

Getting Git

1

Repo basics

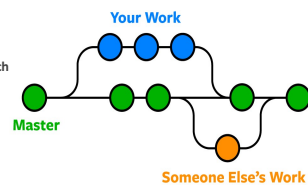
A repo or repository houses files for a project
You can use GitHub as a GUI or a terminal window

git init	will designate a folder/ directory as a repo
git status	checks status of files in the repo
git add	add the file to git
git commit -m "message"	commits the file as a git file
git push	send updates to GitHub
git pull	gets local repo up to date
git checkout	gets current version of a branch or file

2

Branches

The primary branch is master (by default)
master is presumed to be the deployable branch
You can "try out" other ideas / work on
code without affecting master by creating a **branch**



Source: <https://www.nodedesktop.com/learn/git/git-branches>

3

Branches

You may make changes to files in the branch or add files or delete files.

Committing changes to the branch does not affect the master branch

git branch	list branches or create new branch
git switch <branch>	switches to the new branch
git push origin <branch>	pushes changes from branch

4

Merging Back to Master

When a branch is tested and ready to move to production, it can be **merged** back into the master branch

Any conflicts will need to be reviewed in order to merge correctly

git merge <branch> merges changes from branch into master (must be in master branch)

git branch -d <branch> delete a branch

5

Adding Collaborators

Go to "settings" for the repo

Choose "manage access"

Click the "invite a collaborator" button

6

Try It!

Go into GitHub and create a repo for your group.

Create a file called group members.txt and put the group name in it

Each member of the group

- create a local version of the repo
- edit the file to add their name – and commit any changes
- Do a "pull" from the master file to be able to see all group members in the file

7