# Documentation

Buttons Added
All the buttons are working in both TurtleArt and TurtleJS unless specified.

1. Create a new file
2. Save to that file
3. Add
4. Status
5. Commit
6. Commit History
7. Revert back to a commit
8. Diff of last 2 commits
9. Load a File
10. Clone/Pull and Load Turtle Repo
11. Push

Create a new File: This buttons enables the user to create a new .tb file to work on. The repo has to be loaded before creating a file. Prompts user for a file name

Save to the file: Saves pallette code to the file user is working on.

Add: Git Add “File User is working on”

Status: Shows complete Git Status

Commit: Git commit locally

Commit History: Shows the commit history

Revert back to a commit : Gives list of all the commits to the user

Diff of last 2 commits:

Load a File: Load a .tb/.ta file to work on

Clone/pull and load a repo: Clone the turtle code repo if it is not available, check for git pull if the repo is available and then load the repo to work on.

Push: Git Push to the remote repo(Not currently working on TurtleArt)

Example 1

1. Clone/Pull the repo
2. Create a new file in that folder/Load an already existing file in that repo
3. Make changes to the blocks
4. Save it to the file
5. Check the status
6. Git Add to add the current working file
7. Check the status
8. Commit with a commit message
9. Push the changes

Example 2

1. Clone/pull the repo and load the repo
2. Load the file
3. Look at commit history
4. Revert to any of the previous commits.
5. Save it to the file
6. Git add
7. Git status
8. Git commit
9. Push the changes

Documentation for the Backend API(To use the backend functions in any activity)

backend.py contains the code
Functions in the backend.py:(will explain in function in detail)

1. Local init
2. Load repo
3. Create File
4. Edit file
5. Add
6. Get status
7. Get commit history
8. Get commit id and message
9. Clone remote
10. Commit logs
11. Set current file name
12. Get diff
13. Update local
14. Push

For JS activities:

The original TurtleBlocks JS project is mantained at https://github.com/walterbender/turtleblocksjs . This project was a part of GSOC 2016 and the aim of this project was to add git functionalities to the web activity so that user can use basic git functionalities
in the activity itself and need not have the knowledge of git but use it indirectly as a tool to save things. This project uses the git backend code https://github.com/vikramahuja1001/GitBackend which is based on dulwich(https://www.dulwich.io). Basic information of Git is necessary before using the Git Backend. Give Git a try- https://try.github.io/levels/1/challenges/1 . Backend app contains the code for the main TB activity. Test app is just a test flask app to test some of its functionalities.<br />

Git Functions Implemented:<br />
1. Create a new file - Create a new .tb file to work on. The user will get prompted for a file name. This file will be saved in turtle folder(repo) and will contain the necessary code. Will also set the current file value to the new file created.
2. Save to file - Saves the palette code in the file which is loaded into the activity, i.e. the current file.
3. Add - Similar to Git Add Command.
4. Status - Similar to Git Status Command
5. Commit - Similar to Git Commit, remember to use add button before committing. User will be prompted to enter a commit message.
6. Commit History - Shows the commit history, i.e., the total number of commits, their timestamp and the person who made them.
7. Revert back to a commit - Revert to any of the previous commit. In order to make any previous commit as the latest state
of the commit, just commit again after reverting to that commit.
8. Diff of last 2 commits
9. Load a File - Load an existing .tb file to the palette.
10. Clone/Pull and Load Turtle Repo - It will clone the Turtle Codes repo (https://github.com/vikramahuja1001/TurtleCodes) if not available
,if it is available then it will check for changes using Git Pull and load that repo for
11. Push - Similar to git push. Will push to the remote Turtle Codes repo (https://github.com/vikramahuja1001/TurtleCodes)

Backend App:
The turtle folder contains the .tb files which contains turtle blocks code, the diffs folder contains equal number of files as there are in the turtle folder. The files in diff folder are named as diff_file_(filename) and they contain the n*(n-1) diffs if there are n commits in that file. The diff file is updated each time Revert back to a commit or Diff of 2 last commits

functions are called. The app folder contains the main code. Static folder in it contains all the TurtleBlocks JS code which converted to the flask application. Flask is required to run the JS activities. A views.py file has to be created which will create necessary functions to be called by the browser. XMLHttpRequest has to be used from the browser side to connect the JS functions to their python counterpart.

There are 3 main files in the app folder:

1. views.py : Contains corresponding python code for the implemented git functions called by the browser.

2. diff_func.py : Code for creating the diff file of commits

3. backend.py : Necesary code for creating the backend.

For python activities: