations and change dramatically from one decade to the next, but the unix shell and commands are as pristine, long lived, and as widely applicable as you'll find in the computing world. I have a much more mixed view of git, Python, R, and C++. Another example of a very durable computing paradigm is SQL, which we'll get to in a few weeks.