# Getting Started with Git

This is a tutorial designed to get KBotics students familiar with working with Git (command line). However, it can be useful for anyone trying to familiarize themselves with this beautiful piece of wizardry.

**You will need a basic understanding of the terminology of git in order to complete this tutorial.**

The following steps should be followed IN ORDER to ensure everything runs smoothly with the project.

1. **Fork the github.com/KBotics repository you want.** This should only be done once.
2. **Download Git 1.8.4.** Version 1.8.5 does not have a stable build for Windows (despite their obvious LIES on the website). If you download 1.8.5, YOU WILL NOT BE ABLE TO PUSH TO YOUR REPOSITORY. Please make sure you download 1.8.4.
3. **Clone your fork.** This can be done with the command:  
   *git clone https://github.com/(your\_user\_name)/(repository\_name)*. It should automatically download the contents into a folder named *repository\_name* in your current directory.
4. **Add the KBotics repository to your remotes.** When you initially clone your fork, you’ll have a remote called *origin*. This origin pulls and pushes to YOUR FORK. The pushing is important, but we want to pull form the KBOTICS remote. Type in  
   *git remote add upstream https://github.com/KBotics/(repository\_name)*.
5. **Pull from the KBotics repository.** *git pull upstream master* updates your fork to the latest version of the robot code. YOU NEED TO RUN THIS EVERY TIME YOU LOG IN. You may also need to run this throughout the day, as we might make changes to the main branch that you are unaware of.
6. **Push to your fork.** You don’t need to do this if the previous command returned “already up-to-date” (or something similar).
7. **Write code.** Make all your changes. TEST THEM. If they work, continue on to step 8.
8. **Add the files you’ve changed.** *git add(files\_that\_you’ve\_changed).* The main thing here is to not include ALL the files that are changed, as sometimes things in the *nbprojects* folder will be modified automatically. If you want to add an entire folder – like src – you can write *git add (folder\_name)* and it will add everything within the folder that has been changed.
9. **Commit your changes.** *git commit –m ”(your\_message)”* will commit without opening the vim terminal. IF YOU ACCIDENTALLY OPENED THE VIM TERMINAL: the letter I will allow you to start typing. Type in your commit message under the lowest # symbol, then press ESC. Typing “:w” will save the changes you’ve made, and “:q” will exit.
10. **Push your changes to your fork.** Generally you only have to type *git push*, but you can specify by saying *git push origin master*.
11. **Ask KBotics for a pull request.** In your fork on Github.com, there will be a green square that says “Compare and Review” when you hover over it. Click it, review your changes, then press “Pull Request”. GIVE A VERY DETAILED EXPLANATION OF WHAT YOU CHANGED AND WHY. Then press done.

That’s it! You’ve managed to make some changes and ask us to merge those changes with the final code! If you want to keep coding, just follow this tutorial again from step 7. Otherwise, feel free to turn off git and start at step 2 tomorrow!