**Git Config Commands**

Now let Git know who you are. This is important for version control systems, as each Git commit uses this information

* git config --global user.name 'username in github'
* git config --global user.email "github email address"

Initialize Git

* git init

Then we check the Git status and see if it is a part of our repo

* git status

Files in your Git repository folder can be in one of 2 states:

Tracked - files that Git knows about and are added to the repository

Untracked - files that are in your working directory, but not added to the repository

Git Staging Envn

* git add fileName
* git add --all / git add -A (to add more than a file at once)
* git status

**Git Add Commands**

Files in your Git repository folder can be in one of 2 states:

**Tracked** - files that Git knows about and are added to the repository

**Untracked** - files that are in your working directory, but not added to the repository

**Git Staging Envn**

* git add fileName
* git add --all / git add -A (to add more than a file at once)
* git status

Adding commit keep track of our progress and changes as we work.

Git considers each commit change point or "save point". It is a point in the project you can go back to if you find a bug or want to make a change.

**Git Commit**

Git considers each commit change point or "save point". It is a point in the project you can go back to if you find a bug, or want to make a change

* git commit -m ‘reason of commit’

**Git Commit without Stage**

*Short status flag* :

* ?? -Untracked files
* A -files added to stage
* M -modified files
* D -deleted files