This will almost entirely a live-code session, and one of the first times we “shatter the assumptions” of learners about how their computers really work.

* Overview of lesson—command line review, Q&A GitHub review, git review, Q&A
* Command line review
* **NOTE: this review will be done in the MacOS terminal, but all of these commands apply to bash shells on other operating systems and we’ll talk about that in a minute.** 
  + What is the terminal? The terminal app is one terminal emulator
  + Where a “computer terminal” is the “interface” to the command line.
  + This interface is purely text.
  + It allows you to run commands on a computer in a “headless” environment, the same commands that your computer would do whether you had a really nice interface or not.

* + Why do we need it?
    - easy to use
    - **much** faster than clicking around in a GUI
    - scripting
    - composability
  + Differences in CLIs in Windows, Mac, Linux
    - CMD ≠ Powershell ≠ bash ≠ Git for Windows
    - Unix (e.g., macOS) ≠ Linux (see <https://upload.wikimedia.org/wikipedia/commons/7/77/Unix_history-simple.svg>)
    - Recommended: bash on Windows 10 <https://msdn.microsoft.com/en-us/commandline/wsl/install_guide>
    - Git on Windows: https://git-scm.com/download/win
  + Brief Review
    - ls
    - pwd
    - cd
    - touch
    - mkdir
    - touch
    - clear (use Ctrl-L to speed things up)
  + Helpful other commands
    - mv
    - cp
    - rm
    - man
    - nano
    - Invite students to look into grep
  + Invite students to try using the command line exclusively for a while, share Linux CLI getting started guide here: <https://linux.die.net/Linux-CLI/>
  + GNU Coreutils: <http://www.gnu.org/software/coreutils/manual/html_node/index.html#SEC_Contents>

Small explanation of my shell environment and how commands are “found”.

* Overview of Git
  + Purpose
    - Backup system
    - Manage versions
    - Discover bugs
    - Work cooperatively
    - Find out why something was coded
  + History
    - Linus Torvalds (created git and the kernel - distribute wow factor to learners, etc.)
    - Other version control systems, (subversion, CVS, mercurial, SCCS)
  + DVCS
    - Distributed version control
    - Remote vs local repositories
  + Git ≠ Github
  + Github vs. Gitlab vs. Bitbucket
  + Illustrate git with a live drawing of tree with branches; show SHA-1 examples, authors, branch names, merging
* **Q&A**
* Git review
  + Livecode
    - Begin project to use through remainder of course, separate by branches
    - Create local git repository, add a few commits
      * In lesson
        + Git init
        + Git add
        + Git commit
      * New commands
        + Git config
        + Git reflog
  + Set up Github repository
    - Don’t initialize with anything; we already have our local repository
    - Git remote add REMOTE\_NAME REMOTE\_URL
      * Difference between https and ssh; easier for students to use https starting out
      * Refer students to https://help.github.com/articles/connecting-to-github-with-ssh/
    - git push REMOTE SOURCE:TARGET
  + Github pages
    - Git push REMOTE SOURCE:gh-pages
    - Accessing github pages
    - Refer students to <https://pages.github.com/> for more information

**FINAL Q&A**

* If more time
  + git branch
  + git checkout
  + git blame
  + git merge
  + .gitignore

**More resources and links to share**

Bash: <https://www.gnu.org/software/bash/>

Zsh: <http://www.zsh.org/>

Bash configuration: <http://www.linuxfromscratch.org/blfs/view/8.0/postlfs/profile.html>

[https://upload.wikimedia.org/wikipedi...](https://upload.wikimedia.org/wikipedia/commons/7/77/Unix_history-simple.svg)

[https://msdn.microsoft.com/en-us/comm...](https://msdn.microsoft.com/en-us/commandline/wsl/install_guide)

[https://git-scm.com/download/win﻿](https://git-scm.com/download/win)

[https://linux.die.net/Linux-CLI/﻿](https://linux.die.net/Linux-CLI/)

[http://www.tldp.org/LDP/Linux-Filesys...](http://www.tldp.org/LDP/Linux-Filesystem-Hierarchy/html/)

[https://www.gnu.org/software/coreutil...](https://www.gnu.org/software/coreutils/manual/html_node/index.html#SEC_Contents)

[https://github.com/goj/coreutils﻿](https://github.com/goj/coreutils)

[https://github.com/DavidAwad/sample-r...](https://github.com/DavidAwad/sample-repo)

Bash: <https://www.gnu.org/software/bash/Zsh:> <http://www.zsh.org/Bash>configuration: [http://www.linuxfromscratch.org/blfs/...](http://www.linuxfromscratch.org/blfs/view/8.0/postlfs/profile.html)

[https://git-scm.com/﻿](https://git-scm.com/)

[https://pages.github.com/﻿](https://pages.github.com/)

​[https://github.com/DavidAwad/davidawa...](https://github.com/DavidAwad/davidawad.github.io)

**Learning Standards to double check at the end to wrap up**

* **Learning Standard**: [State the differences between local and remote repositories](https://www.codecademy.com/en/courses/freelance1-u3/lessons/learn-git-workflow/exercises/remote-repository)
* **Learning Standard**: Create versions of code stored in a local Git repository
* **Learning Standard**: Store versions of code in a remote Git repository
* **Learning Standard**: Deploy to the web using GitHub Pages