How to: GitHub

**Before we begin:**

This is a scary looking tutorial! But once you are set up with Git/Github, it really all boils down to 5 or 6 commands. GitHub is massively useful for collaborating on projects, and is a highly sought skill in industry.



# Step one:

Create an account on [GitHub](https://github.com/) (<https://github.com/>)

Note: Choose the free account

# Step two:

Go to your desktop and create a folder (Right click>New>Folder). Name the folder ‘MyRepo’. A folder is a ‘repo’ or ‘repository’ on GitHub. It’s essentially the folder that holds an entire coding project.

# Step three:

Create a text file in that folder. (Right click>New>Text document). Name the document ‘myfile.txt’ and add a line of text to it.

# Step four:

Now that you have a project set up in a repo, let’s Git to it! Right click on the project folder and select ‘Git bash here’. This will open Git’s command window, and you can start telling Git what to do.

# http://ericsteinborn.com/github-for-cats/img/ironcat.png

# Step five:

Our first step is to tell Git that this is a repo. Type ‘git init’ into the command window and hit return. This tells git that the folder you are currently in is a repo, and is the first command you should learn

All commands start with the word git! Easy enough to remember!

# Step six:

Now we tell Git to add files to its memory for tracking. Git only tracks what we tell it to. Type ‘git add \*’ to add ALL files in the folder, or ‘git add myfile.txt’ to add specific files.

# Step seven:

Now we tell Git that we are happy with this set of files as being the current version of the project. We are committing to this being one of our project versions, so we use the commit command!

We also need to name our commit. You can name it ‘version 1.0’ or ‘first commit’ or whatever you want, but it is best to be descriptive and say something about the version (as you may need to find it later!) So type **(IN FULL!)** “git commit –m ‘version name’”

**Git will give you a warning about email addresses etc, ignore it!**

**This shouldn’t happen at home.**

# Step eight:

Now we need to link our Git project to our GitHub account. GitHub is known as a ‘remote’ (an external place to store the code). We need to let Git know about GitHub and to do this we need to ‘add a remote’. However, before we do this we must set up some space on GitHub to store our new project!

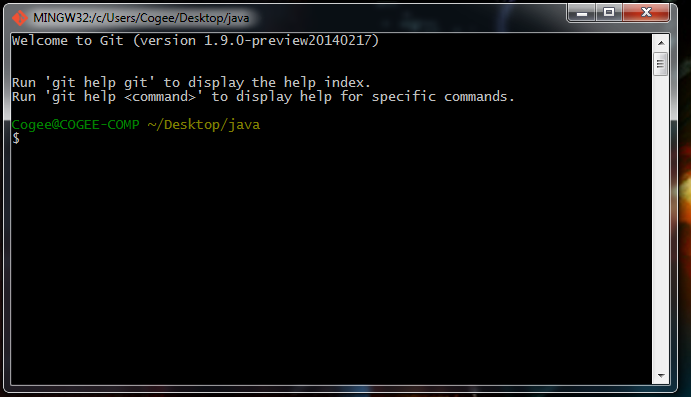
Start by clicking the ‘new repository’ button on GitHub. Then all you have to do is give it a name and hit create repository; nothing else needs to be altered.



Take this code from the next screen and paste it into the command window\*

**YOU CANNOT CTRL+V OR RIGHT CLICK AND PASTE IN GIT BASH!**

**Click here instead and go to edit>paste**

****

# Step 9:

Now we push our code up to the cloud, using the push command. ‘git push –u origin master’ \*\*

# http://ericsteinborn.com/github-for-cats/img/socialite.png

# Step 10:

Go to your newly furnished repo on GitHub, then into the settings, and add the person next to you as a collaborator!

# Step 11:

Now that you know how to create a project and push the code to the cloud, you need to know how to pull code back down. You can pull code down from any public repository using this simple command: ‘git pull https://github.com/username/repoName.git’

Click on your friend’s repo and find out his username and repo name, and use these to pull down his code.

**HINT: There is a clone URL down the bottom right of a repo’s main page**

# Step 12:

Now that you have pulled down the code, make a change to it, use git add to track the new changes, make a new git commit, add your friend’s GitHub repo as a remote and push your changes up!

* Git add \*
* Git commit –m “whatever message”
* Git remote add origin https://github.com/username/repoName.git
* Git push –u origin master

**To avoid conflicts make sure to use git pull if you think your partner has made new changes and you have an old version.**



\*git remote add origin https://github.com/YourUsername/YourRepo.git

This is the address of the remote space Git will look for when you push the code later

Name it origin for future use

Add a remote or external place to store this code

\*\*git push –u origin master

Basically means: push the master code (the stuff we’re working on right now), to the remote we named ‘origin’ earlier. The –u is complicated to explain, but it essentially gives the version we are pushing a tracking number of sorts!

