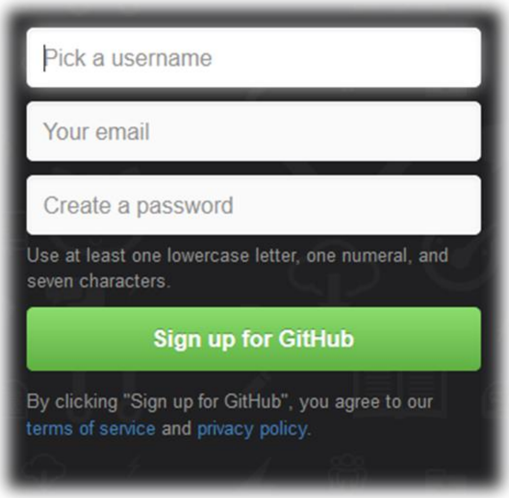
USING GITHUB

Hi there, Michael Bell here. You may remember me from such computing capers as singing ‘on top of old Smokey’ in class for no reason and creating Beltrak.

Today I am showing you how to use Github in your computing project and demonstrating what I show you here in class. It’s worth paying attention but don’t worry if you miss anything, it’s all covered here.

JOINING GITHUB


Pick a username

Your email

Create a password

Use at least one lowercase letter, one numeral, and seven characters.

Sign up for GitHub

By clicking "Sign up for GitHub", you agree to our terms of service and privacy policy.

The first thing we need to do is create an account, this is necessary to access all the features you need, to create an account go to **github.com** and fill in the boxes on the homepage that look like this:

Obviously it’s important to remember your username and password because if you forget them then you will lose access to your project and (while not an unrecoverable situation) it is somewhat annoying so why not write down your chosen details in the picture, just to be safe.

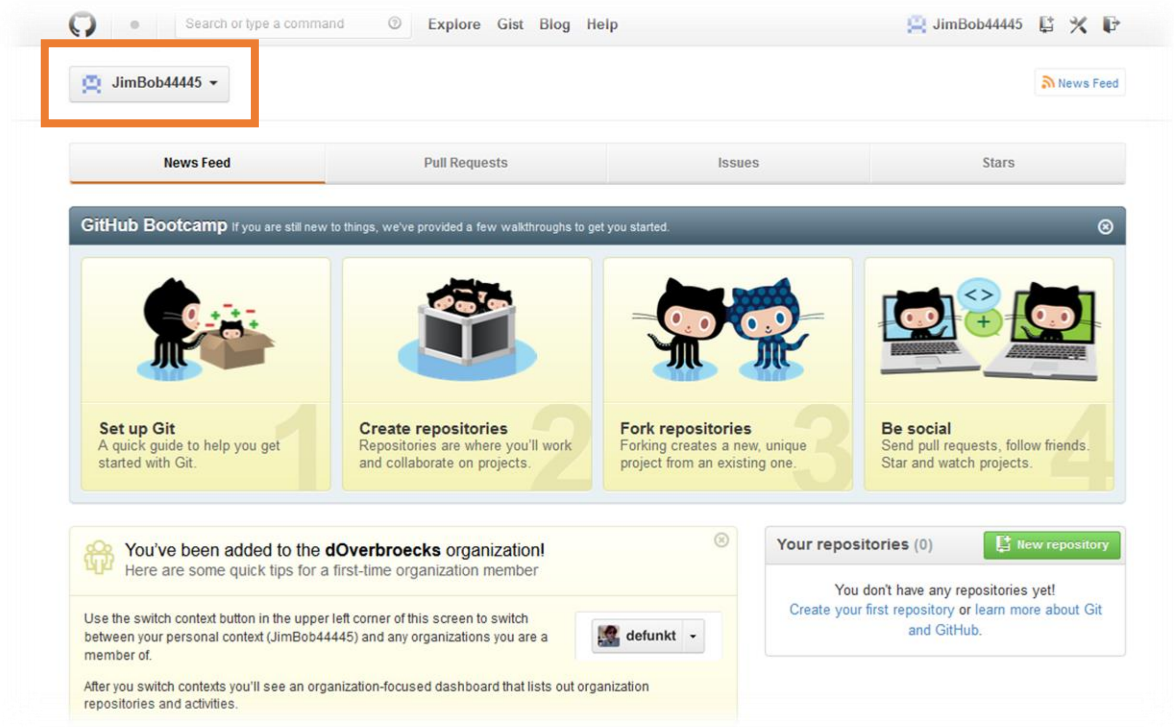
Once you click sign up you will be taken to the home screen.

It’s looking a bit barren at the moment but that’s ok, the next step is to get your account added to d’Overbroecks. This is an easy step because I do it for you. If you’re in class then I will do this during the lesson, if you have missed the lesson for some reason don’t worry, just drop me an email or Skype me (my details are at the end of the instructions) be sure to include your chosen github username and the name of your project (if you know it yet).

SETTING UP GITHUB

Once you have created an account and I have registered it, log on to github using the logon button at the top of the homepage.

When you log in you will see this page:



If you are still new to things, we've provided a few walkthroughs to get you started.

A quick guide to help you get started with Git.

Repositories are where you'll work and collaborate on projects.

Forking creates a new, unique project from an existing one.

Send pull requests, follow friends. Star and watch projects.

You've been added to the **dOverbroecks** organization!
Here are some quick tips for a first-time organization member

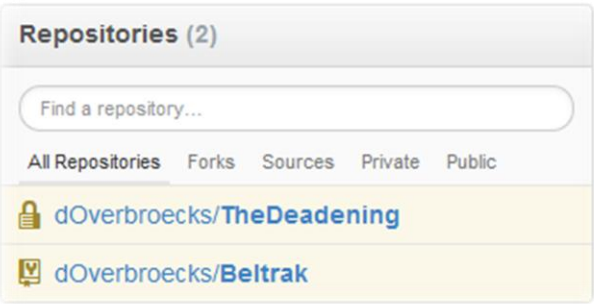
Use the switch context button in the upper left corner of this screen to switch between your personal context (JimBob44445) and any organizations you are a member of.

After you switch contexts you'll see an organization-focused dashboard that lists out organization repositories and activities.

You don't have any repositories yet!
Create your first repository or learn more about Git and GitHub.

It's somewhat misleading because it says you have no repositories (where you store your project) don't try and create one because it will give you a choice between paying or making your project public. Github have kindly given us a free account for you to store your work on but you must switch to it first, before we can do anything we must access it.

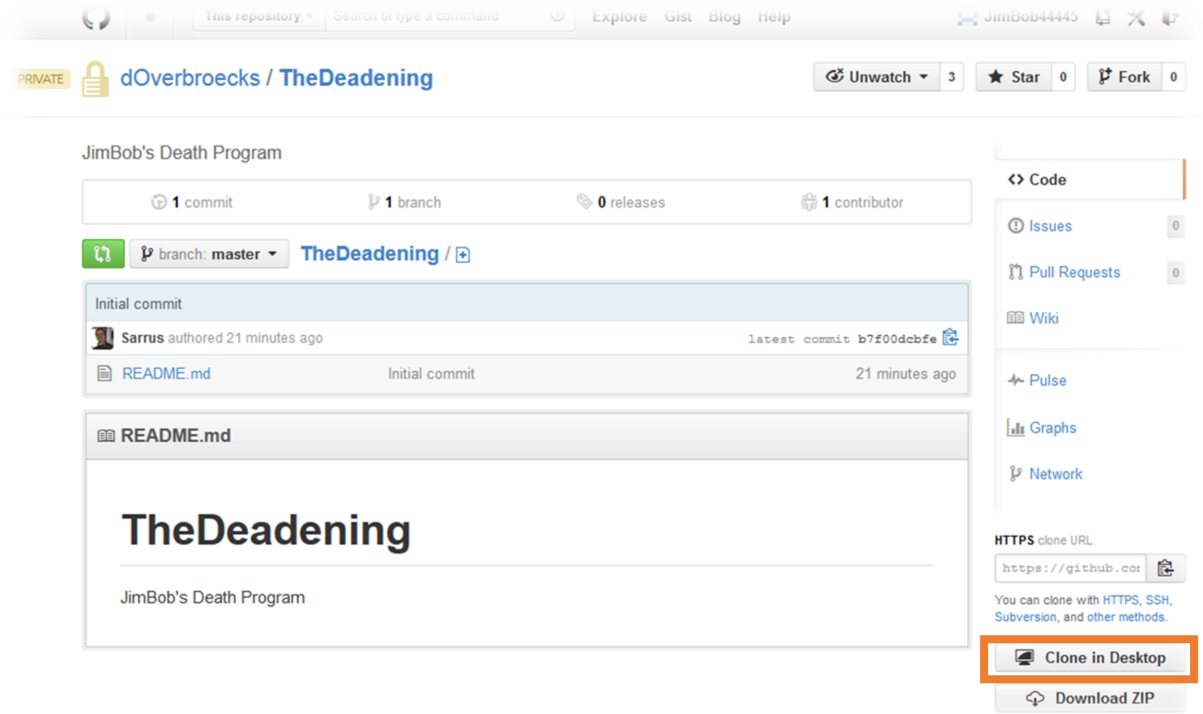
If you click on your name highlighted above you will get a dropdown list with d'Overbroecks on it, if you click d'Overbroecks you get taken to our group page, it's an explosion of fun but ignore most of it for now.



Playfully snuggled down the side is the repository's list. It has two entries in it for now, one is your project which has your chosen name next to it. (Don't be intimidated by the padlock, which just means that it isn't viewable by the general public)

Underneath that is Beltrak (my old project). I'm not planning on talking about it here but feel free to have a browse in your free time, It's a good example of how to use github with a project, just don't try to steal any of my ideas! (Because I will find you)

If you give your project a good old fashioned click you will get taken to its page, this is the meat of your project so it's worth bookmarking it (or setting it as your home page) if you get an error when you return to your bookmark (or home page) it's because you aren't signed in. this is what it looks like:

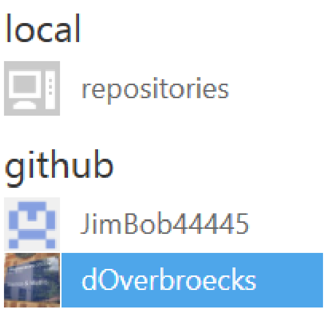


Ok, so there are a lot of buttons there which you aren't ever going to use (that said, a good student would click them anyway, just to see what they do).

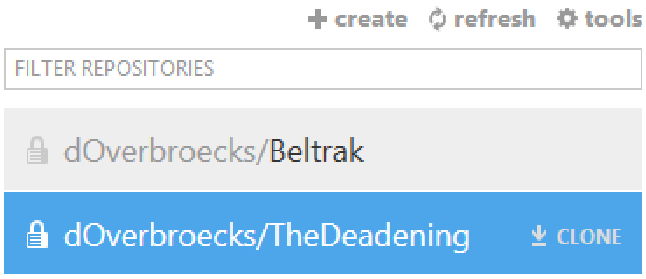
For now we will worry about the one I have highlighted in that lovely hellfire orange. Click this, and you get taken to a page getting you to install github on whichever operating system you have. Do this and once installed, open the program.

I am going to show you how to use github in windows (because it's better) but the mac version is more or less the same, it just looks different and the buttons are in different places but they do the same things.

You're probably going to be asked to log in during the installation (for which you should use the account you created earlier) but if for some reason you didn't just click the log in button and sign in.



This then shows down the side, the local repositories section will be empty because you don't have any yet, you should ignore the one with your username on it because you won't be using that, instead, click on D'Overbroecks.



This shows in the middle, it's a familiar list because you saw the same one on the d'Overbroecks homepage on Github's site.

If you hover your mouse over your project (like I have here) you will see a 'clone' button magically appear (pretty cool huh?)

Click clone and it starts doing things. To be more precise it is now copying your project off the server and putting it in a folder in your computer, this folder is important because Github will only work on things inside it so **be sure to always put your project in this folder** I will show you how to find it in a moment.

Once cloning® is complete it replaces the clone button with a picture of a computer. This means that it is now local on your computer.

You can copy Beltrak as well if you like, just to have a look but you won't be able to edit it.

You may want to work on your project on multiple computers, just install github desktop and clone your project onto the new computer. As long as you follow the steps in the next section every time you work on your project then your project will be synchronised between both computers.

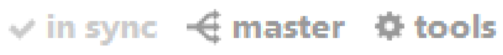
Congratulations, you're all set to start working on your project!

USING GITHUB

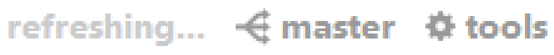
Every time you work on your project you should follow these steps, there are 3 of them and they are all important, you must do step 1 before starting work, you must do step 2 every time you make a significant change to your project (called a 'commit') and you must do step 3 when you are finished working.

STEP 1

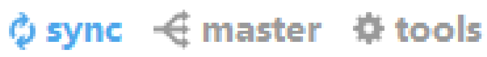
This step involves starting github and checking for changes and it's dead easy. Run github (from wherever you installed it) click on repositories (under local) then click on the arrow next to your project.



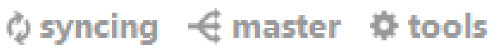
look at the top of the page if you can see a tick which says 'in sync' like it does here then you are ready to go and you can move on to step 2.



If you see the word 'refreshing…' github is checking to see if any changes have been made else ware, just wait for a few seconds and it will change.



If you get the word 'sync' on its own and highlighted in blue then you **must** click it before you start work



it will change to the word ‘syncing’ and spin a bit, it is copying all your changes onto your computer, wait for a bit and it will change to ‘in sync’ when it does you are good to go!

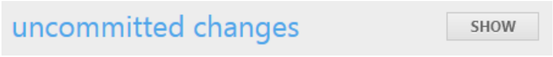
Once you are ready it helps to open the folder github has created for your project (coz dats hip) to do that just click ‘tools’ and then ‘open in explorer’ or ‘open in finder’ on a mac.

If you have already started your project or the write up, planning or anything else relating to it, you should copy it into this folder. From now on you should work from here.

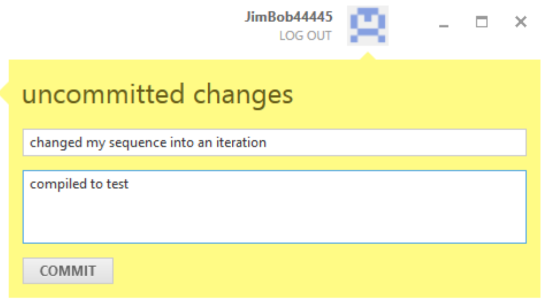
STEP 2

After completing step 1 you are now ready to start working on your project, go ahead and write some code, add to your write-up or do some planning, just be sure to save it in the folder we created. Take a look at Beltrak if you’re looking for an example of how to organise your folders.

Every time you make a significant change to your project for example, created or edited a function, added a section to your write-up or just renamed something, save your work. Changes won’t show up in the github window until you save so you need to do that. Save Save Save, I can’t say it enough.



After saving (see I said it again) github will say that there are uncommitted changes, click the show button to see them.



changed my sequence into an iteration

compiled to test

Github will show you what you have changed and give you this box, give your commit a meaningful name so you know what you did, this is important for when you do your write-up and want to go over how you made the project, in the description box below it’s a good idea to write down any testing you did, as your write-up will need to include a list of **all** the alpha testing you did, if you write it here it is easy to retrieve along with a list of all the changes you made.

Once filled in, click commit. You can now go back to work until the next milestone comes along, then just save, title and commit all over again.

STEP 3

When you are finished working, save and close your work. If you have anything uncommitted then follow step 2 again.

Once you have committed things they must be synced with the server, this is indicated by the ‘in sync’ indicator being replaced with the blue ‘sync’ button, just click it and wait for the ‘in sync’ indicator to return. You can then quit Github and go back to your frolicking.

I hope that these instructions have helped you use github, if you have any questions about this or anything else in your project or if you miss my bubbly but sometimes somewhat disturbing humour. Feel free to contact me. My email is wiz_boy_8@hotmail.co.uk and my Skype is wizboy8 Good Luck!