

Software: Git and Github

This guide provides instructions for setting up **Git** and **GitHub Desktop**.

Version control is essential for managing code changes and collaborating with others. For the **CEE6501** course, I will be uploading example files, lecture materials, and interactive Jupyter notebooks to a GitHub repository.

Repo Link: <https://github.com/Bruun-Automation-Research-Lab/CEE6501>

The lectures are designed to be hands-on, and you'll be expected to follow along using the provided notebooks. To do this effectively, you'll need a basic working understanding of the Git workflow we will be using in this course.

Learning Objectives

By the end of this guide, you will:

- Installing Git and Github Desktop
- Making a GitHub Account
- Cloning the course repository
- Pulling updates as new content is added



Set Up Git



Windows

- Visit the [Git download page](https://git-scm.com/downloads): `https://git-scm.com/downloads`
 - Download the `.exe` installer and run it
 - Follow the on-screen instructions, accepting default options
 - **Important:** Select "*Git from the command line and also from 3rd-party software*" during setup
-



macOS

Open the Terminal and run the following:

```
# Install Homebrew if not already installed  
/bin/bash -c "$(curl -fsSL  
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"  
  
# Install Git  
brew install git  
  
# Verify Git is installed  
git --version
```



Set Up GitHub

Make a Github Account

1. Go to <https://github.com> : <https://github.com>
2. Click **Sign up** and follow the steps to create a free account
3. You can use any email, but if you use your GT email as you can later apply for a few educational upgrade if you choose to
4. Once your account is created, verify your email address

Access Course Repo

Find course repo: <https://github.com/Bruun-Automation-Research-Lab/CEE6501>

The screenshot shows the GitHub repository page for 'Bruun-Automation-Research-Lab / CEE6501'. The repository is public, as indicated by the red underline over the 'Public' link in the header. The main interface includes a navigation bar with links to Platform, Solutions, Resources, Open Source, Enterprise, and Pricing. On the right side of the header are search, sign in, sign up, and a profile icon. Below the header, there are buttons for Notifications (0), Fork (0), and Star (0). The repository name 'Bruun-Automation-Research-Lab / CEE6501' is displayed in blue. The repository has 1 branch and 0 tags. A 'Code' button is highlighted in green. Below the repository name, a commit list is shown, starting with a commit from user 'ebruun' adding a bat file for slide gen. The commit details are: 'added bat file for slide gen' by 'ebruun' at '91acf1a · 1 hour ago' with '6 Commits'. The commit list also includes additions to .vscode, Assignments, Lectures/L1, .flake8, .gitignore, .markdownlint.yaml, .pre-commit-config.yaml, and .prettierrc.yaml. To the right of the repository details, there are sections for 'About', 'Releases', and 'Packages'. The 'About' section notes 'No description, website, or topics provided.' The 'Releases' section says 'No releases published'. The 'Packages' section is currently empty.

Platform Solutions Resources Open Source Enterprise Pricing

Sign in Sign up

Bruun-Automation-Research-Lab / CEE6501 Public

Notifications Fork 0 Star 0

Code Issues Pull requests Actions Projects Security Insights

main 1 Branch 0 Tags Go to file Code

ebruun added bat file for slide gen 91acf1a · 1 hour ago 6 Commits

.vscode added bat file for slide gen 1 hour ago

Assignments added extra files 3 hours ago

Lectures/L1 added bat file for slide gen 1 hour ago

.flake8 added extra files 3 hours ago

.gitignore add environment filem initialize Readme 3 days ago

.markdownlint.yaml added extra files 3 hours ago

.pre-commit-config.yaml added extra files 3 hours ago

.prettierrc.yaml added extra files 3 hours ago

About

No description, website, or topics provided.

Readme

Activity

Custom properties

0 stars

0 watching

0 forks

Report repository

Releases

No releases published

Packages



Set Up Github Desktop

Install GitHub Desktop

- Visit the [GitHub Desktop download page](https://desktop.github.com) : `https://desktop.github.com`
- Download the installer for your OS
- Install it like any other application:
 - **Windows:** `.exe` file
 - **macOS:** `.dmg` file

Configure GitHub Desktop

1. **Sign in to GitHub**

Launch GitHub Desktop and sign in with your GitHub account.

2. **Set Integrations**

Choose your preferred editor (e.g., VS Code) and shell (e.g., Git Bash or Terminal).

3. **Git Configuration**

Ensure your username and email are set correctly under *File > Options > Git*. These are stored globally in `.gitconfig`.

Clone a Repository to your Local Machine

1. Go to **File > Clone Repository**
2. Under the **URL** tab, select the course repo URL

```
https://github.com/Bruun-Automation-Research-Lab/CEE6501
```

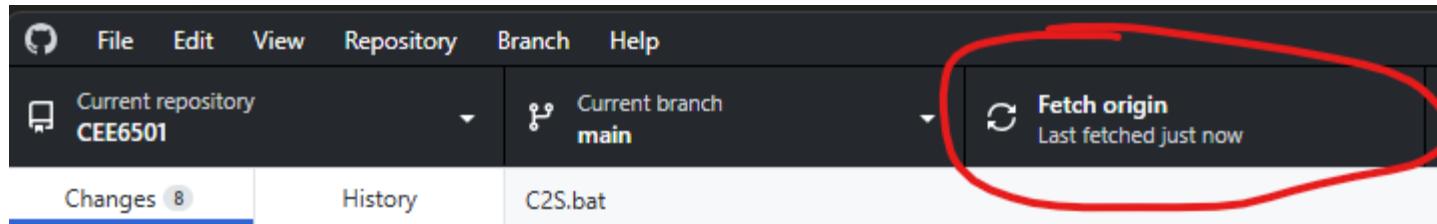
3. Select where you want the repo to be saved on your local machine. I suggest making a folder called Github under Documents and saving all repos there.

Note: You only need to do this once per repository.



Fetching and Pulling Updates to the Repo

1. Click **Fetch** to check if any changes have been made to the remote repository
2. If changes exist, click **Pull** to download the latest updates into your local copy



Note: You should ****fetch**** and ****pull**** every time you start working to make sure you have the latest files pulled to your local folder.

3. **Push** is used to upload your own changes — but for this course, updates will be **one-way only**

- You will **not** be contributing to the shared repo
- It is used solely to distribute updated course materials and lecture notebooks
- **⚠️ IMPORTANT:** If you make changes to any of the Jupyter notebooks (e.g., adding notes, solving exercises),
make sure to **save your edited copies in a different folder**
— any changes made in the cloned repo folder will be overwritten
when you pull updates



What's Next

Now that you have Git and GitHub set up and can clone the course repository, you're ready to focus on running and modifying code locally. In the next tutorial, we'll install Python and Conda, which will give you a reliable and reproducible environment for executing the course notebooks and scripts.