Repo Owner Presentation

* **What is Git?**

Git is version control software that runs on your local machine. It allows members of a team to make independent changes to a project without overwriting others’ work. Team members work separately before merging the work together.

Git also allows you to revert to earlier snapshots of the project.

Further reading: <http://readwrite.com/2013/09/30/understanding-github-a-journey-for-beginners-part-1>

* **What is GitHub**

Github is an online repository hosting service.  I.e. it is a platform that makes it easy for developers to collaborate.

* **How to install Git on your computer**

<http://readwrite.com/2013/09/30/understanding-github-a-journey-for-beginners-part-1>

Scroll down to ‘setting up github and git for the first time’.

or the documentation:

<https://help.github.com/articles/set-up-git/>

* **How to create a new repository on Github and get setup to connect your local Git with the Github repo.**
* Create new repo on GitHub website.
* Create local directory where your project will be: mkdir, cd into it.
* Git init:  this tells the computer to recognise this directory as a git repository.
* You can now start using git commands

**Connect GitHub repo and local repo:**

* Git remote add origin <https://github.com/username/myproject.git>
* Git now knows that there is a remote repo and this is where you want the work that you do locally to go when you push it.
* To check, do: git remote -v    and you should see the url of your remote repo.

**Example for making a first commit and push:**  
git init

git touch readme.txt  
git addreadme.txt  
git commit -m "first commit"  
git remote add origin<https://github.com/foundry-matrix/blog.git>git push origin master

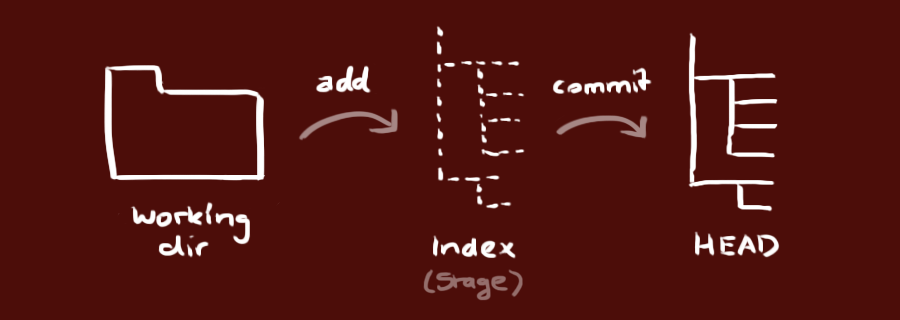
* **Common Git commands**

Please note, all Git commands start with “git”

* git config - validate your username and email
* git init - initialise the git repo
* git clone - clone a git repo from a remote source
* git add - add file from your local to the index.
* *git add -u  — looks at currently tracked files and stages changes if they’ve been altered or removed.*
* *git add . — stages changes that are new files or alterations, but* ***not*** *those removed.*

* *git add -A — combined add-u and add .    Stages new files, changes, and deletions.*
* git commit - takes changes in the index and creates new commit object (milestone) pointing at it. Sets the branch to point at this new commit.  (Needs a comment, -m)
* *git commit -a*  combines ‘git add’ and ‘git commit’.
* git push origin master - pushes all modified local changes up to the remote version on github.
* git status - shows if there’s any changes,  and which branch you’re on.
* git branch - gives you a list of branches and shows where you are.
* git checkout - change branch. //*git checkout -b new-branch-name  is shorthand for creating a new branch and switching to it.*
* git merge — merges a branch back onto the master.
* git branch -d *branch-name* to delete a branch
* git help - Type this into the command line to bring up the 21 most common git commands. You can also be more specific and type “git help init” or another term to figure out how to use and configure a specific git command.

**What’s the Difference Between add, commit and push?**



* + Add moves the changes from your working directory to the ‘index’ - a staging area. Nothing is yet committed.
  + Commit - this commits the files that have been previously added, and creates a new version (a snapshot). The changes not yet reflected in the remote repo.
  + Push - this moves the changes to the remote repo,  so your team can now see it, and pull your code if they want.

**- Common Bash commands**

* mkdir - new directory
* ls - list files
* touch - create file
* rm - delete file
* **Basic workflow with Git and Github.**

1. Forking
2. Creating branches
3. Add