159.251 - Version Control with Git

Installation and Tutorials

Installing Git

- Git for your OS (Available for Linux, Windows, Mac OS and Solaris)

  https://git-scm.com/downloads

- Detailed installation guide is provided here:

  https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

---

Git - quickstart

Getting started

Git is very low impact to get started, as you can work initially solely in your work directory. Just open Command Prompt or Git Bash (Git for Windows provides a Bash emulation used to run Git from the command line) and run your commands.

- After installing git (as above) –check that git is installed correctly:

```
git
```

- configure your Git- tell Git who you are (important- used to track commits):

```
git config --global user.name "ABC XYZ"

git config --global user.email ABC@home.com
```

- To initiate a repository (turn a folder into a repository):

```
git init
```

- this creates the *.git* folder but doesn't affect anything else

Then just carry on working as usual, but remember to commit when you have completed a "feature", or just before you start making any major changes. To commit, do both of:

First, add files

```
git add Class.py                                    #to add a file

git add .                                           #to add ALL files in the folder
```

Then commit that to your local repository

```
git commit -am "Your commit message"                #commit –and add a message
```

You can also create a new branch using the following command

```
git branch <branchName>
```

Make sure that make the current branch is the working branch

```
git checkout <branchName>
```

And then you can merge the branch later (make sure that you)

```
git merge <branchName>                                  #merging branchName with 'master'
```

A branch be deleted using the following command

```
git branch d- <branchName>
```

Finally, you can use Git clean to remove untracked files from the working tree

```
git clean -n   # dry run of git clean - show you which files are going to be removed without actually perform the clean.
```

```
git clean -f   # to remove all untracked files from the current working directory.
```

For more information about ***Git clean***, read this: https://www.atlassian.com/git/tutorials/undoing-changes/git-clean

Other useful commands

Display help	git help
To see what's changed	git status
To see what's changed since last commit	git diff
To see a side-by-side view of changes	git difftool

Note: When viewing the git log, or git status, press **q** to quit viewing, **space** for next page, **enter** for next line.

Git tutorials

Download the zip file from Stream and then unzip the file in one of your home selected directories. Do the following tasks and show it to us:

1. Create (initiate) a new Git repository of the same folder ,
2. Add and commit only **.py** files (add a meaningful message when committing the files)
3. Create a new branch (FirstBranch)
4. Add and commit only **.java** files
5. Create a new branch (SecondBranch)

6. Add and commit ALL other files
7. Merge both branches with the *master* branch (in the same line! Use only one command)
8. Delete both new branches
9. Show the your branches using

```
Git log   # to display the entire commit history
git show-branch –all
git ls-tree -r master --name-only   #to list all files on master branch
```

10. Host your repository in a remote site, and push all commits to the remote repository. You can use any Git hosting sites such as BitBucket, GitHub, or GitLab. You will need to setup an account before you create a repository.

    Some important command that you will need

```
git remote add origin <URL site>  # The hosting site will generate a unique URL
for your newly created repository. URL example: https://XYZ@bitbucket.org/XYZ/test.git
```

```
git push -u origin --all  # pushes up the local repository for the first time
```

```
git push origin --tags    # pushes up any tags
```

11. Download and use a Git desktop client
    a. Import your remote repository
    b. Repeat steps 1,2,4,5 and 6 (for the single branch only) using the GUI interface (much easier than the command line!!).

Git tutorials

- The second edition of the “Pro Git” Book is available for free from https://progit.org/
- One of the best available tutorial is "Git in one hour" - a video tutorial http://www.youtube.com/watch?v=OFkgSjRnay4 (awful audio but great tutorial!)
- http://www.ralfebert.de/tutorials/git/

- http://gitimmersion.com/index.html

Git Readings and FAQ

Development with version control systems

Git Workflow: Clarification and Avoiding Antipatterns

A discussion about the ways of using the different branches, with some suggestions, and things to avoid: http://stackoverflow.com/questions/6905557/git-workflow-clarification-and-avoiding-antipatterns

Git, Binary Files and Cherry Picking Patches

How does git decide which files can be merged and which can't? This post explains that, and indicates how you can flag some filetypes as "non-mergable" even though they're text (e.g. XML):

http://www.bluishcoder.co.nz/2007/09/git-binary-files-and-cherry-picking.html

Features are small

*Old-school branch-per-feature meant that branches were large and long living to avoid having to integrate because it was a pain. This was a vicious circle as the feature would diverge further and further from other features or the mainline. Features should be as atomic as possible and your development process should abide by the Open Close Principle. Features should be small.*https://plus.google.com/109096274754593704906/posts/R4qkeyRadLR

Interactive web visualisation of what git's doing, with commands

http://ndpsoftware.com/git-cheatsheet.html

How do you roll back (reset) a git repository to a particular commit?

A slightly less scary way to do this than the git reset –hard method is to create a new branch. Let's assume that you're on the master branch and the commit you want to go back to is c2e7af2b51. http://stackoverflow.com/questions/1616957/how-do-you-roll-back-reset-a-git-repository

git repos on a server using non-standard SSH ports

If you're working with a remote server that has SSH on a non-standard ported (e.g. NOT port 22), these tips will be useful:
readystate4.com/2011/03/30/working-with-git-repos-on-non-standard-ports

How do I clone all remote branches with Git?

http://stackoverflow.com/questions/67699/how-do-i-clone-all-remote-branches-with-git