Brody McLaughlin

09/09/2023

Deb Harding

CS 3300 002

Technical Document

Purpose:

The purpose of this technical document is to give more information about the git and python we will use for GE01. There will be resources on installing python, as well as a command terminal to use for git. There is also a guide to set up git on the terminal, as well as commands that will be very useful when navigating git. There is an organized list of commands at the end of the page that briefly explains each command. The purpose of the assignment was to get familiar with how the basic git commands work and to begin to speculate how future guided explorations will go.

**Resources:**

Git/Github and Python

Start exploring git, github, command line, python with virtual environment.

[2 Python and IDE](https://docs.google.com/document/d/1GXnP1p7t8o0jhHuwP_EQaP9AdZdINGBZ/edit#heading=h.7a4jn11vv6wq)

[Install Python](https://docs.google.com/document/d/1GXnP1p7t8o0jhHuwP_EQaP9AdZdINGBZ/edit#heading=h.79csvznoivco)

[Install IDE VS Code](https://docs.google.com/document/d/1GXnP1p7t8o0jhHuwP_EQaP9AdZdINGBZ/edit#heading=h.9gomil77gszl)

[2 Set-up git and github repository](https://docs.google.com/document/d/1GXnP1p7t8o0jhHuwP_EQaP9AdZdINGBZ/edit#heading=h.3fp0cqgnykx1)

[2 Add, Commit, Push Practice](https://docs.google.com/document/d/1GXnP1p7t8o0jhHuwP_EQaP9AdZdINGBZ/edit#heading=h.27n2hu32nsae)

[3 Branching](https://docs.google.com/document/d/1GXnP1p7t8o0jhHuwP_EQaP9AdZdINGBZ/edit#heading=h.tyjcwt)

**1 Python and IDE**

**Install Python**

1. Open the command window and check your python version to see if you have it installed.
2. Install python version 3.11 [Download Python](https://www.python.org/downloads/).
   1. If on windows and have older version of python you  should uninstall first : [How to Uninstall Python](https://www.pythoncentral.io/how-to-uninstall-python/)
3. We will be using a virtual environment [Python venv: How To Create, Activate, Deactivate, And Delete](https://python.land/virtual-environments/virtualenv)

**Install IDE VS Code**

You can use a different IDE but this is what I will be using and supporting. This has a nice tool to view the sqlite database for your apps.

<https://code.visualstudio.com/download>

Open the project in Visual Studio Code: In Visual Studio Code, go to `File -> Open Folder` and select the folder that contains your Django project (the folder named "myproject" in this case).

1. Configure the Python interpreter: In Visual Studio Code, open the Command Palette by pressing `Ctrl+Shift+P` (Windows/Linux) or `Cmd+Shift+P` (Mac). Search for "Python: Select Interpreter" and choose the Python interpreter associated with your virtual environment (e.g., `myenv`).

A screenshot of a computer program

Description automatically generated

1. Install the Django extension developed by Baptiste Darthenay: In Visual Studio Code, go to the Extensions view and search for the "Django" extension. Install it to benefit from Django-specific features and enhancements for what we will be doing later.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. You can use this to edit your python file for practice.

|  |
| --- |
| Take a screenshot of the ide you have set up and the python file from the repository once you edit it below. |

**2 Version Control**

Set-up git and github repository

Use the command line tool of your preference in your environment. I ended up using command prompt on my windows but also have used windows powershell.I use the generic command tool on my mac. I ran both this summer.

Here is an example of using the default command prompt

A screen shot of a computer

Description automatically generated

Create a github account if you do not have one

1. Create a github account if you do not have one.
2. Create a github repository from the web US
   1. [Github’s Instructions for Creating a Repository](https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-new-repository)
   2. Call it cs3300-version-practice
   3. Include “readme” file
   4. For now, you can leave the “*.gitignore*” template set to “none”.. (In the GE01 you will add a python gitignore file).
   5. Make it public
3. Clone the repository to your local computer using a command line tool
   1. [Cloning a repository - GitHub Docs](https://docs.github.com/en/repositories/creating-and-managing-repositories/cloning-a-repository)
4. You may need to generate an SSH Key pair to configure remote access to your repositories. Github’s instructions for this process can be found [here](https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent).
5. See [git Documentation](https://git-scm.com/docs/git)

Add, Commit, Push Practice

1. You can just work with updating the readme file to explore the following or with a python file.
2. Check the git branch and status

git branch

git status

1. Before you can commit the version you must add the new file to the index (the staging area)

git add .

git status

1. Record changes to the local repository with a description but first you might need to  include the author identity. Then check the status

git config --global user.email "you@email" (email associated with repository)

   git config --global user.name "Your Name"

git commit -m ‘new bash for introductions’

git status

1. You will add your code, commit and push. Then explore the repository on the remote server, github

     git push

git status

A screenshot of a computer

Description automatically generated

Branching

1. From the command line in your repository on your computer check the log and what branch you are on.

git log

git branch

1. Create a branch called Iteration01 and check the log and branch

git branch 'iteration01'

git log

git branch

1. Switch to Iteration01 branch to check out code:

git checkout 'iteration01'

git branch

git status

1. Modify readme file

git status

1. Add the file to the staging area

git add .

git status

1. Record the new version in the local repository

git commit -m 'bug fix 1'

git status

git log

1. Share the changes with the remote repository on the new iteration01  branch. Go to your github and you will see you now have two branches. Click to view the branches. Now others working on the branch can pull your updates from the iteration01 branch.

git push --set-upstream origin iteration01

git status

git log

A screenshot of a computer

Description automatically generated

1. Update the remote main branch repository with the change from iteration01 branch. Remember to make sure you are on the correct branch. Then go to github to see the versioning.

git push

git status

git log

1. Delete the branch when it is no longer needed. Be very cautious before deleting branches.

git branch -d iteration01

git push

1. Tag the main branch ‘v1.0’ then view the tag and push to remote repository. When you go to the remote repository you should see the tag listed.

git tag -a v1.0 -m 'Release 1.0'

git tag

git push origin v1.0

A screenshot of a computer

Description automatically generated

Important Commands

* **git clone – clone creates a new repository using an existing repository**
* **git add – adds a file to the index to be used with git**
* **git commit – commits changes to the repository such as pulls and pushes**
* **git status – lists the status of the environment as well as files to be committed.**
* **git log – lists the version history for the branch, including changes**
* **git tag** – gives a tag to a commit
* **git branch – lists all branches in the working repository**
* **git checkout – used to switch to a branch, or create a new one**
* **git merge – used to merge a branches history onto another**
* **git push – this sends the commits from a branch to the repository**
* **git pull – this command merges changes from the remote server**