**Day 5: Git and GitHub**

**On this page**

* [**Reminders for Tony**](https://cprg210.github.io/schedule/day-05-git-github/#reminders-for-tony)
* [**Prep**](https://cprg210.github.io/schedule/day-05-git-github/#prep)
* [**Prerequisites**](https://cprg210.github.io/schedule/day-05-git-github/#prerequisites)
* [**Goal for today**](https://cprg210.github.io/schedule/day-05-git-github/#goal-for-today)
* [**Lecture 1: Command line basics**](https://cprg210.github.io/schedule/day-05-git-github/#lecture-1-command-line-basics)
  + [Live-code objectives](https://cprg210.github.io/schedule/day-05-git-github/#live-code-objectives)
  + [Activity 1: Navigating the file system](https://cprg210.github.io/schedule/day-05-git-github/#activity-1-navigating-the-file-system)
* [**Lecture: Your first git repository (maybe)**](https://cprg210.github.io/schedule/day-05-git-github/#lecture-your-first-git-repository-maybe)
  + [Top Git commands](https://cprg210.github.io/schedule/day-05-git-github/#top-git-commands)
  + [Live-code objectives](https://cprg210.github.io/schedule/day-05-git-github/#live-code-objectives-1)
  + [Activity: Project work](https://cprg210.github.io/schedule/day-05-git-github/#activity-project-work)
* [**Mid-day Huddle**](https://cprg210.github.io/schedule/day-05-git-github/#mid-day-huddle)
* [**GitHub Tour**](https://cprg210.github.io/schedule/day-05-git-github/#github-tour)
* [**Activity 3: Publish your website to GH Pages**](https://cprg210.github.io/schedule/day-05-git-github/#activity-3-publish-your-website-to-gh-pages)
* [**Summary**](https://cprg210.github.io/schedule/day-05-git-github/#summary)

**Reminders for Tony**[**Permalink**](https://cprg210.github.io/schedule/day-05-git-github/#reminders-for-tony)

1. Put on mic
2. Open Zoom chat
3. Record! Anyone else?

**Prep**[**Permalink**](https://cprg210.github.io/schedule/day-05-git-github/#prep)

* [CLI (command line interface) Cheat Sheet](https://www.git-tower.com/blog/command-line-cheat-sheet/)
* [Configuring Git](https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup)
* [Git and GitHub for Poets](https://www.youtube.com/playlist?list=PLRqwX-V7Uu6ZF9C0YMKuns9sLDzK6zoiV)
* [Why does GitHub recommend HTTPS over SSH?](https://stackoverflow.com/questions/11041729/why-does-github-recommend-https-over-ssh)

**Prerequisites**[**Permalink**](https://cprg210.github.io/schedule/day-05-git-github/#prerequisites)

* Git installed
* VS Code installed
* Github Account created

**Goal for today**[**Permalink**](https://cprg210.github.io/schedule/day-05-git-github/#goal-for-today)

By the end of the day, you should have a local project repository “pushed” and synced to the remote copy located on GitHub.

1. Create a new repository on Github. It should have a similar name to your local project.
   * It needs to be public for GH Pages to work
   * DO NOT initialize with a README.md
   * Click “Create repository”. Save the next page; it has important info.
2. After creating your repo, you should see a list of instructions for different situations. We will start with the first one: “…or create a new repository on the command line”.

But FIRST! We need to learn some command line…

**Lecture 1: Command line basics**[**Permalink**](https://cprg210.github.io/schedule/day-05-git-github/#lecture-1-command-line-basics)

Most of the command line tools we will be using in this course (like Git, [Node](https://nodejs.org/en/) and [NPM](https://www.npmjs.com/)) will depend greatly on which folder you are in. We will cover the three system commands that help up us navigate the file system.

Warning: There are two flavours of “command line”: Windows and Unix. We will be using the latter: Git Bash (installed along with Git) on Windows and Terminal on Mac (Mac is unix-based so almost any terminal app will do; Tony uses iTerm).

* pwd
  + displays your ‘present working directory’. This will usually default to your home folder when you first open your terminal.
* ls
  + Lists the contents of your current directory.
  + The -l flag lists extra information and -a lists hidden files. -la will list extra information and also all hidden files. Examples:
    - $ ls -l
    - $ ls -a
    - $ ls -la
* cd: changes your directory based on the path you provide after the command (separated by a space). Examples:
  + $ cd Documents/cprg210
  + $ cd /h/sait/cprg210
  + $ cd .. (moves up one directory)

Quality of life cli tips

* tab: auto-completes file names and directories
* up arrow: browse through command history

**Live-code objectives**[**Permalink**](https://cprg210.github.io/schedule/day-05-git-github/#live-code-objectives)

* What is the starting directory when you open a terminal on your system?
* What files and directories are in this initial directory?
* What file did you last open? Try to find it using only the terminal.
* What’s at the top of your directory structure? How do you get there?
* Using the terminal, find the directory you’ve chosen to store your cprg 210 projects.

**Activity 1: Navigating the file system**[**Permalink**](https://cprg210.github.io/schedule/day-05-git-github/#activity-1-navigating-the-file-system)

You will be working in teams. Make sure you have installed Git on your system. We will be using Git Bash for this exercise.

Try navigating to various directories on your system to get accustomed to the command line.

**Lecture: Your first git repository (maybe)**[**Permalink**](https://cprg210.github.io/schedule/day-05-git-github/#lecture-your-first-git-repository-maybe)

Let’s go back to that page of instructions we received after creating our **GitHub** repo. Remember, we’ll be starting with the top option: “…or create a new repository on the command line”.

**Top Git commands**[**Permalink**](https://cprg210.github.io/schedule/day-05-git-github/#top-git-commands)

* $ git status: Show the current status of your repository (repo)
* $ git add: Add file contents to be committed
* $ git commit: Record changes to the repository
* $ git push: Update remote refs along with associated objects
* $ git pull: Fetch from and integrate with another repository or a local branch

**Live-code objectives**[**Permalink**](https://cprg210.github.io/schedule/day-05-git-github/#live-code-objectives-1)

* How will you organize your projects?
* If you’ve just installed Git, [set your name and email](https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup) using the git config command described in the Prep.
* Create a git repository for the project you’ve been working on in the HTML/CSS portion of this course.
  1. Navigate to your project directory on the command line.
  2. git status to make sure you are not already in a repo.
  3. git init to initialize a new repo. This is usually done only once (if that) per project.
  4. git status to see what files are modified/deleted/untracked.
  5. git add . to add all your files for an initial commit
  6. git status to double-check before you commit.
  7. git commit -m "concise description of your changes" to commit your changes.
  8. git status again to confirm that your “working directory is clean”.
  9. Make some changes to your code and start again at Step 4. You should be committing your changes multiple times a day.

When you are ready, enter the last two lines of the GitHub instructions to attach and sync your local and remote repos.

$ git remote add origin https://github.com/acidtone/hello-world.git

$ git push -u origin master

There are two ways to connect your system to GitHub: HTTPS and SSH. It’s widely recommended that beginners use the HTTPS option. You will be asked for your login credentials the first time but your terminal should remember this for future connections.

**Activity: Project work**[**Permalink**](https://cprg210.github.io/schedule/day-05-git-github/#activity-project-work)