GitHub Cheat Sheet

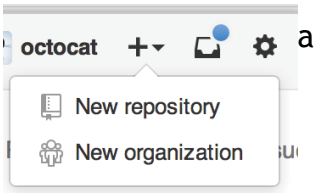
GETTING STARTED

Downloading:

GitHub is the software used to record changes in your code.

Windows: https://windows.github.com
Mac: https://mac.github.com

Creating a new Repository:



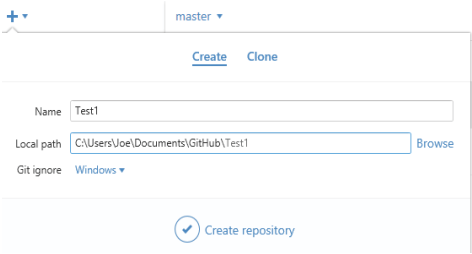
A repository is a folder that stores all of your code for a project.

Go to github.com and click the "+" to create a new repository.

-Add a ".gitignore" based on the language you're programming in
-Add a License

Pro Tip: The MIT License is a good default.

Adding Repositories to the GitHub GUI:

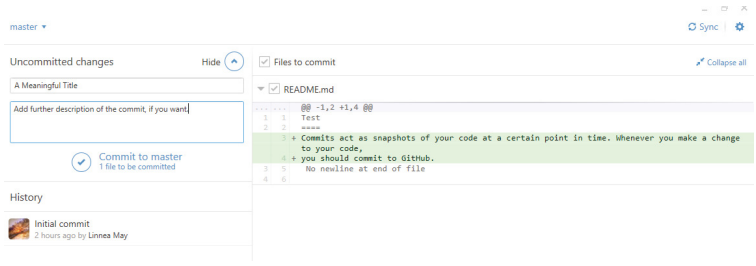


In the GitHub GUI, make a copy of the repository on your computer by clicking "clone".

Pro Tip: Add your repository to the default directory called "GitHub" (usually located in the Documents folder.)

COMMITTING

Each version of your code is stored as a commit in GitHub. Any time you make a change, you should commit to GitHub.



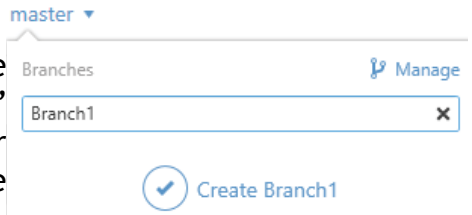
When committing, add a short but meaningful description of the changes you made.

Syncing:

After every commit, you should sync so that the updates can be accessed from other computers.

Pro Tip: ALWAYS sync when things are going well, and NEVER if experiencing issues.

BRANCHES



Branches are different "folders" within your repository- each one stores specific code. Your default is the "Master" branch.

Go to the branch selector in GitHub to create a new branch.

Pro Tip: An example of different branches includes one for Teleop, and another for Autonomous.

Multiple Branches:

Different files will live in different branches, depedning on use. When working with many branches:

-ALWAYS commit before switching branches
-Do NOT try to access files that aren't located the branch you're currently in.
-NEVER have more that one person working in the same branch at once.

Merging Branches:

Whenever you make significant progress within a certain branch, you will want to "merge" it to Master, where all of your most current working code will be kept.

In the branch selector, click "Manage" to merge the two branches together.

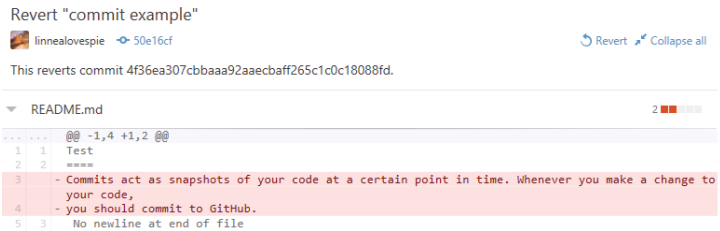
GitHub Cheat Sheet

DISASTER RECOVERY

Unfortunately, there will always be a time where issues arise.

Revert:

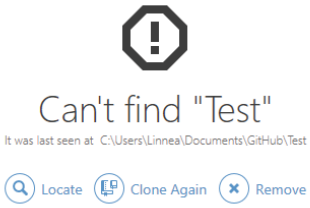
If something goes wrong when committing and one needs to undo the immediate changes, revert back to the previous commit.



Pro Tip: The Git Shell can be used to "roll back" to even older commits.

If Things Get REALLY Bad:

If reverting the commit won't solve the problem, delete the local repository in the directory and reclone it using the GitHub GUI.

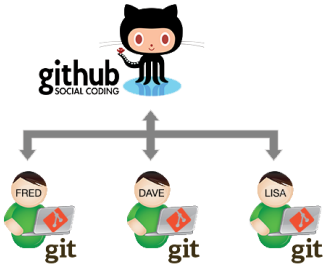


Common Destructive Scenarios:

-"*Backing up*" syncs, when one person forgets to sync their commits and another tries to sync changes from a separate computer

-"*Merge conflicts,*" when one file is edited in two different branches.

GIT VS GITHUB



GitHub is the website where one can interact with a repository. Git is the actual software that stores versions of code, and is what is used on one's computer when committing.

USING THE GIT SHELL

While using the GUI has its benefits, the shell can provide advanced options in manipulating the repository. To change the default shell, go to Settings > Options.

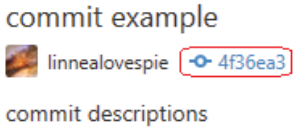
Pro Tip: GitBash is recommended due to its compatibility and ease of use.

BASIC GIT COMMANDS

These are a couple git commands to get you started with using the shell.

$ cd
Change your current directory to the folder with your repository.
$ git status
Show the status of your local repository- if you need to commit, and if it needs to be pushed to the server.
$ git log
Shows the previous commits in the branch.
$ git checkout <branch>
Switch to specified branch.
$ git pull --rebase
Gets updates from GitHub.
$ git commit -a
Opens a text file to log commit messages
$ git push origin <branch>
The Git equivalent of "sync", sends your local updates to the server.
$ git revert --no-commit <head>
Revert to a previous commit, specified by the commit's "head."

Pro Tip: An commit's heading will look something like "0d1d7fc32" in the GUI.



For more on using Git and GitHub, visit: teamquadx.org or github.com
Questions? Email quadx6299@gmail.com

Team 6299 QUADX
Austin, Texas
GitHub: FTC6299