<https://cognizant.udemy.com/course/github-ultimate/>

.git folder....holds the actual repository. – created with init command

**git init demo** - creates a directory in the demo folder with a .git file

Local Git States

* Working Directory
* Staging Area
* Repository (.git folder)

**git status**

**git add filename ... will move a file to staging**

**git commit <filename> # opened the file for editing then waited for me to close**

**git log ...shows all the changes made to the repository**

**git show .....show the last change**

**git ls-files .... shows what files are being tracked**

**git commit –am “adds file and commit changes”**

**git help log .....pulls up the help menu**

**git log --oneline --graph --decorate --all**

**touch example.txt ...creates a new file called example.txt linux command**

**git mv old\_file\_name new\_file\_name ...renames a file**

**git rm filename ...removes a file after commit is done**

**git add –u ....for updating a file**

**git add –A ..stages all changes** <https://stackoverflow.com/questions/572549/difference-between-git-add-a-and-git-add>

**gitignore ...a file that show all the filenames to be ignored using wildcards.**

**git diff**

**start here branching and merging**

**git help....output**

**C:\Program Files\Git\mingw64\share\doc\git-doc more help files here**

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)

clone Clone a repository into a new directory

init Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)

add Add file contents to the index

mv Move or rename a file, a directory, or a symlink

reset Reset current HEAD to the specified state

rm Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)

bisect Use binary search to find the commit that introduced a bug

grep Print lines matching a pattern

log Show commit logs

show Show various types of objects

status Show the working tree status

grow, mark and tweak your common history

branch List, create, or delete branches

checkout Switch branches or restore working tree files

commit Record changes to the repository

diff Show changes between commits, commit and working tree, etc

merge Join two or more development histories together

rebase Reapply commits on top of another base tip

tag Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)

fetch Download objects and refs from another repository

pull Fetch from and integrate with another repository or a local branch

push Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some concept guides. See 'git help <command>' or 'git help <concept>'

to read about a specific subcommand or concept.

**start here branching and merging**

[**https://cognizant.udemy.com/course/github-ultimate/learn/lecture/4731798#overview**](https://cognizant.udemy.com/course/github-ultimate/learn/lecture/4731798#overview)

**branching**

**git branch ...shows the current branch**

**HEAD ...points to the last commit in the branch**

**git checkout –b branchname ...create ‘branchname’ and move to that branch**

**git merge <branchname> ..must be in an upper branch**

**types of merges: fastforward ,**

**git branch –d <branchname> ....delete ‘branchname’**

**cat <filename> ..show the contents of a file on the command line**

**git tag <tagname> ...creates a tag for a milestone in the repository**

**git tag –list ...show the tags you created**

/c/Users/Chris/AppData/Local/atom/atom.exe ...used to open atom in git bash

**git reset <idnumber> <--soft | --hard> ...go back to a previous state**

**git reflog .....shows changes in a branch**

**video 37 remote repository**

<https://cognizant.udemy.com/course/github-ultimate/learn/lecture/4731828#overview>

**used git hub to create a new repository --- gitprojectsdemo**

<https://cognizant.udemy.com/course/github-ultimate/learn/lecture/4731842#overview>

linking to github repository

git remote –v ....shows our remote connections

git remote add origin git@github.com:adamchr9024/gitprojectsdemo.git

**name url**

**git remote –v ...will show remote connections**

**video 43 pushing changes to git**

**git push –u origin master --tags (only use – u on the initial push)**

**-u use to create a tracker for changed to local and remote repository**

**origin ---is name of the local repository**

**master ---branch name**

**Section 8 GitHub Repository edit, delete, create, view ....crud**

$ git clone https://github.com/adamchr9024/mywebsite.git

<http://www.initializr.com/>

$ git push origin master

video 54 fetch vs pull

pull does a fetch and merge which may destroy you changes to local repository

fetch does not merge the data

ALWAYS DO A PULL OR FETCH BEFORE PUSHING IF YOU ARE SHARING A REPOSITORY WITH OTHERS

$ git fetch

remote: Enumerating objects: 5, done.

remote: Counting objects: 100% (5/5), done.

remote: Compressing objects: 100% (3/3), done.

remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0

Unpacking objects: 100% (3/3), done.

From https://github.com/adamchr9024/mywebsite

47d74d0..b37c436 master -> origin/master

$ git status

On branch master

Your branch is behind 'origin/master' by 1 commit, and can be fast-forwarded.

(use "git pull" to update your local branch)

Changes not staged for commit:

(use "git add <file>..." to update what will be committed)

(use "git checkout -- <file>..." to discard changes in working directory)

modified: css/main.css

no changes added to commit (use "git add" and/or "git commit -a")

stop on video 55