Introduction to the used tools

Top10PonyVideos

This document pretends to be a really simplified document with all the tools you will need for working with us. The only knowledge that may be assumed throughout it will be Python. This document will only cover Windows OS though the steps are easily

Sections

1 Visual Studio Code

We will be using Visual Studio Code (henceforth VSC) for most of our programming. Feel free to use any other IDE you want but everybody using the same one will be benefitial in terms of coordinating. The installation is really simple. The program can be found here

2 Install GIT

As you probably know we will be using Github to host our software. If you have never used Github don't worry, all you need to know is that it uses "GIT" which allows us to keep track of every release we make and keep changes organized.
Download GIT. This will also install GIT Bash, a console command we will be using later.

3 How does everything work?

The most basic workflow for a Github project goes like this:

3.1 Cloning

First of all you will want to make a copy of the project in your local machine. Select the folder you are gonna be working on ("Open folder"). To actually clone the repo go to your VSC console (Found within the "view" tab) and use the "git clone" command. First you will need to go to the folder in which you want your project to be. Do so using the "cd" command. For example:

```
cd C:\Users\Derpy\OneDrive\Documents\Top10PonyGithub

git clone https://github.com/TheTop10PonyVideos/
Top10PonyVotingProcessing.git

cd Top10PonyVotingProcessing
```

That last line will put you in the folder you just cloned. Now we can actually work on our project.

3.2 Commit

So, you have made the changes you wanted, awesome. Now you need to commit those changes. The command "git status" is always really useful since it lets you know what changes have been made between your copy and the previous commit. But what exactly is a commit? Well, it's basically letting your system know whatever you recently edited is the final version. Commit does not actually modify anything in Github, just your local system. First you need to add your changes and then you commit them:

```
git add .
git commit -m "MANDATORY COMMIT MESSAGE"
```

If you are only modifying files and not adding one you may use "-am" which adds and commits at the same time

```
git commit -am "MANDATORY COMMIT MESSAGE"
```

You always need a commit message. Please make those as clear and descriptive as possible.

3.3 SSH Keys

Now your local files are what you would like to upload to Github. This process is called "Pushing". But to do so we need Github to know that we are actually who we say we are. This is done with "SSH Keys". Learning how those work is beyond the scope of this document. Open Git Bash (It will have installed alongside GIT) and input the following command:

```
ssh -keygen -t rsa -b 4096 -C "Derpyemail@example.com"
```

Choose whatever file name you want and no passphrase is needed either. Two new files will be in `C:\Users\Derpy\.ssh`.

```
cd .ssh
cat id_rsa.pub
```

This commands will give you a long string of characters ending with your email. Do not share the files with anyone, that would compromise the keys and anyone could pretend to be you. Once you have that long string you can go to your Github account keys settings and add it under whatever alias you like. Lastly we add those credentials to our local system:

```
eval "$(ssh-agent -s)"
ssh-add ~/.ssh/id_rsa
```

`Git branch`	Shows all branches
`Git checkout -b name`	Creates branch “name"
`Git branch -d name`	Deletes branch “name"
`Git checkout name`	Moves you to branch “name"
`Git push origin name`	Pushes your changes to branch “name”

3.4 Pushing

Finally we are ready to actually push our changes.

```
git push origin master
```

This pushes from origin (your regular folder) to master (the standard name for the main branch). Pushing into master is NEVER a good idea and you should always work in a branch and ask for a pull request (henceforth PR) to merge.

3.5 Branching

So, imagine working on master and making a mistake. Well, now you gotta fix master and it’s gonna be down for a while before anyone can use it again. Now imagine a team of a 100 people doing that each working on different details of a massive project... If it sounds like a pain in the ass it’s because it is... or it would be, we have branches to solve that.
Think of a branch as a complete copy of the project that follows a different timeline. You edit your stuff and push it, but the master isn’t actually touched, you just upload a copy of your changes. Others then review those changes and finally approve it, ending the branch. The process of unifying branches is called “merging”. Branches are also usually used to apply hotfixes for example but that is beyond what we are doing. Some commands related to branches:

You can always check your active branch on your status bar (bottom left).

4 Black

This section assumes you have python installed. If for whatever reason you don’t go here.Run the following command:

```
pip install black
```

Then you just need to add the extension for VSC. Go here and follow the instructions. Black should automatically edit your code every time you save.

5 Contact

Feel free to contact me over Discord if you need anything (Vari2508). If I am not available and it is urgent please ping @Tech-team on The Top 10 Pony Videos Discord server.