



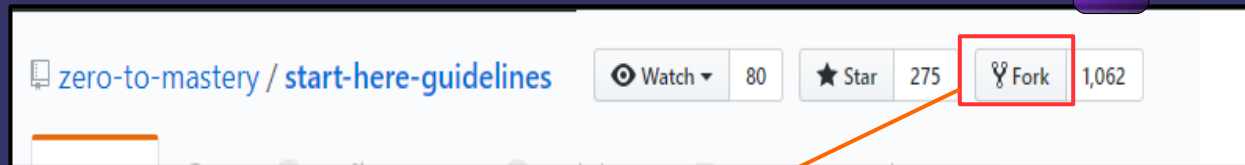
Git and GitHub guide Part 3

Contribution To Open Source

Zero To Mastery
by Andrei Negoie

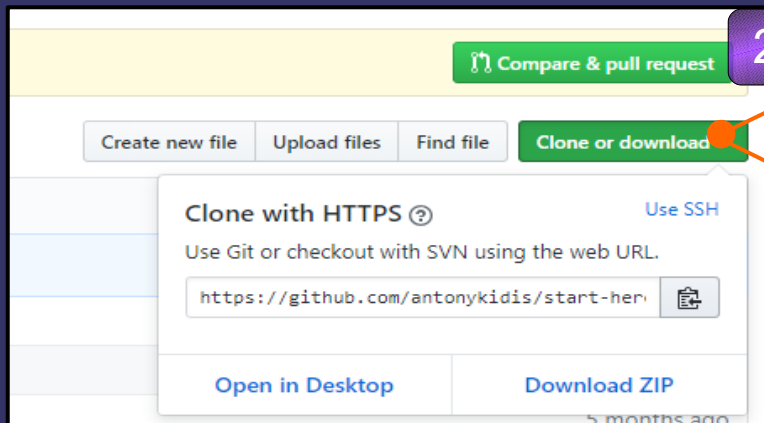
Lecture 148. We start from **05:41 minutes** (follow these steps)

1



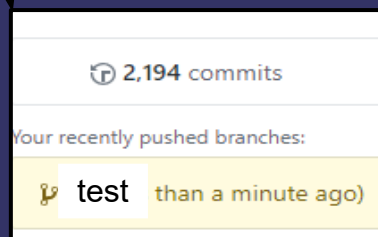
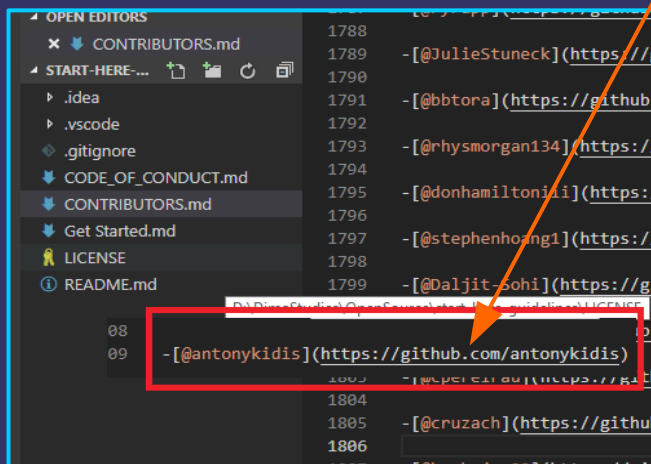
3. git clone <https://github.com/antonykidis/start-here-guidelines.git>
4. Cd into start-here-guidelines.
5. ls

CODE_OF_CONDUCT.md CONTRIBUTORS.md 'Get Started.md' LICENSE README.md



2

6. Type **start code .** (for VS Code should open the entire folder)
7. Branch out git checkout -b test
8. Add your name to the list of contributors
9. Save all CTRL+S
10. **git add .**
11. **git commit -m "test"**
12. **git push origin test**
13. **git push**
14. go back to <https://github.com>
15. See a notification
16. Finish a pull request as appears in the video.



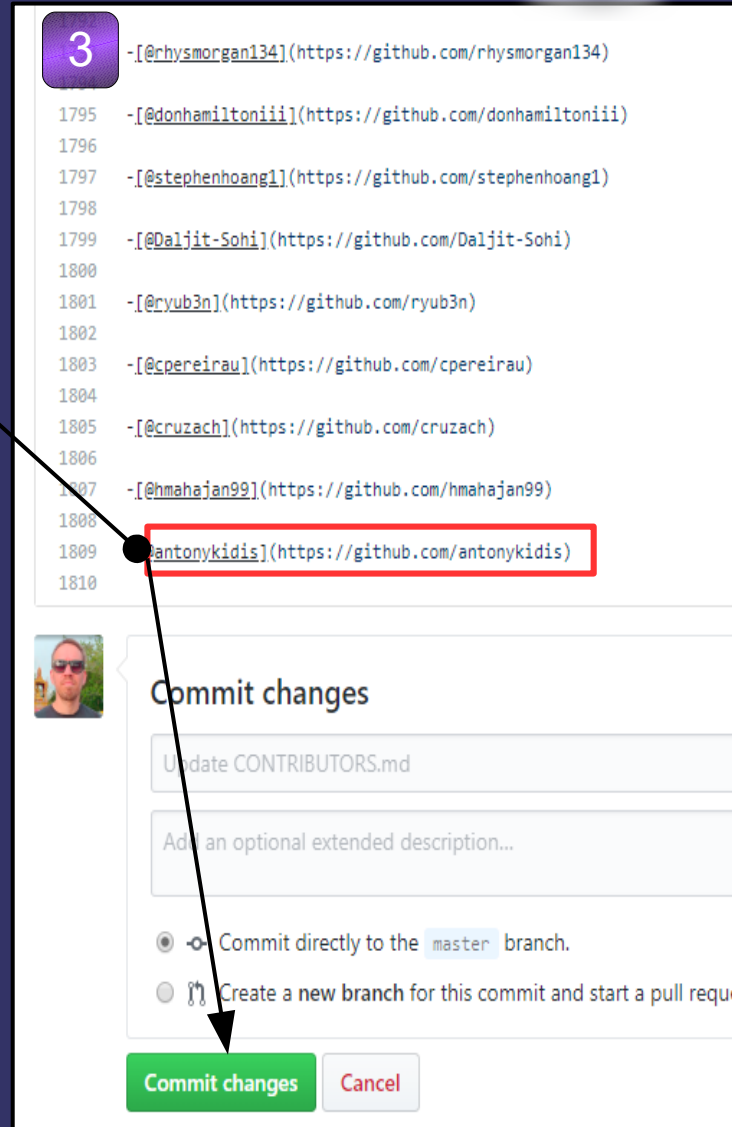
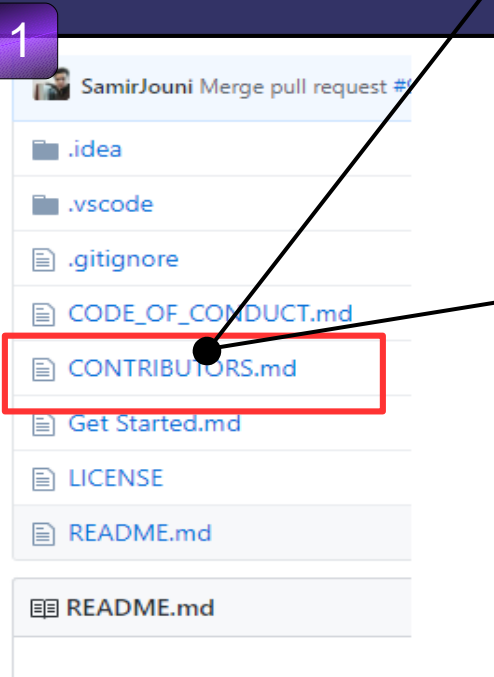
A different way adding yourself to CONTRIBUTORS.MD

We could skip the previous page and do the following

** Another way adding yourself to a CONTRIBUTORS.md list would be editing it straight on a start-here-guidelines GitHub repository.*

1. Go to the forked repo <https://github.com/antonykidis/start-here-guidelines>
2. At your left hand side click the CONTRIBUTORS.md file.
3. Click Edit (pencil icon)
4. Add your name to the very end of the contributors list (using the mark down syntax)
5. Commit changes.

I've decided to add myself this way

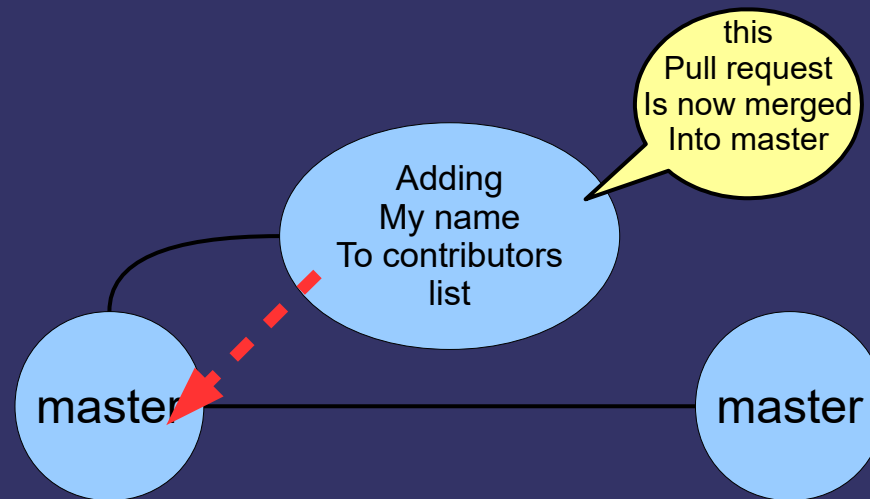


6. You will receive a new Pull Request notification.
7. Finish this procedure by clicking Compare and Pull request button.
8. Administrators will accept this pull request and merge it into master.

Congratulations on adding yourself as a new contributor!

*...Meanwhile somewhere in the cosmos
One of the administrators have merged
our pull request into the master.*

We've been added to a CONTRIBUTORS.md list.

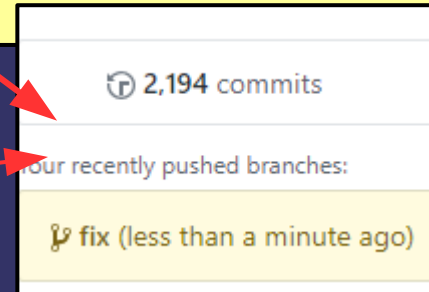




After adding myself to a contributors list via GitHub website, I finally decided to repeat these steps, but this time by cloning a repository, adding myself to the Contributors list, and pushing it back to a gitHub (as it appears in the original video)

I can say that I've edited the contributor's list two times so far.

1. By editing it on a GitHub website.
2. By cloning a repo, modifying a contributors list, and pushing it back to a GitHub
3. This caused the notification to appear once again.



Okay I have a new pull request notification.

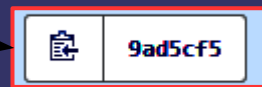
But I don't need it because my name is already in the contributors' list
Hmmm...

Is there any way to get rid of it?
It's a good question...



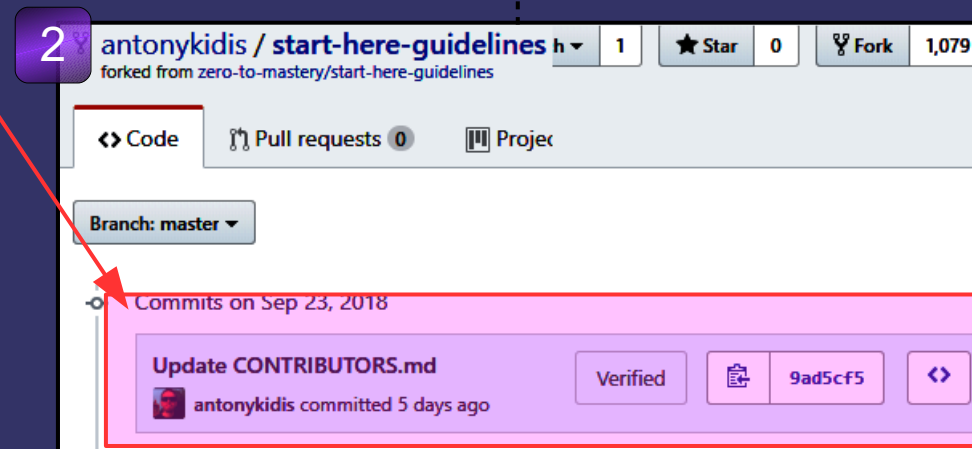
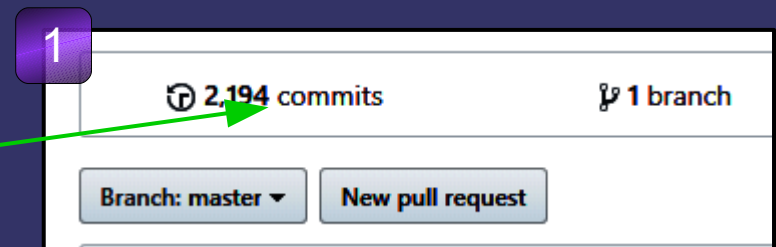
We could try to delete the commit like this

1. Go to a forked repo <https://github.com/zero-to-mastery/start-here-guidelines>
2. Click commits link.
2. Locate your commit.
3. Copy the commit ID
4. Go to terminal.
5. type `git push origin +9ad5cf5^:master`



+ Specify what destination ref to update with what source object.
The format of a <refspec> parameter is an optional plus +, followed by the source object <src>, followed by a colon :, followed by the destination ref <dst>

^ HEAD^ means the first parent of the tip of the current branch.



- You will get the following output
- Go back to commits (gitHub) and check if it's been deleted.
- Repeat the previous steps again to delete another desired commits you want.

- Once every commits is removed, the pull request Notification will no longer appear on the GitHub website.

- You can then delete a remote branch like this:

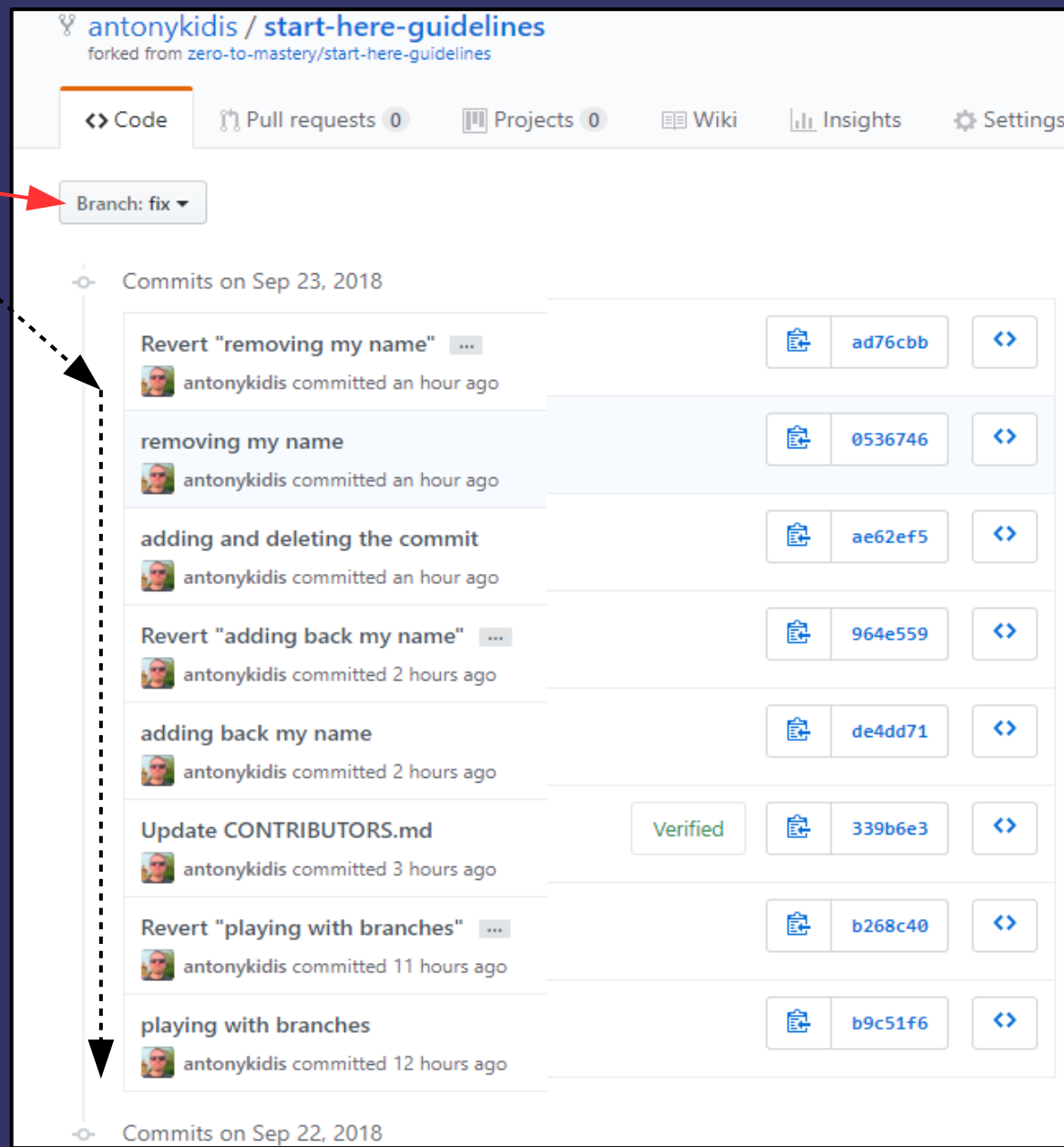
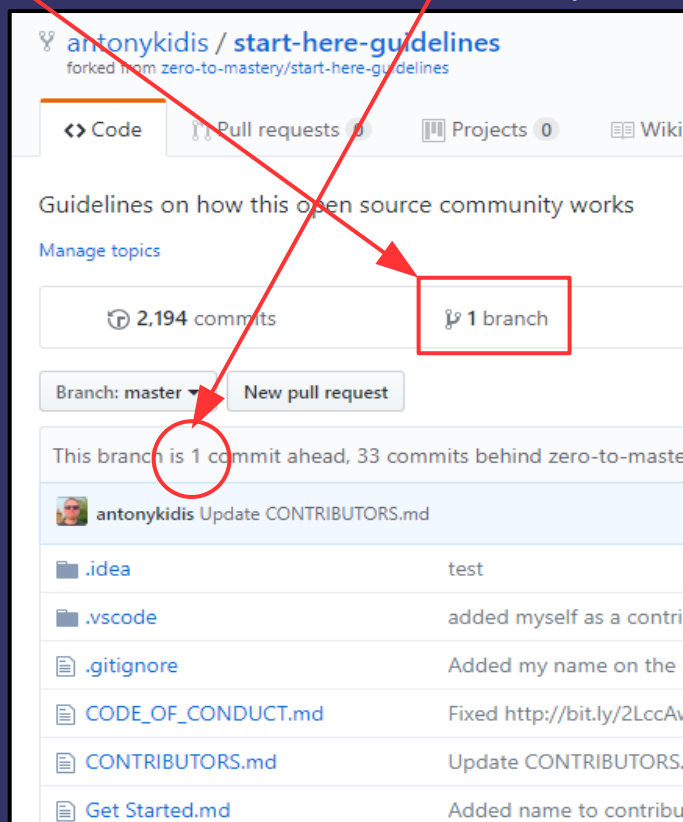
- Type `git push origin --delete fix`

In my case **fix** was the name of the branch

As you can see i've successfully deleted Commits and a remote branches.
We now left with only one commit, and one master branch on our forked repository.

```
Dima Mironov@Dmitry-M MINGW64 /d/DimaStudies/OpenSource/start-here-guidelines (fix)
$ git push origin +19fe22b^:fix
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/antonykidis/start-here-guidelines.git
+ 19fe22b...e79238a 19fe22b^ -> fix (forced update)

Dima Mironov@Dmitry-M MINGW64 /d/DimaStudies/OpenSource/start-here-guidelines (fix)
$ |
```



Additional Information

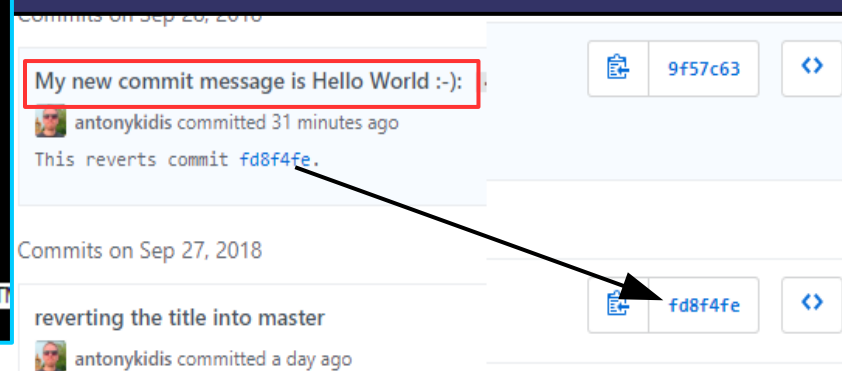
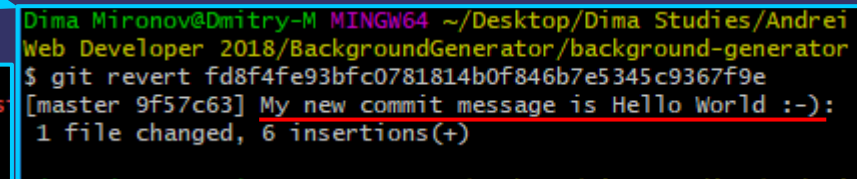
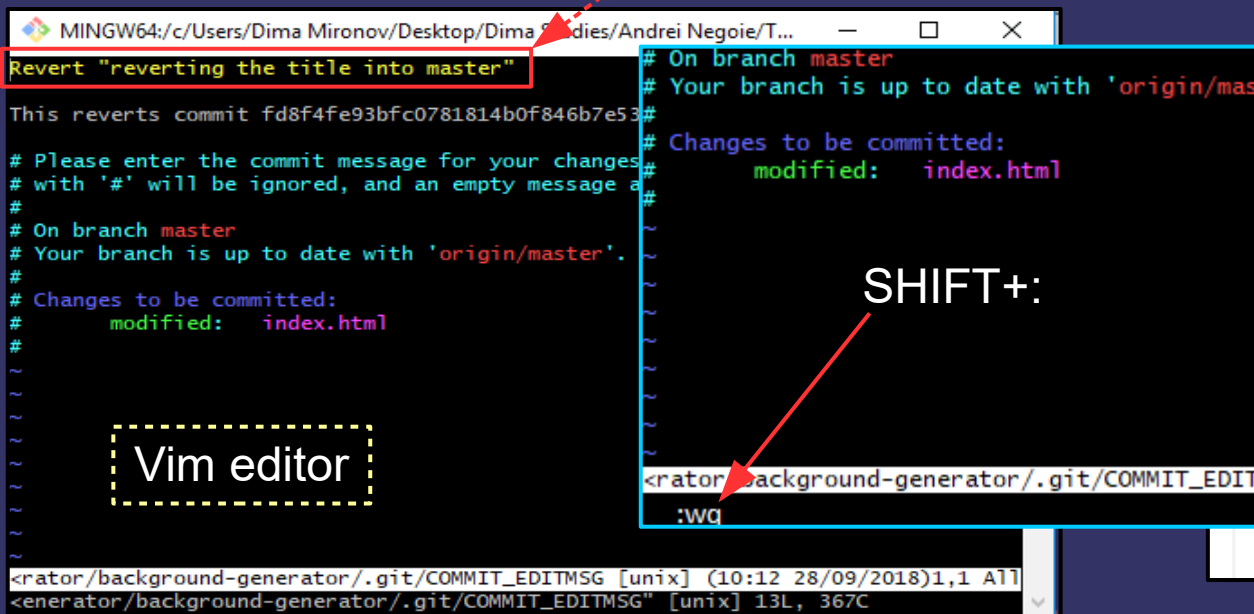
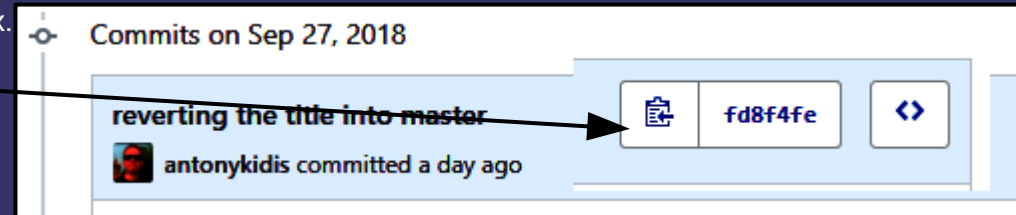
You could encounter with some unexpected things while working on the Git and GitHub section of the course.

They are as follows:

1. What if we mistakenly entered a bad commit's message. How to fix that without pushing a whole new files into the GitHub again?
2. Your bash terminal showed a weird vim editor window (only for win. Users) How to use it?

Solution: How to Edit a bad commit message.

1. Go to a forked repository, open commits link, and find a commit you want to fix.
2. Copy a commit's ID to a clipboard
3. Open terminal window.
4. Use this command. `git revert fd8f4fe`
5. hit ENTER.
6. The vim editor window will appear.
7. The yellow text will represent a commit message.
8. Hit **insert** button on your keyboard. This will allow you to manipulate a text.(use arrows to navigate)
9. Delete the yellow text, and write **hello world** (for example)
- 10.Push ESC button to exit the text editor mode.
- 10.Hold down **SHIFT+ :** this will bring you to the vim script mode(check the left bottom corner, you will put commands there)
11. Type **wq** (w= write) (q=quit saving changes) and hit ENTER
12. You will receive the following output, with updated commit message
13. type **git push** to apply changes



Step 14. Go back to GitHub and see the changes

antonykidis / background-generator

Unwatch 1 Star

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings

My new commit message is Hello World :-):

This reverts commit fd8f4fe.

master

antonykidis

 committed 38 minutes ago

1 parent fd8f4fe commit 9f57c63d3bc36e11a98b15

Showing 1 changed file with 6 additions and 0 deletions.

6 index.html

	@@ -11,7 +11,13 @@
11	11 </head>
12	12 <body id="gradient">
13	13 <div class="cntr">
	14 + <<<<<< HEAD
	15 + <<<<<< HEAD
	16 + =====
14	17 <h1>Background Generator!</h1>
	18 + >>>>>> master
	19 + =====
	20 + >>>>>> 17720583859bd152a1731c8c054a77840d9ac9f8
15	21 <input class="color1" type="color" name="color1" value="#00ff00">
16	22 <input class="color2" type="color" name="color2" value="#ff0000">
17	23 <input class="color2" type="color" name="color2" value="#ff0000">

We've just successfully changed the commit's message. But the files are left the same. Great. It works.

Dmitry



A few things to keep in mind before we continue

- ➔ Sometimes you will merge a master branch to one of your other branches, just to keep things up to date.
- ➔ You will probably encounter this VIM editor window again and again, especially if you on windows.
- ➔ Take a few minutes to read a short Introduction to VIM (in the next section), to avoid unpredictable situations, and other questions you might encounter with.
- ➔ Usually gitbash will automatically open ~/.git/commit_EDITMSG file
- ➔ You can edit this file using notepad. But gitbash does it in a more intuitive way without exiting the shell (all in one solution)
- ➔ No need to learn VIM, simply learn the basic commands.

A short introduction to VIM

WHAT IS VIM?

A brief introduction

by: Tatiana Tylosky

WHAT IS VIM?

VIM is a text editor that lives inside the terminal. VIM allows for more efficient coding and workflow!

WHY VIM?

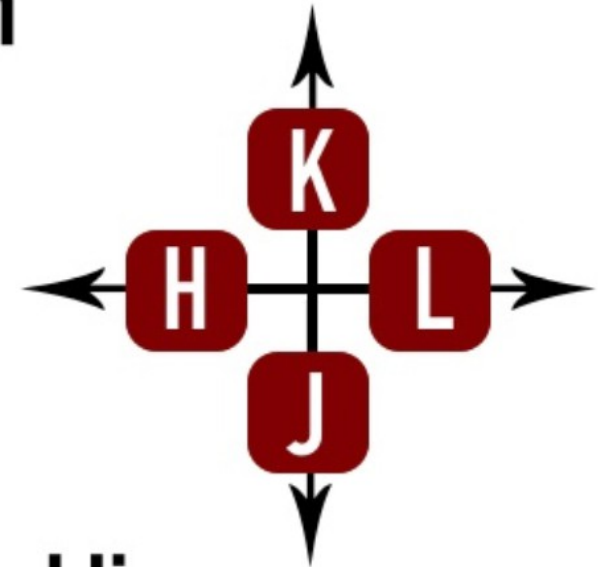
“Why use vim when I already use sublime?”

- To save time**
 - Very efficient way of editing text**
 - Vimscrip is a programming language for text editing**
- Extremely customizable for your particular work habits**
- To be cool.**
 - :%s/cool/a total badass/g**

VIM BASICS

Type “vim yourfilename” into your terminal
to open your file in vim

- Navigate text space in vim using hjkl
- Use the vim command line
 - “:” lets you enter the vim command line
 - “:q” quits vim
 - “:w” writes changes (aka saves file changes)



DON'T BE THIS GUY

If you learn nothing today, please at least learn that
you can exit vim using “:q” or “:q!”



I Am Developer
@iamdeveloper



Following

I've been using Vim for about 2
years now, mostly because I can't
figure out how to exit it.

[↩ Reply](#) [↻ Retweet](#) [★ Favorite](#) [⋮ More](#)

RETWEETS

4,846

FAVORITES

2,105



4:56 AM · 18 Feb 2014



VIM MODES



vim has two modes and normal mode isn't "normal"

#1 Normal Mode – Enter via [ESC]

In this mode you have access to ENDLESS “vim commands” that are useful shortcuts for editing text

Some fun examples:

a – append, u – undo, dd – delete line

#2 Text edit mode – Enter via a, i, and more

Your keyboard acts likes you would expect

VIM COMMANDS

There are SO many vim commands that you will learn more and more new ones every day!

Here are JUST A FEW useful examples

- Action commands**

- 'a' for append**
- 'd' for delete**
- 'u' for undo**

- Movement commands**

- '0' beginning of line**
- 'w' beginning of word**
- '\$' end of line**

But wait that is not all!

Commands have additional tricks that make them even better!

NORMAL MODE COMMANDS

Normal mode commands have a format so that they are easily repeatable over a specific range.

operator [number] motion

Where:

operator – is what to do, such as **d** for delete

[number] – is an optional count to repeat the motion

motion – moves over the text to operate on, such as **w** (word),
\$ (to the end of line), etc.

Ex. 1. “**d2w**” – deletes the next two words

Ex. 2. “**vi(**” – select inside parentheses

NEXT LEVEL VIMSCRIPT

Example: Mapping

Mapping keys lets you tell Vim:

“When I press this key, I want you to do this stuff instead of whatever you would normally do.”

If you type
“:map <space> u”
vim will now undo
actions when you
press the space bar

Things to do with the time you save using vim

- **Code more**
- **Go hiking**
- **Go to brunch**
- **SO MUCH MORE!!**

With vim the possibilities are ENDLESS!

VIM RESOURCES

- Type “vimtutor” into your terminal
- Play Vim Adventures

[Http://vim-adventures.com](http://vim-adventures.com)

- Read

<https://brigade.engineering/crap-wrong-mode-9021375c2826>

<http://learnvimscriptthehardway.stevelosh.com>

WHAT CRITICS ARE SAYING!

“The power of Vim’s text manipulation shortcuts coupled with its extensibility have made other editors feel inadequate.” – some dude

“Yes, Tatiana, I use vim.” – Saul Diez-Guerra

“Cool kids use vim:wq” – anonymous source

The End of part 3

