Git Tutorial

- Git Download
- Visual Studio Code My preferred IDE for any programming language with git integrated

Getting Started

- 0. Ensure you are in a new, clean directory with a README.md file
- 1. Create repo

```
git init
```

2. Set branch name to "master" or whatever branch name you want

```
git branch -m master
```

3. Set origin to GitHub repository. Where <url> is the URL of your git repository.

```
git remote add origin <url>
```

4. Add files to commit

```
git add .
```

5. Commit the files

```
git commit -m "init"
```

6. Intial push. Ensure you are pushing to the name you gave to remote and the branch name you set. In this example the remote name is "origin" and the branch name is "master"

```
git push -u origin master
```

Workflow

The philosiphy of git is to seperate your code changes into small, readable commits. You may stack up as many commits locally as you like, and push them when you are ready.

Typical Workflow

- 1. Make some changes to code/files
- 2. Add them. You can do git add <file> individually instead.

```
git add .
```

3. Commit you changes and annotate what you changed

```
git commit -m "<DESCRIBE YOUR CHANGES>"
```

- 4. Repeat steps 1 3 until you want to update remote repository
- 5. Push commits. If you are pushing a branch that does not exist remotely, you will need to git push -u origin

branch>

```
git push
```

Common Git commands

Initializes repository

```
git init
```

Clones repository into a folder

```
git clone <url>
```

Pull latest commits from repository

```
git pull # -r
```

Most Used Options

• -r rebase all local commits on top commits pulled from repo

Push all local commits to the remote repository

git push

Most Used Options

• --set-upstream origin <branch>/-u origin <branch> Pushes branch to remote

Stage changes to be committed

git add <files>

Most Used Options

• -a will stage all files

Commit all staged changes

This will open a text editor to briefly describe your changes.

git commit

Most Used Options

- -m "<message>" Passes message rather than opening a text editor
- --amend Adds staged changes to the previous commit

List all branches

git branch <name>

If you include a name, it will create a new branch with that name from your currently active branch.

Most Used Options

- -a Lists all local and remote branches
- -d <branch> Deletes a branch

• -m <branch> Renames a branch

Switch to a branch

Should commit all changes before checking out.

```
git checkout <branch>
```

Most Used Options

• -b Creates and switches to a new branch

Resets all commit to HEAD

```
git reset
```

Most Used Options

- HEAD~1 Undos last commit
- --hard Deletes all local changes

Log commits

```
git log
```

Most Used Options

• --pretty=oneline prints everything nicely in one line

Useful Aliases

Aliases are essentially custom git commands that are defined in C:\Users\%USER%\.gitconfig

```
[alias]
    co = checkout
    lg = log --color --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset
%s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --abbrev-commit
```

undo = reset HEAD~1 --mixed
amend = commit -a --amend