

GIT: Basics

Start

FYI: `rm -rf .git`

Tracking Changes

Set up Atom

make the terminal window available

`CRTL+.` -> FIND KEYMAP key `TAB` -> make empty thingy

- `ATOM/FILE` -> KEYMAP

`'atom-workspace': 'ctrl-§': 'platformio-ide-terminal:toggle'`

`'atom-workspace': 'ctrl-dead': 'platformio-ide-terminal:toggle'`

First commit

1. open project
2. create `bohemian.md`
 - `'Is this the real life? Is this just fantasy? Caught in a landslide No escape from reality'`
1. `git init`
2. `git status`
3. `git add . / bohemian.md`
4. `git status`
5. `git commit -m 'Setting up that rhapsody!'`
6. `git status`
7. `git log`
8. `ls` (Where are my changes? No clutter!)

Change the file

1. Edit file:
 - ‘Open your eyes Look up to the skies and see I’m just a poor boy, I need no sympathy Because I’m easy come, easy go’
1. `git diff`
2. `git commit -m ‘Open your eyes!’`
3. `git status`
4. `git commit -am ‘Open your eyes!’`
5. Show <http://swcarpentry.github.io/git-novice/04-changes/index.html>

Staging Details

1. Edit file:
 - ‘A little high, little low Anyway the wind blows, doesn’t really matter to me, to me’
1. `git add .`
2. `git diff` (nothing)
3. `git diff --staged`
4. `git commit -m ‘A little high?’`
5. `git status`
6. `git log`

What have we learned?

1. **git init**: Start tracking files
2. **git status**: What does Git see? (our best friend)
3. **git add filename / .**: Stage filename / everything
4. **git commit -m**: Create a snapshot of all staged files
5. **git commit -am**: Stage all files previously added and commit them
6. **git log**: See the commit history
7. **Extra**: Delete **.git** subfolder to start from scratch

EXERCISE

1. Go to a new folder which is **not** a subfolder of a Git repository
2. Initialize your Git repository
3. Create a new file
4. What does Git say (**git status**)?
5. Stage the changes
6. What does Git say?
7. Commit them
8. Add some lines to your file and delete others
9. Look at the differences in Git
10. Stage and commit your changes
11. Now change only single words in your file
12. Look at the differences in Git
13. Try **git diff --color-words**
14. What did the last line do?
15. Stage and commit your changes