The complete Web Developer in 2018

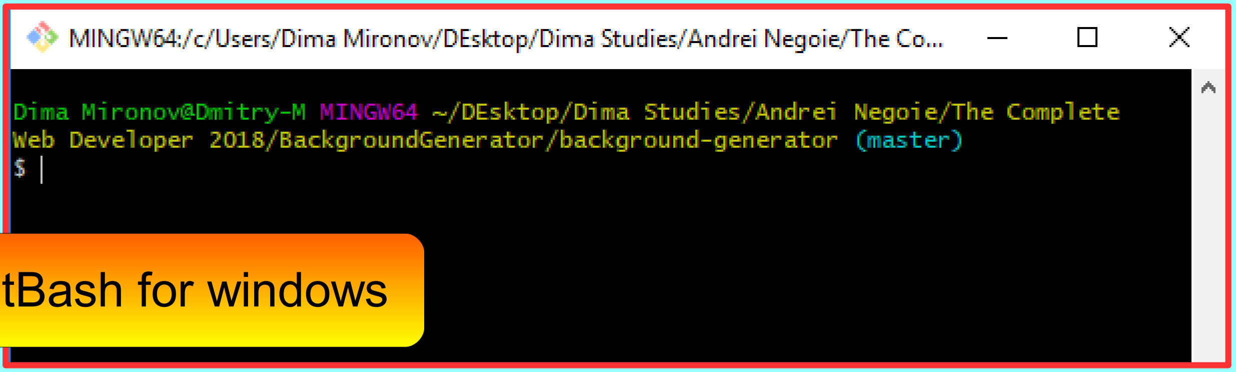
Git and GitHub Guide

Part-2

Branching

Info Page

For all the windows users.
*You can u*se Gitbash terminal, or a Visual Studio Code terminal.
Download it here https://git-scm.com/downloads

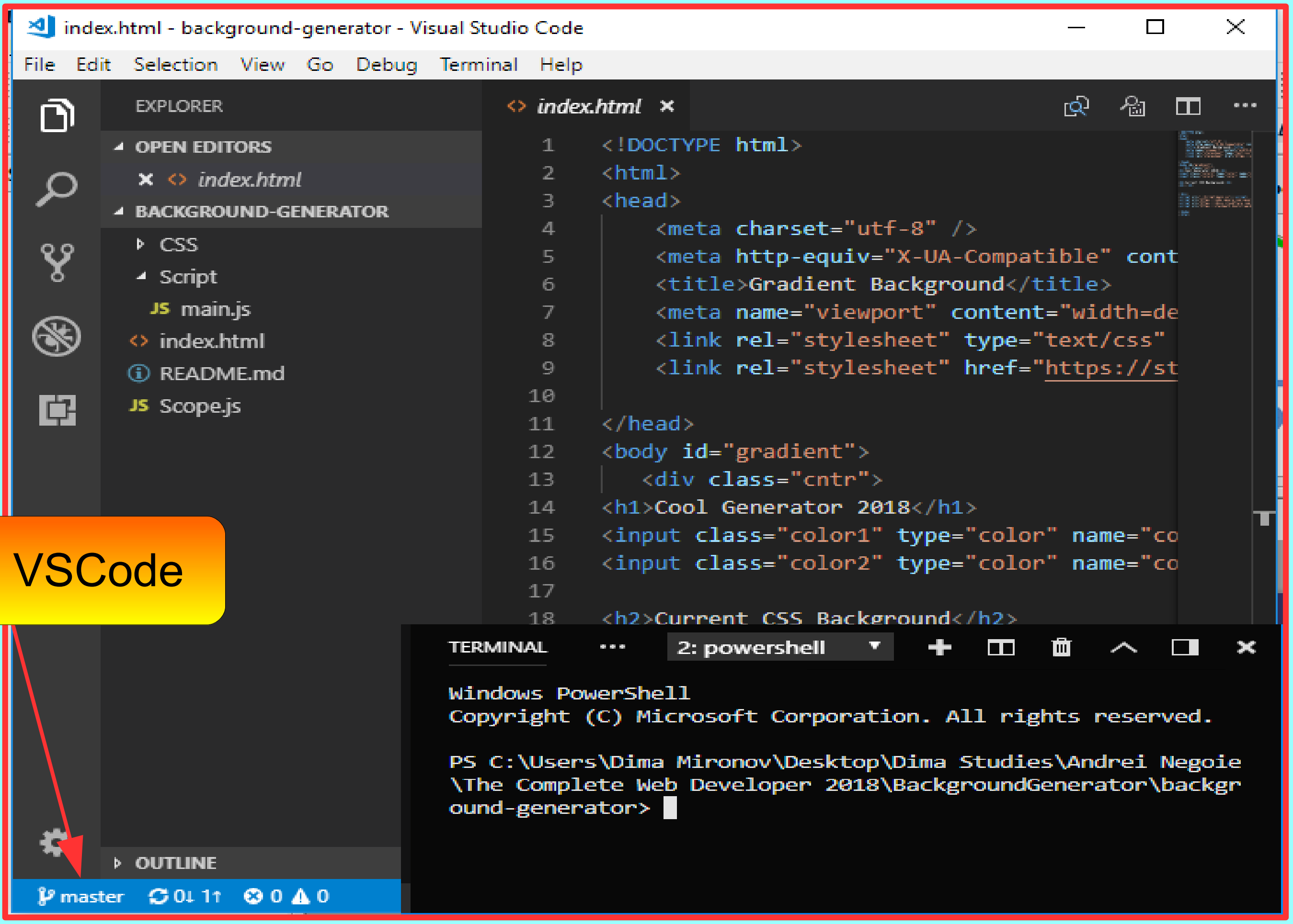

MINGW64:/c/Users/Dima Mironov/DEsktop/Dima Studies/Andrei Negoie/The Co...

```
Dima Mironov@Dmitry-M MINGW64 ~/DEsktop/Dima Studies/Andrei Negoie/The Complete
Web Developer 2018/BackgroundGenerator/background-generator (master)
$ |
```

GitBash for windows

VS Code displays a branch name.
However, It doesn't display a branch name In the PowerShell terminal.

My suggestion is to work with the Git Bash terminal.


index.html - background-generator - Visual Studio Code

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta http-equiv="X-UA-Compatible" cont
    <title>Gradient Background</title>
    <meta name="viewport" content="width=de
    <link rel="stylesheet" type="text/css"
    <link rel="stylesheet" href="https://st

</head>
<body id="gradient">
   <div class="cntr">
<h1>Cool Generator 2018</h1>
<input class="color1" type="color" name="co
<input class="color2" type="color" name="co

<h2>Current CSS Background</h2>
```

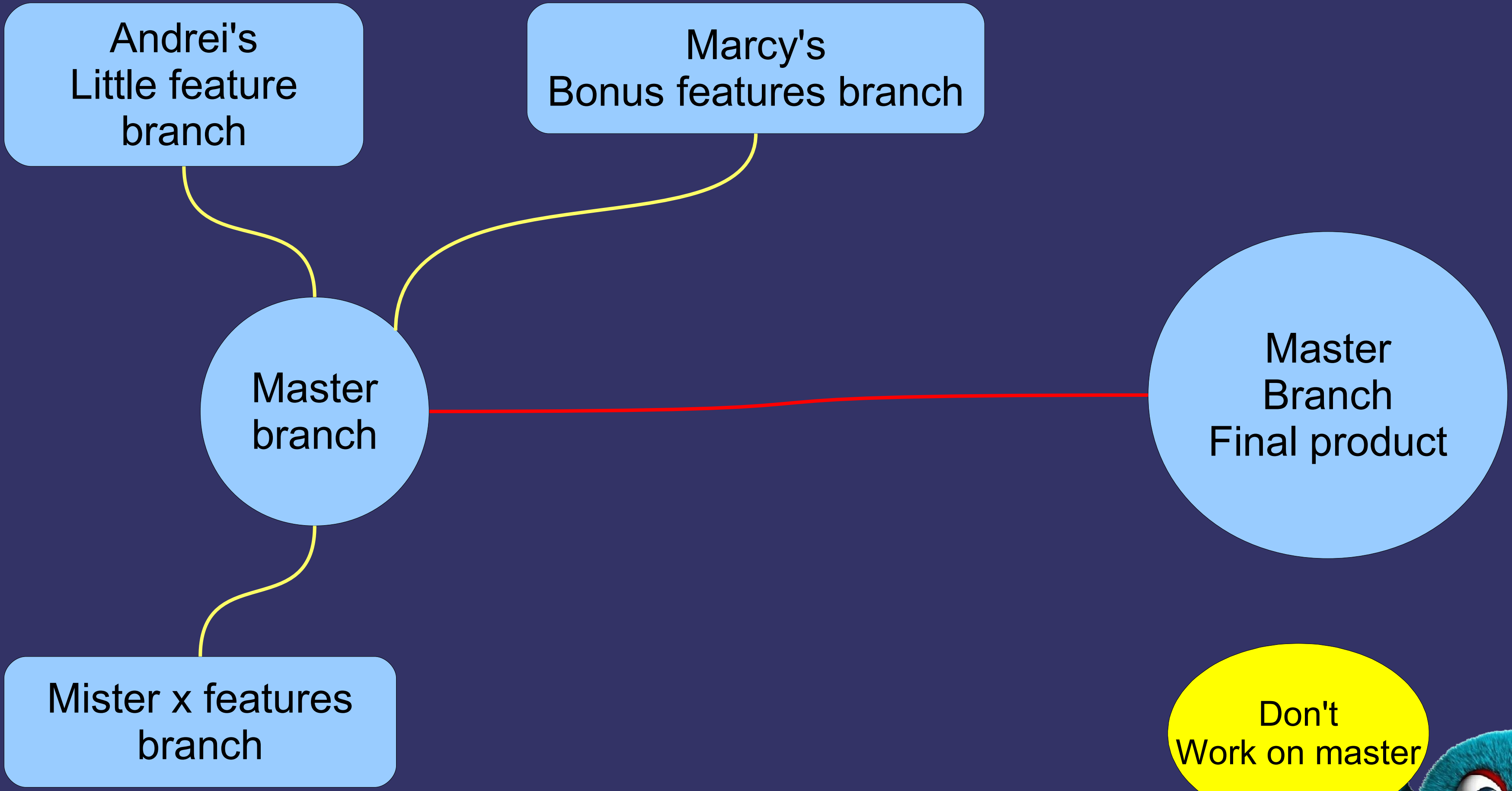
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Dima Mironov\Desktop\Dima Studies\Andrei Negoie
\The Complete Web Developer 2018\BackgroundGenerator\backgr
ound-generator>
```

VSCode

Refers to Section 15, video #145

Git and GitHub Guide Part-2 Branching

The complete
Web Developer in 2018


Andrei's Little feature branch
Marcy's Bonus features branch
Master branch
Master Branch Final product
Mister x features branch

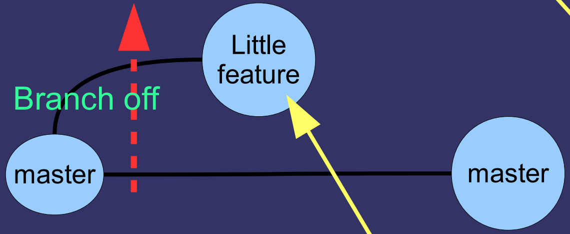
Create a new branch

Let's create a new branch (terminal)

1. Type git branch littlefeature (this will create a new branch)

2. Type git branch (will display all branches available)

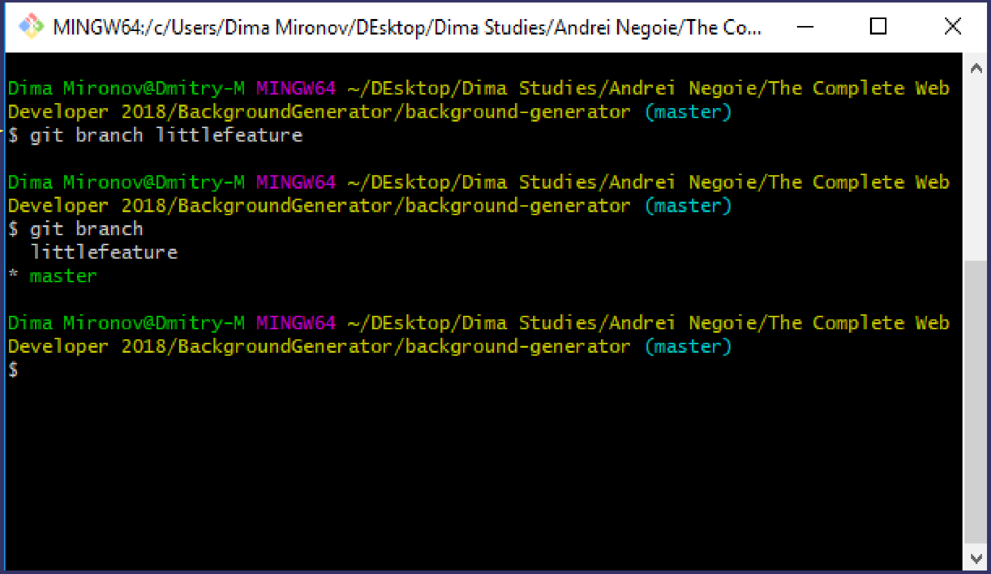
3.Type git checkout littlefeature (Switch to a new branch)



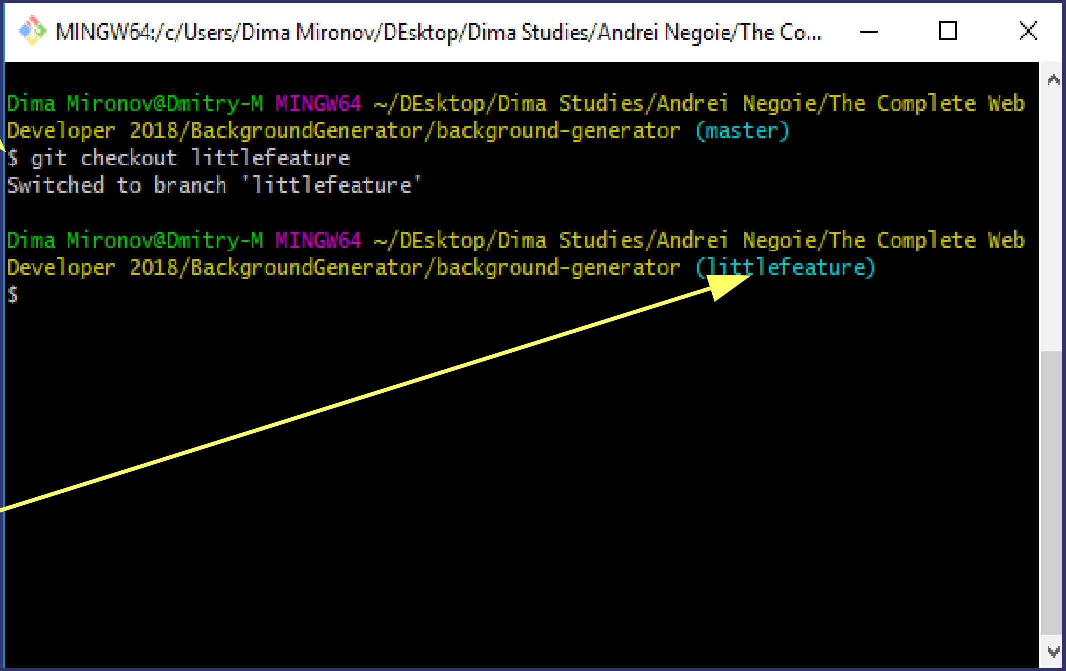
* master means we currently on master branch.

Note:
If something went wrong and you need to delete the branch, type git branch -d littlefeature
The above command will delete the mistaken branch

4. We successfully switched to a new branch


MINGW64:/c/Users/Dima Mironov/DEsktop/Dima Studies/Andrei Negoie/The Co...

```
Dima Mironov@Dmitry-M MINGW64 ~/DEsktop/Dima Studies/Andrei Negoie/The Complete Web
Developer 2018/BackgroundGenerator/background-generator (master)
$ git branch littlefeature

Dima Mironov@Dmitry-M MINGW64 ~/DEsktop/Dima Studies/Andrei Negoie/The Complete Web
Developer 2018/BackgroundGenerator/background-generator (master)
$ git branch
  littlefeature
* master

Dima Mironov@Dmitry-M MINGW64 ~/DEsktop/Dima Studies/Andrei Negoie/The Complete Web
Developer 2018/BackgroundGenerator/background-generator (master)
$
```


MINGW64:/c/Users/Dima Mironov/DEsktop/Dima Studies/Andrei Negoie/The Co...

```
Dima Mironov@Dmitry-M MINGW64 ~/DEsktop/Dima Studies/Andrei Negoie/The Complete Web
Developer 2018/BackgroundGenerator/background-generator (master)
$ git checkout littlefeature
Switched to branch 'littlefeature'

Dima Mironov@Dmitry-M MINGW64 ~/DEsktop/Dima Studies/Andrei Negoie/The Complete Web
Developer 2018/BackgroundGenerator/background-generator (littlefeature)
$
```

Let's modify index.html now

1. Open index html (use IDE or terminal) Type: start code index.html

2. Locate the <h2> tag and change the title to
   <h2>This is thebackground</h2>

3. Save all, then do the following procedure...

4. git add index.html
5. git commit -m“changing title“
6. git push

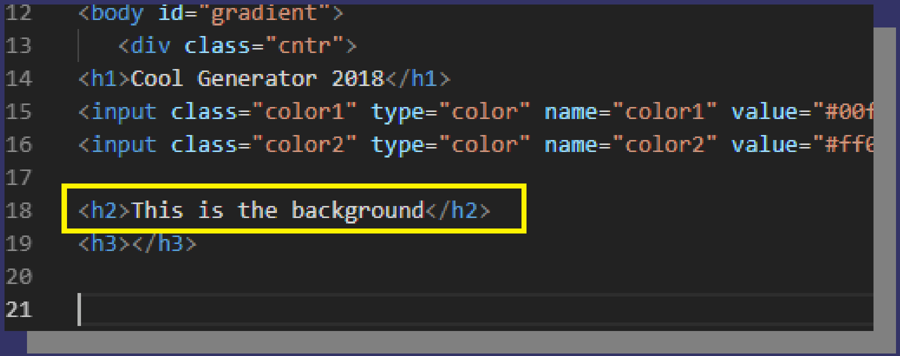
If it doesn't work and you get an error

fatal: The current branch littlefeature has no upstream branch.
To push the current branch and set the remote as upstream,
use git push --set-upstream origin littlefeature

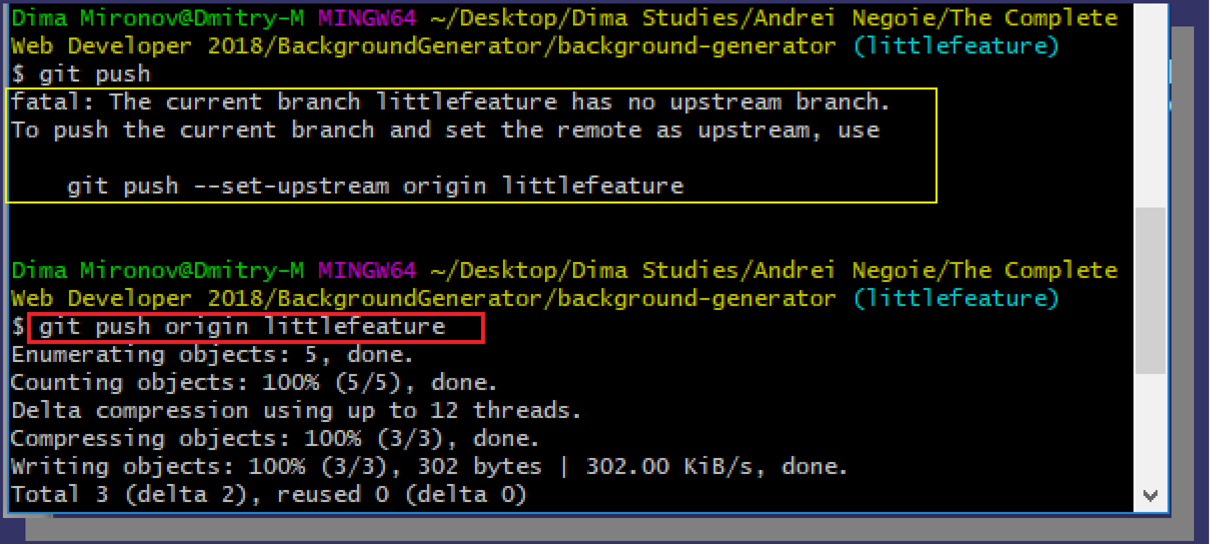
Fix.
Type git push –set-upstream origin littlefeature

After using the above command once, you can then use a regular git push command (without notifications)

Finally, you can cheat and type this
git push origin littlefeature



```
<body id="gradient">
  <div class="cntr">
<h1>Cool Generator 2018</h1>
<input class="color1" type="color" name="color1" value="#00f
<input class="color2" type="color" name="color2" value="#ffe

<h2>This is the background</h2>
<h3></h3>

|
```

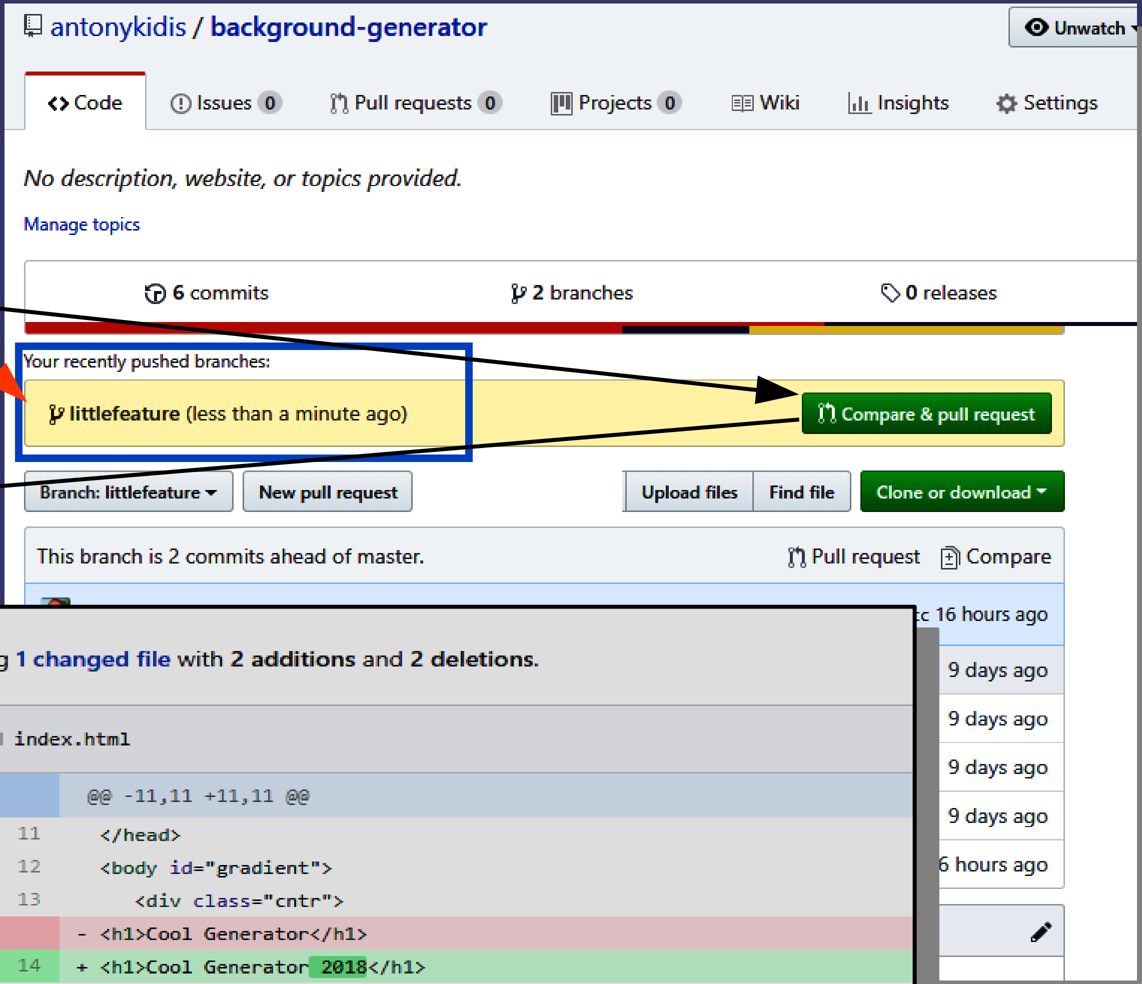


```
Dima Mironov@Dmitry-M MINGW64 ~/Desktop/Dima Studies/Andrei Negoie/The Complete
Web Developer 2018/BackgroundGenerator/background-generator (littlefeature)
$ git push
fatal: The current branch littlefeature has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin littlefeature


Dima Mironov@Dmitry-M MINGW64 ~/Desktop/Dima Studies/Andrei Negoie/The Complete
Web Developer 2018/BackgroundGenerator/background-generator (littlefeature)
$ git push origin littlefeature
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 302 bytes | 302.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0)
```

Go Back to https://github.com

1. We now see a yellow notification
   You recently pushed branches

2. If we click the green button
   Compare & pull request
   It will open the a pull request window



antonykidis / background-generator

Unwatch

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

No description, website, or topics provided.

Manage topics

6 commits | 2 branches | 0 releases

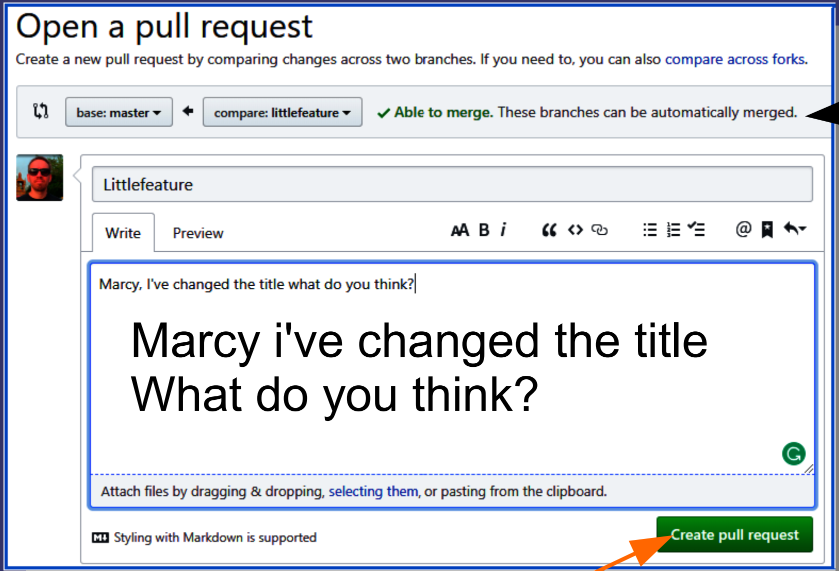
Your recently pushed branches:

littlefeature (less than a minute ago) — Compare & pull request

Branch: littlefeature | New pull request | Upload files | Find file | Clone or download

This branch is 2 commits ahead of master. — Pull request | Compare

16 hours ago
9 days ago
9 days ago
9 days ago
9 days ago
6 hours ago



Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

base: master ← compare: littlefeature ✓ Able to merge. These branches can be automatically merged.

Littlefeature

Write | Preview

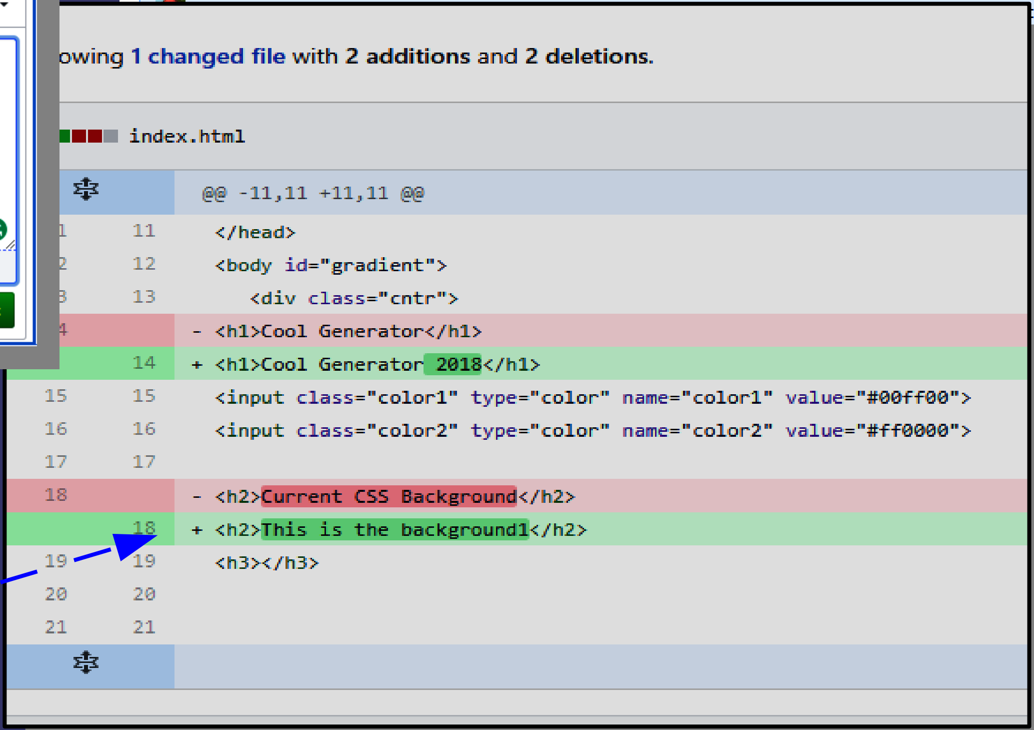
Marcy, I've changed the title what do you think?

Marcy i've changed the title
What do you think?

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

Create pull request



owing 1 changed file with 2 additions and 2 deletions.

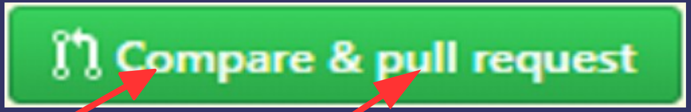
index.html

```
@@ -11,11 +11,11 @@
   </head>
   <body id="gradient">
     <div class="cntr">
 - <h1>Cool Generator</h1>
 + <h1>Cool Generator 2018</h1>
   <input class="color1" type="color" name="color1" value="#00ff00">
   <input class="color2" type="color" name="color2" value="#ff0000">

 - <h2>Current CSS Background</h2>
 + <h2>This is the background1</h2>
   <h3></h3>
```

3. Eneter a message, check the changes
   And finally click the create pull request
   button

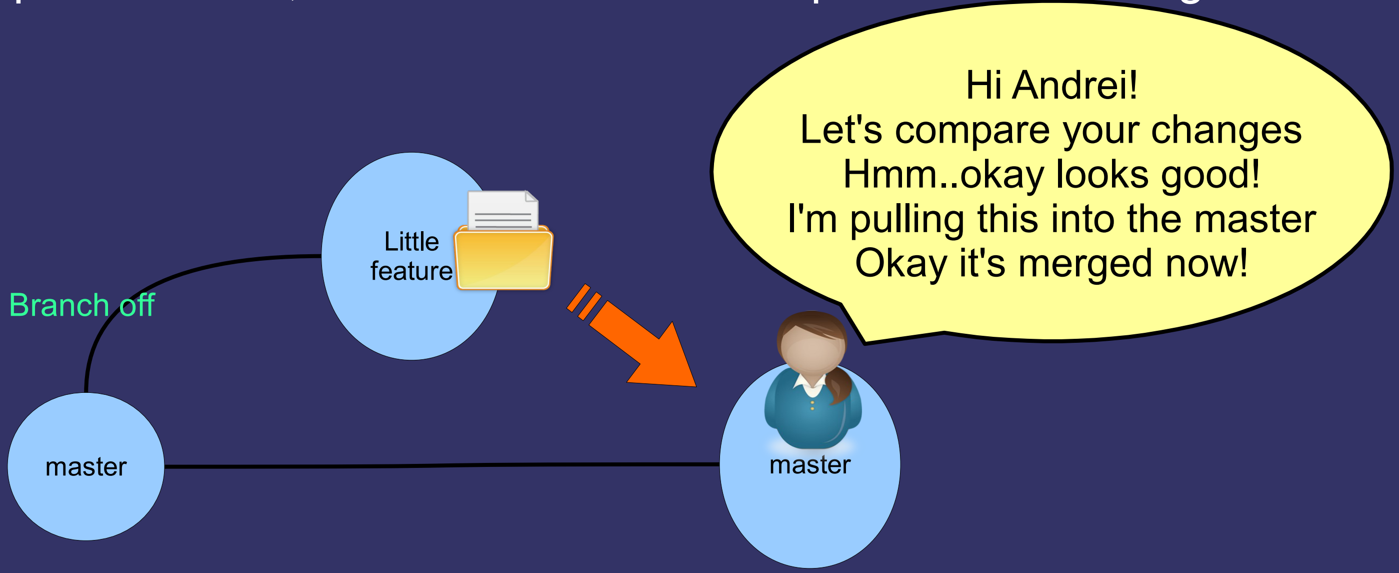
Note:
We've made a change to index.html
Marcy will decide aprove it or not.

Okay now STOP!
Do you really understand what is going on here!?
No...
Yes or Yes!?



Hey Marcy I made a pull request. Please look at my html file,
compare new changes to the previous one.
And if they good enough please pull them, and merge them into the master branch.

Marcy is an administrator and she will open your pushed files on a littlefeature branch.
She will compare them to the previous one, and decide whether to prove these changes or not



Important
At this point you as a developer
Just need to wait for Marcy's approval.
That's it.
Don't be confused by the rest of the diagrams.

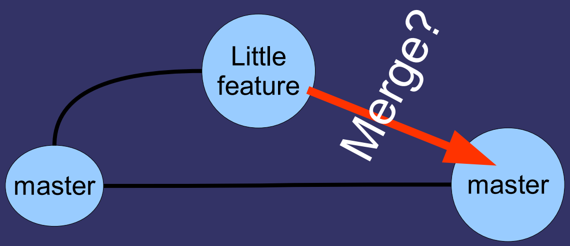
Let's see How Marcy handle my request on her side as Administrator.

With this information in mind let's Proceed to the next step.

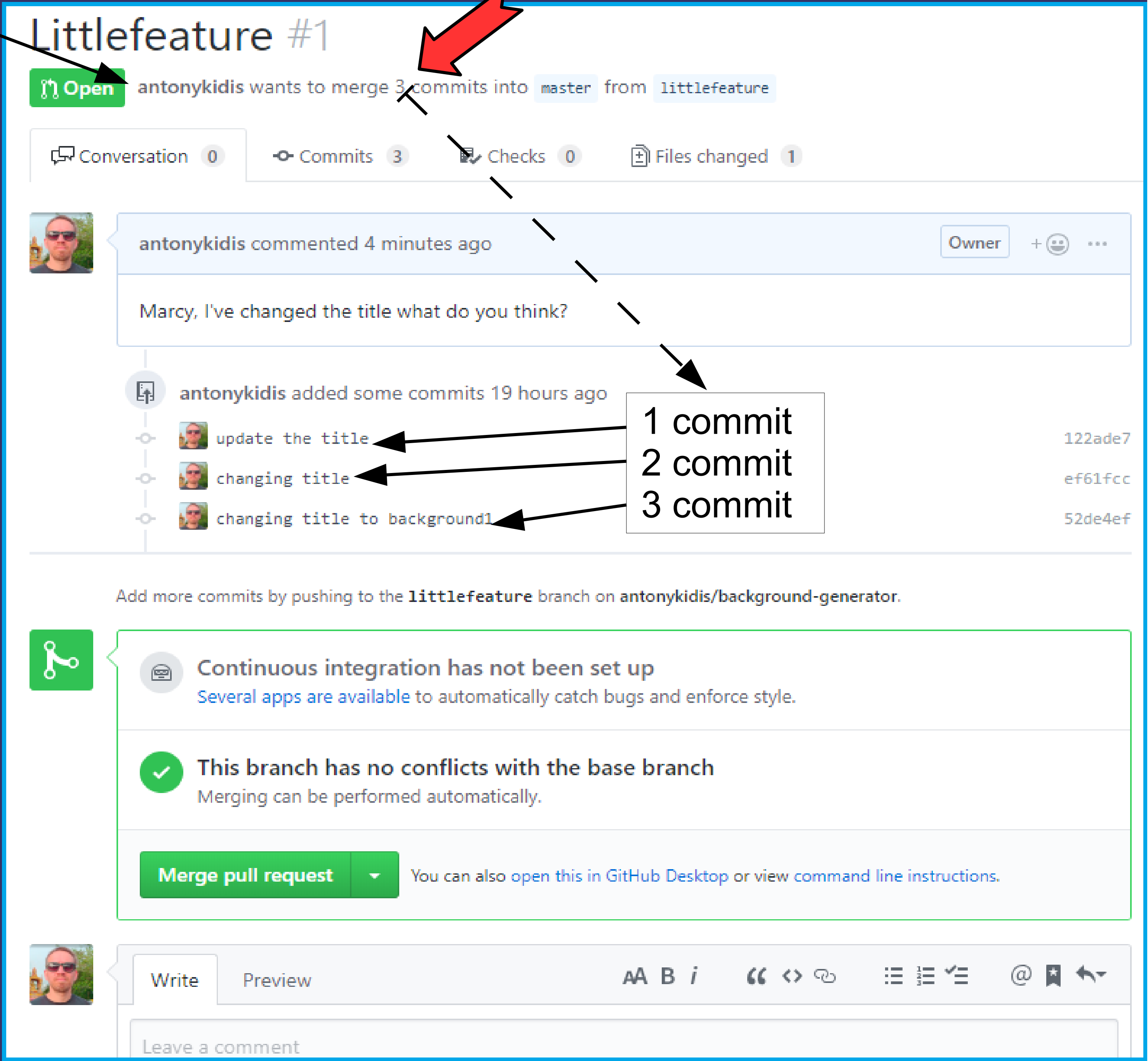
We successfully created a pull request (PR)

As you can see antonykidis(Andrei) wants to merge 3 commits into **master** from **littlefeature**

Hey Marcy I did a little Change on little feature, is it okay for me to Merge it to master?



Let's see what Marcy will probably do..



Littlefeature #1

Open antonykidis wants to merge 3 commits into `master` from `littlefeature`

Conversation 0 · Commits 3 · Checks 0 · Files changed 1

antonykidis commented 4 minutes ago · Owner

Marcy, I've changed the title what do you think?

antonykidis added some commits 19 hours ago

```
update the title                 122ade7
changing title                   ef61fcc
changing title to background1    52de4ef
```

1 commit
2 commit
3 commit

Add more commits by pushing to the **littlefeature** branch on **antonykidis/background-generator**.

Continuous integration has not been set up
Several apps are available to automatically catch bugs and enforce style.

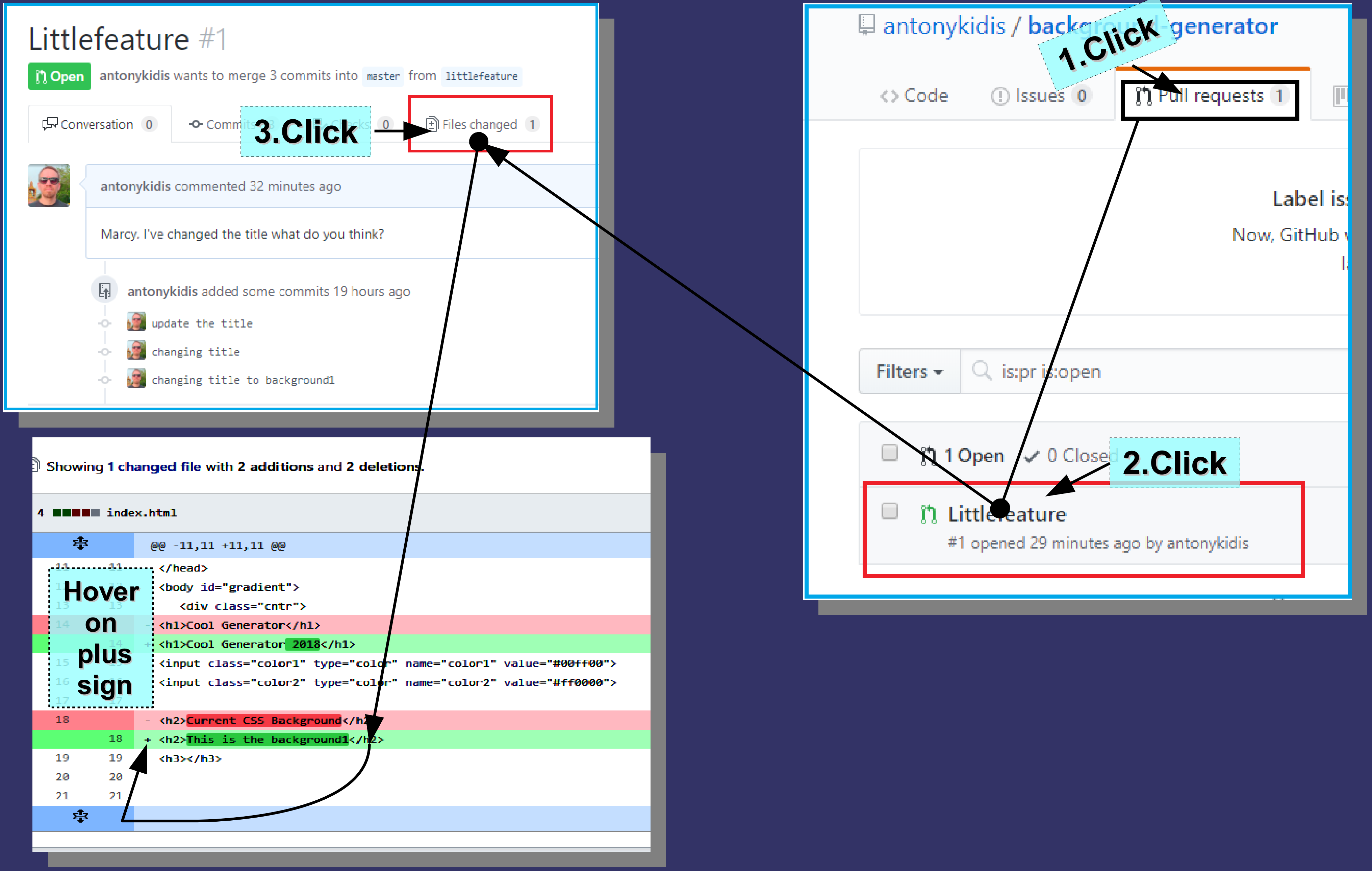
This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request · You can also open this in GitHub Desktop or view command line instructions.

Write · Preview

Leave a comment

Marcy will Open-up a pull request link



antonykidis / background-generator

<> Code ⓘ Issues 0 Pull requests 1

Label is

Now, GitHub

Filters ▾ is:pr is:open

1 Open ✓ 0 Closed

Littlefeature

#1 opened 29 minutes ago by antonykidis

Littlefeature #1

Open antonykidis wants to merge 3 commits into `master` from `littlefeature`

Conversation 0 Commits Checks 0 Files changed 1

antonykidis commented 32 minutes ago

Marcy, I've changed the title what do you think?

antonykidis added some commits 19 hours ago

`update the title`

`changing title`

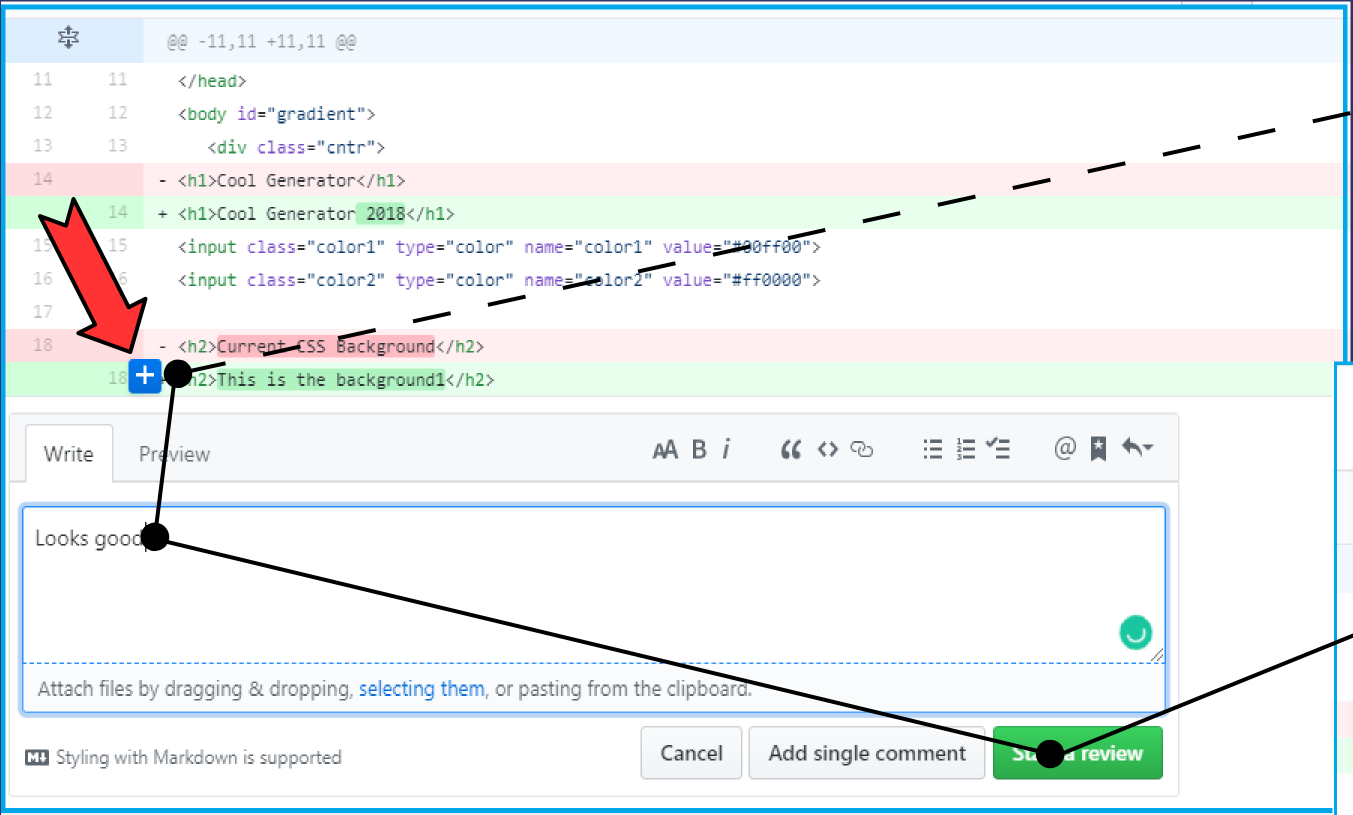
`changing title to background1`

Showing **1 changed file** with **2 additions** and **2 deletions**.

4 🟩🟥🟥🟥⬜ index.html

```
@@ -11,11 +11,11 @@
   </head>
   <body id="gradient">
      <div class="cntr">
 - <h1>Cool Generator</h1>
 + <h1>Cool Generator 2018</h1>
   <input class="color1" type="color" name="color1" value="#00ff00">
   <input class="color2" type="color" name="color2" value="#ff0000">

 - <h2>Current CSS Background</h2>
 + <h2>This is the background1</h2>
   <h3></h3>
```

Hey everything Looks good!
(Click a plus sign)
1.Strat a review, add some comments, submit the Review.
2. Click Review changes, add a comment, and Finally click Submit review
3. Click Merge pull request, then confirm merge.



```
@@ -11,11 +11,11 @@
  </head>
  <body id="gradient">
    <div class="cntr">
- <h1>Cool Generator</h1>
+ <h1>Cool Generator 2018</h1>
  <input class="color1" type="color" name="color1" value="#00ff00">
  <input class="color2" type="color" name="color2" value="#ff0000">

- <h2>Current CSS Background</h2>
+ <h2>This is the background1</h2>
```

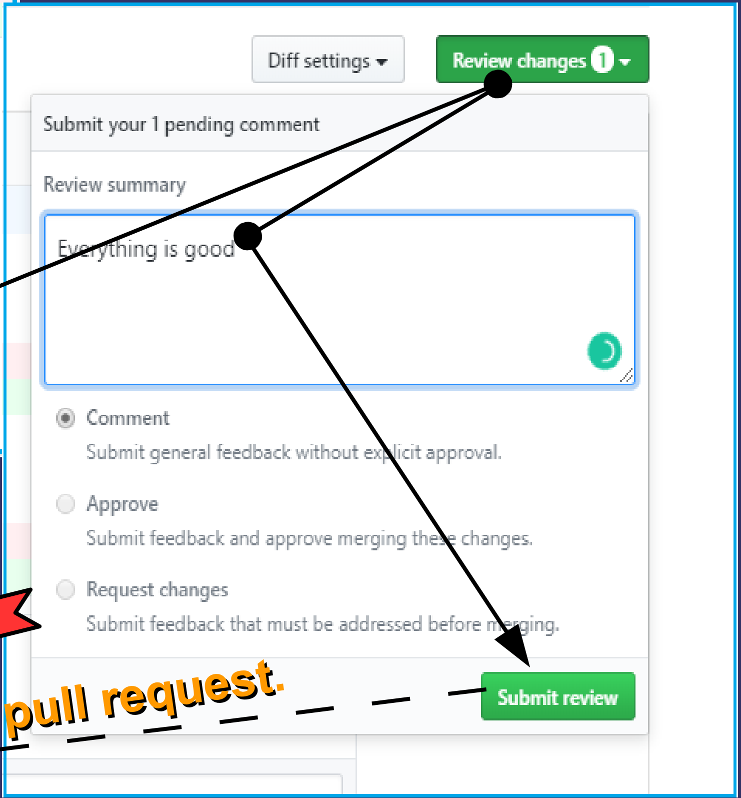
Write | Preview

Looks good

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

Cancel | Add single comment | Start a review



Diff settings | Review changes 1

Submit your 1 pending comment

Review summary

Everything is good

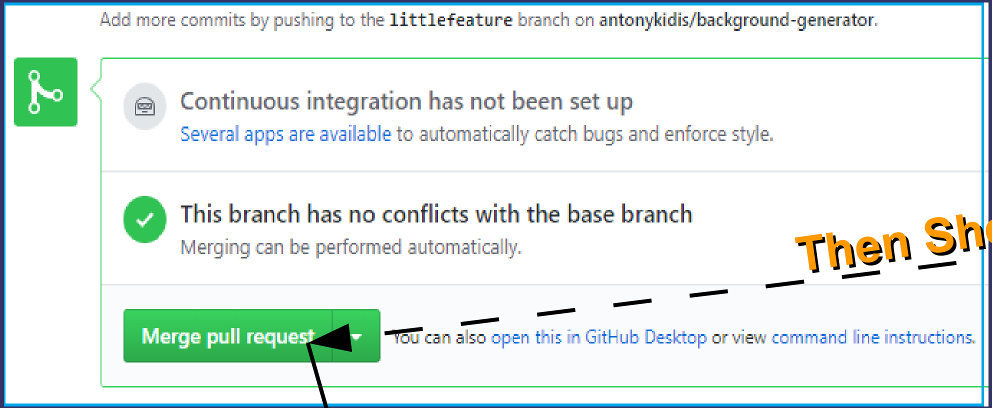
[✓ Comment]
Submit general feedback without explicit approval.

Approve
Submit feedback and approve merging these changes.

Request changes
Submit feedback that must be addressed before merging.

Submit review

Then She will merge a pull request.

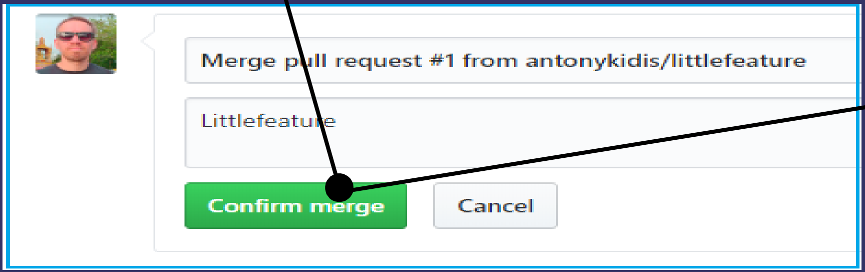


Add more commits by pushing to the **littlefeature** branch on **antonykidis/background-generator**.

Continuous integration has not been set up
Several apps are available to automatically catch bugs and enforce style.

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request | You can also open this in GitHub Desktop or view command line instructions.

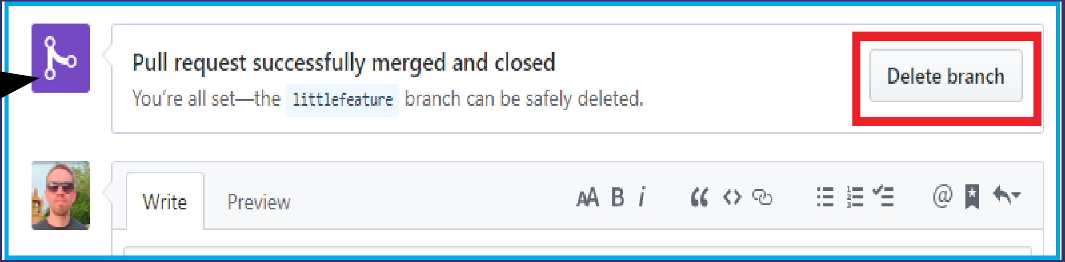


Merge pull request #1 from antonykidis/littlefeature

Littlefeature

Confirm merge | Cancel

Pull request is now merged into the master.
Marcy can now delete littlefeature branch if she want.



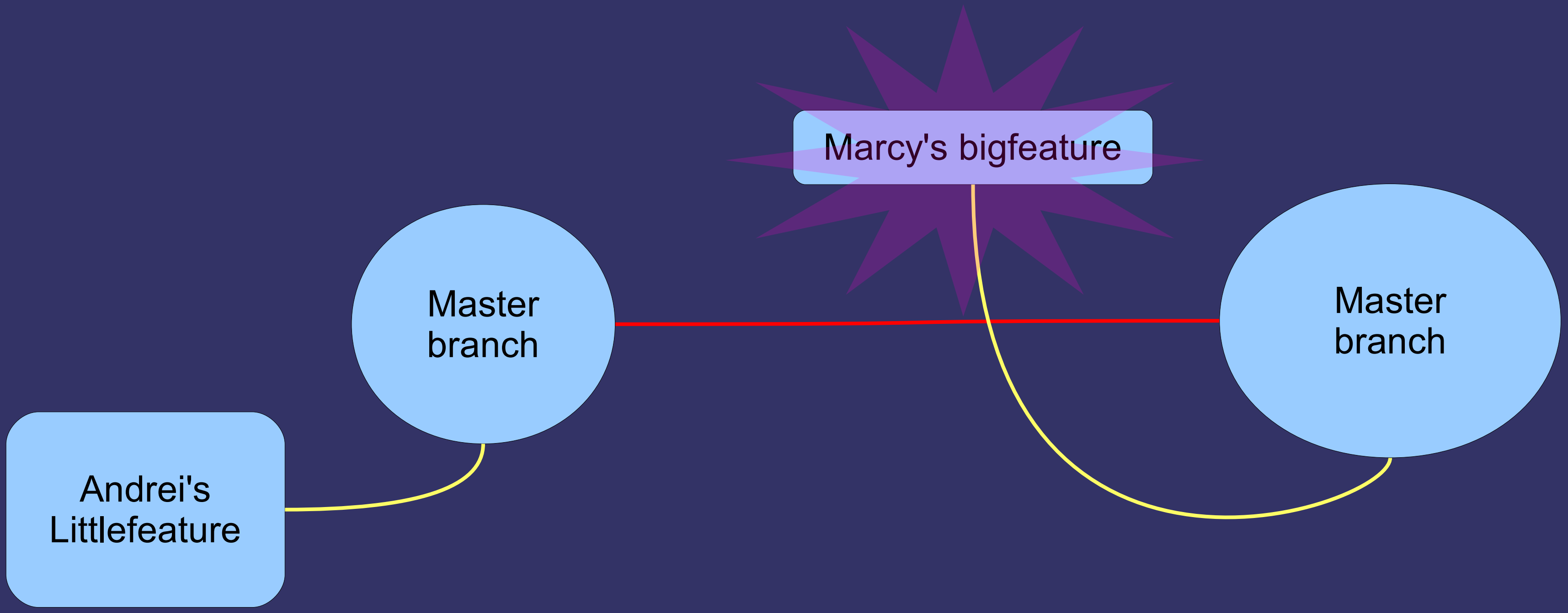
Pull request successfully merged and closed
You're all set—the littlefeature branch can be safely deleted.

Delete branch

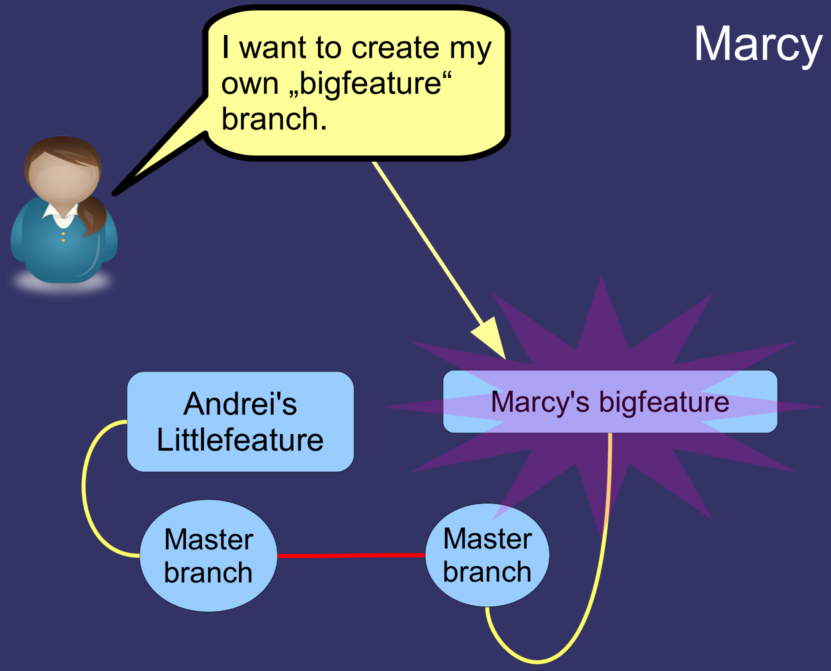
Write | Preview

Merge Conflicts 07:45 (time)

What if Marcy wants to create her own feature!?

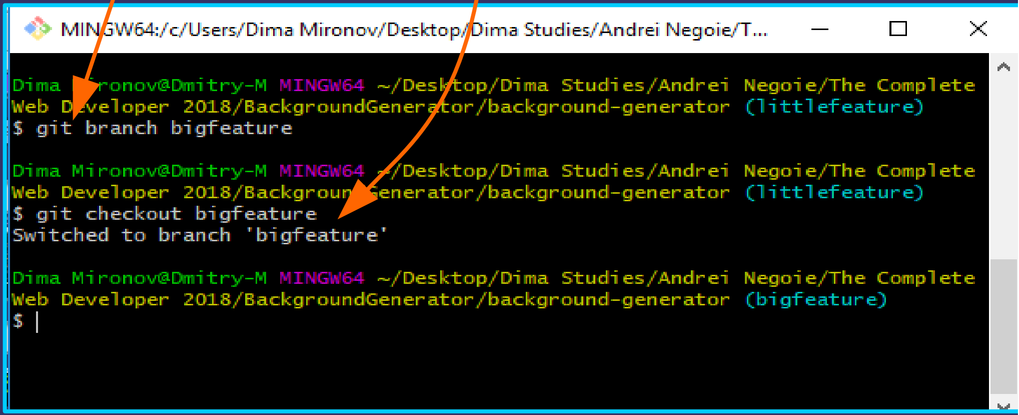


Marcy will create a new branch like this



Step 1

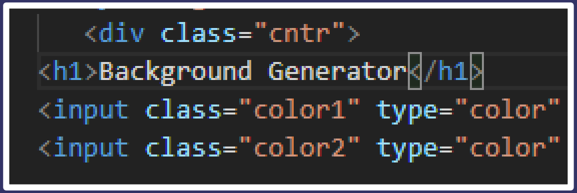
1	git branch bigfeature
2	git checkout bigfeature



```
MINGW64:/c/Users/Dima Mironov/Desktop/Dima Studies/Andrei Negoie/T...

Dima Mironov@Dmitry-M MINGW64 ~/Desktop/Dima Studies/Andrei Negoie/The Complete
Web Developer 2018/BackgroundGenerator/background-generator (littlefeature)
$ git branch bigfeature

Dima Mironov@Dmitry-M MINGW64 ~/Desktop/Dima Studies/Andrei Negoie/The Complete
Web Developer 2018/BackgroundGenerator/background-generator (littlefeature)
$ git checkout bigfeature
Switched to branch 'bigfeature'

Dima Mironov@Dmitry-M MINGW64 ~/Desktop/Dima Studies/Andrei Negoie/The Complete
Web Developer 2018/BackgroundGenerator/background-generator (bigfeature)
$ |
```

Step 2

Marcy changed the title back to the Background generator



```
    <div class="cntr">
<h1>Background Generator</h1>
<input class="color1" type="color"
<input class="color2" type="color"
```

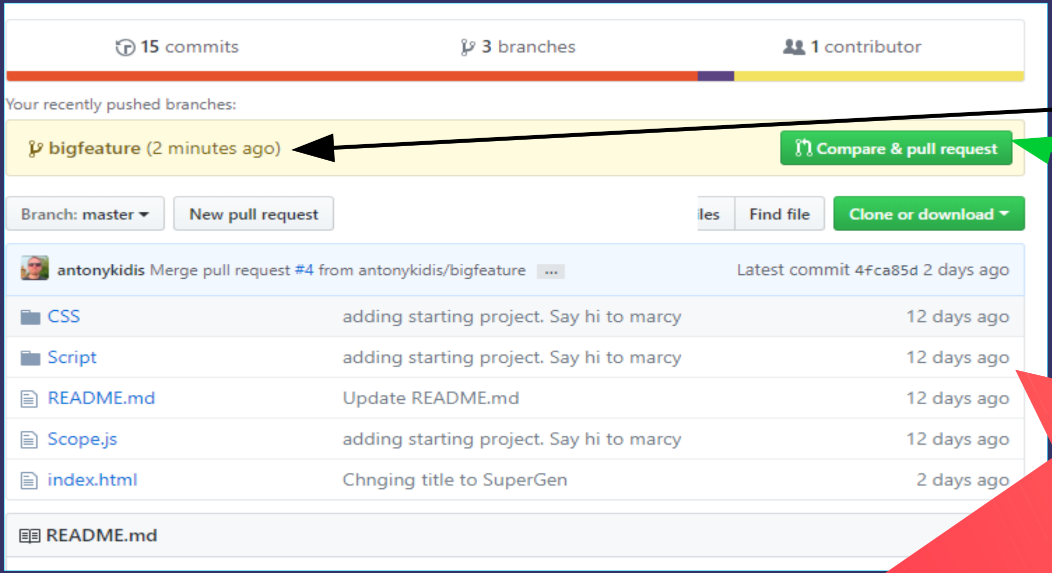
Step 3 She will then do a regular procedure, to save the changes.

1	git add index.html
2	git commit -m "reverting back to old title"
3	git push

If something went wrong, and you want to delete the branch, you can use The following commands. And start all over again.

git branch -d branch_name

git branch -D branch_name



15 commits | 3 branches | 1 contributor

Your recently pushed branches:

bigfeature (2 minutes ago)

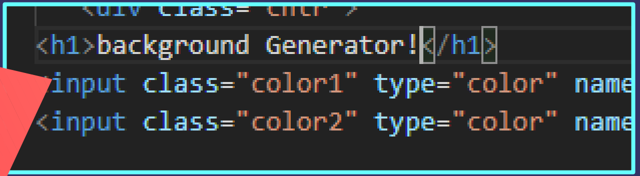
antonykidis Merge pull request #4 from antonykidis/bigfeature … Latest commit 4fca85d 2 days ago

CSS	adding starting project. Say hi to marcy	12 days ago
Script	adding starting project. Say hi to marcy	12 days ago
README.md	Update README.md	12 days ago
Scope.js	adding starting project. Say hi to marcy	12 days ago
index.html	Chnging title to SuperGen	2 days ago

README.md

Now go back to a GitHub. We again see The pull request. **Don't compare and pull request This time!**

Before comparing and pulling a request Marcy decided to add an exclamation mark to the title

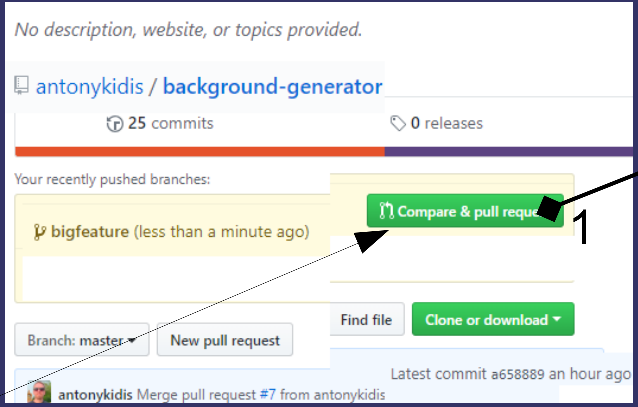


```
<h1>background Generator!</h1>
<input class="color1" type="color" name
<input class="color2" type="color" name
```

Marcy adds exclamation mark

She then Save all...and follow the below steps

1	git add index.html
2	git commit -m “adding exclamation mark“
3	git push



No description, website, or topics provided.

antonykidis / background-generator

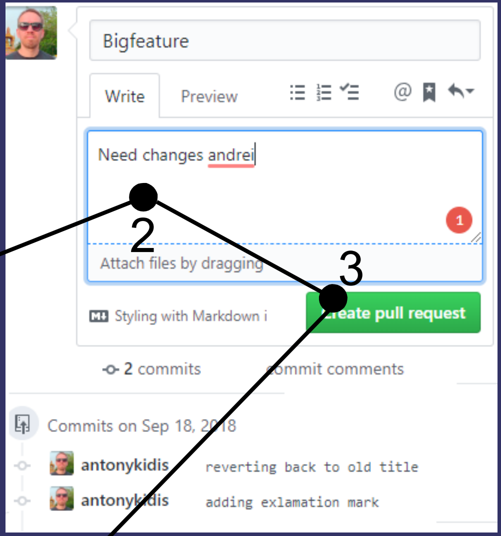
25 commits | 0 releases

Your recently pushed branches:

bigfeature (less than a minute ago)

Latest commit a658889 an hour ago

antonykidis Merge pull request #7 from antonykidis



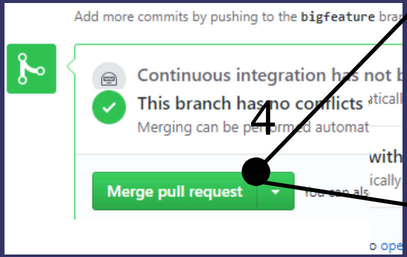
Bigfeature

Need changes andrei

2 commits | commit comments

Commits on Sep 18, 2018

antonykidis	reverting back to old title
antonykidis	adding exclamation mark

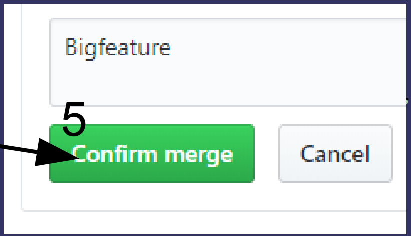


Add more commits by pushing to the bigfeature bra

Continuous integration has not b

This branch has no conflicts

Merging can be performed automat

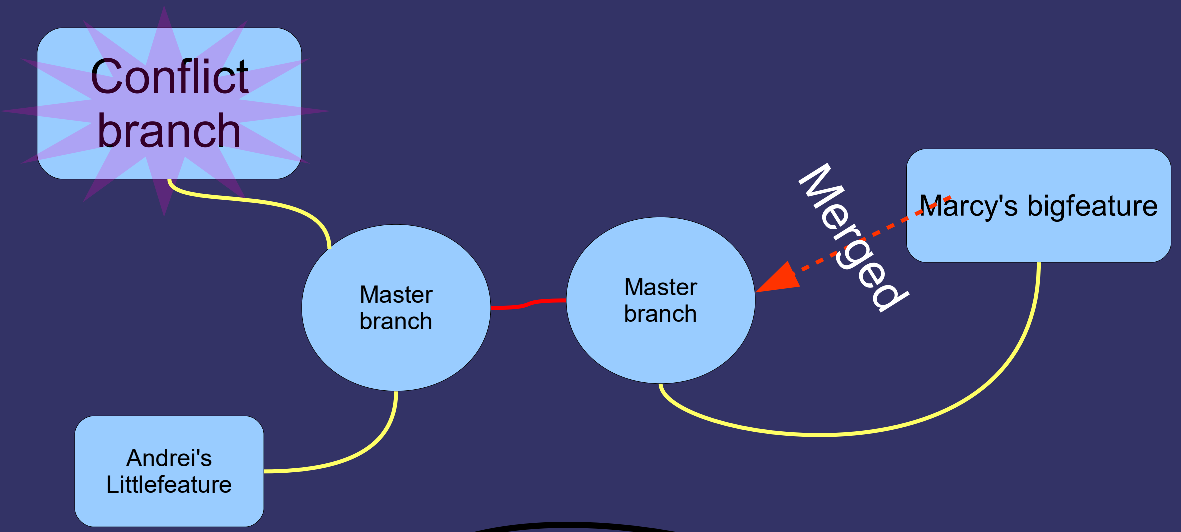


Bigfeature

Let's go back to GitHub.
So now we have 2 commits
1. is the background generator
2. and the background generator! with the exclamation mark

1. Click Compare and pull request button,
2. Add a comment.
3. Click Create Pull Request button.
4. Merge pull request button.
5. Finally click Confirm merge button.

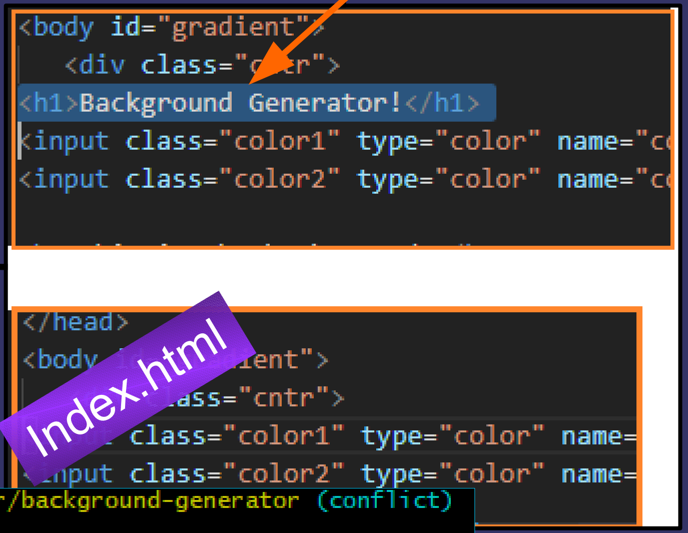
Marcy just merged a pull request from a bigfeature branch. Meanwhile, Andrei decided to create a new branch too



In the terminal window, Andrei wil enter the following commands:

1. git branch conflict
2. git checkout conflict

3. Andrei decided to completly remove the background generator title
4. save all

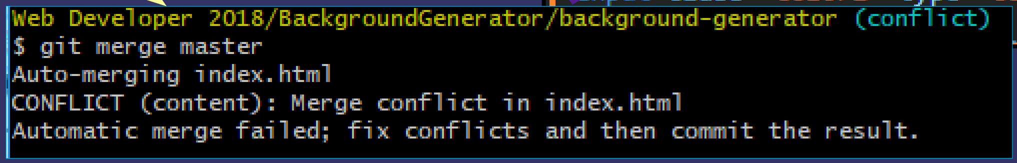


```
<body id="gradient">
   <div class="cntr">
<h1>Background Generator!</h1>
<input class="color1" type="color" name="co
<input class="color2" type="color" name="co
```

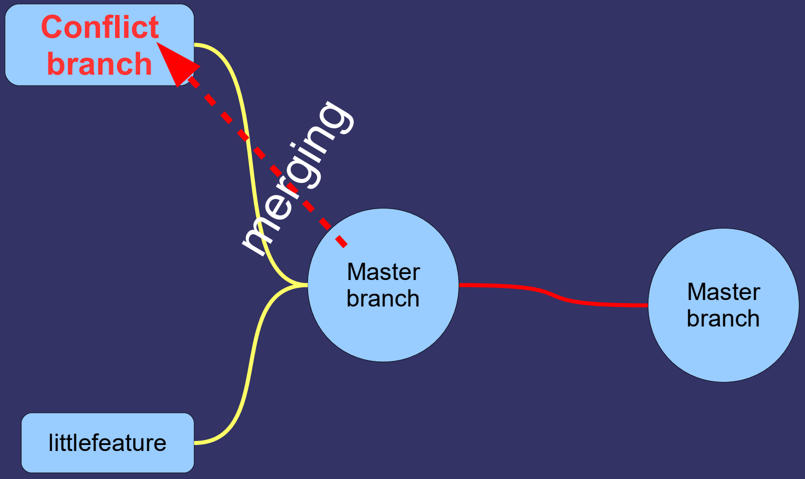
```
</head>
<body id="gradient">
       class="cntr">
    class="color1" type="color" name=
  input class="color2" type="color" name=
```

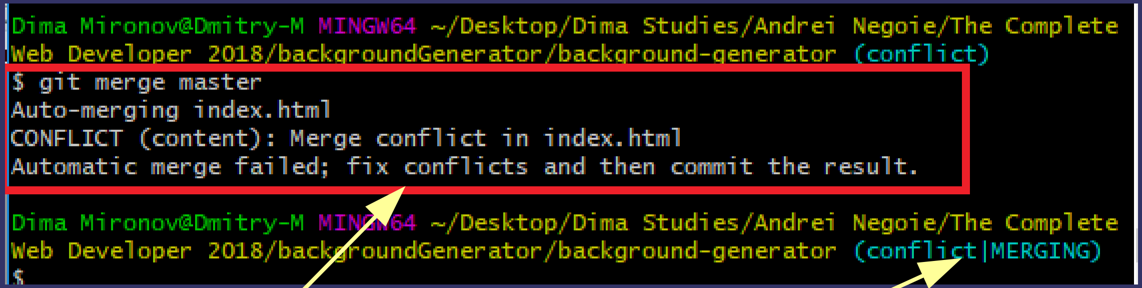
He will then go back to a terminal And enter the following commands:

5. git add index.html
6. git commit -m“deleting title“
7. git checkout master
8. git pull
9. git checkout conflict
10 .merge master into the conflict branch.
11. type git merge master



```
Web Developer 2018/BackgroundGenerator/background-generator (conflict)
$ git merge master
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

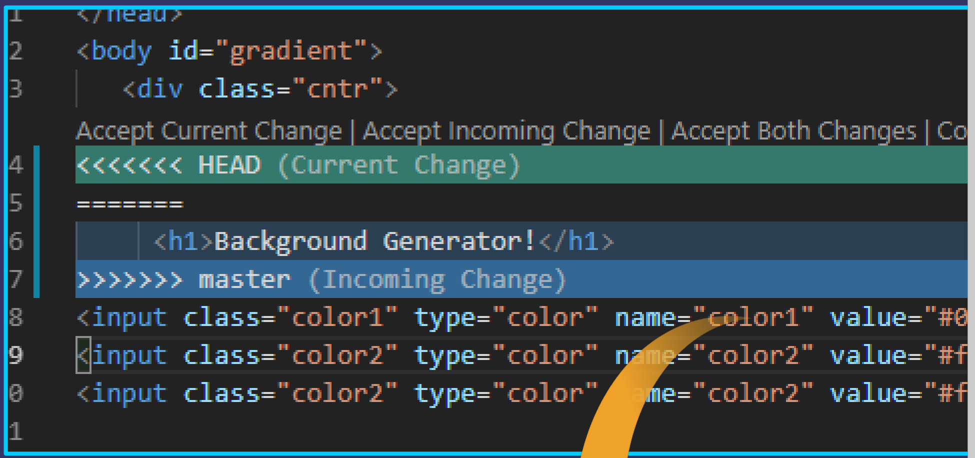




```
Dima Mironov@Dmitry-M MINGW64 ~/Desktop/Dima Studies/Andrei Negoie/The Complete
Web Developer 2018/backgroundGenerator/background-generator (conflict)
$ git merge master
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.

Dima Mironov@Dmitry-M MINGW64 ~/Desktop/Dima Studies/Andrei Negoie/The Complete
Web Developer 2018/backgroundGenerator/background-generator (conflict|MERGING)
$
```

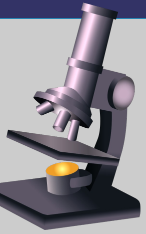
Important to read!

6. By now you should recieve the following error
7. If you Open a (Visual Studio Code/Sublime)
You sould see the the following output:



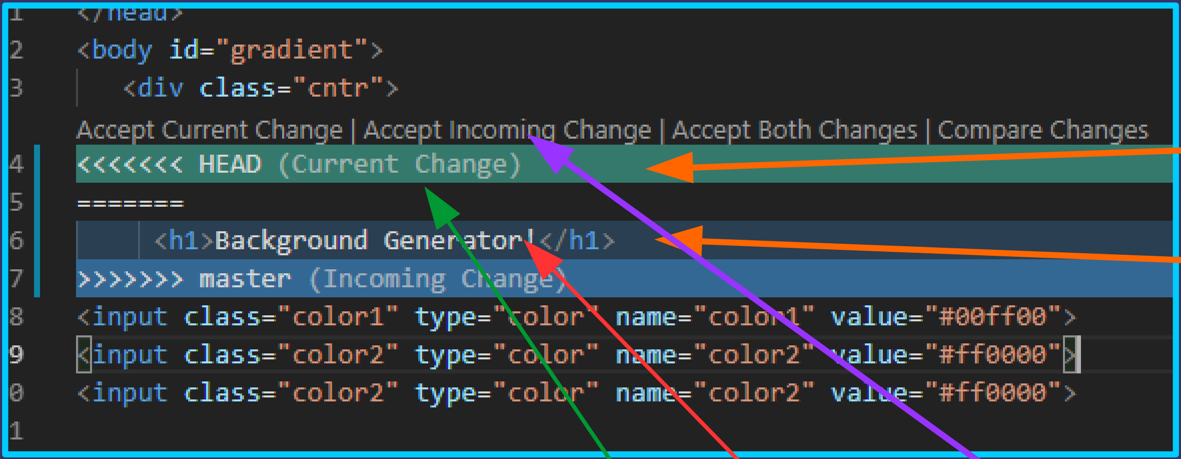
```
</head>
<body id="gradient">
  <div class="cntr">
Accept Current Change | Accept Incoming Change | Accept Both Changes | Co
<<<<<<< HEAD (Current Change)
=======
    <h1>Background Generator!</h1>
>>>>>>> master (Incoming Change)
<input class="color1" type="color" name="color1" value="#0
<input class="color2" type="color" name="color2" value="#f
<input class="color2" type="color" name="color2" value="#f
```

Let's understand the problem.



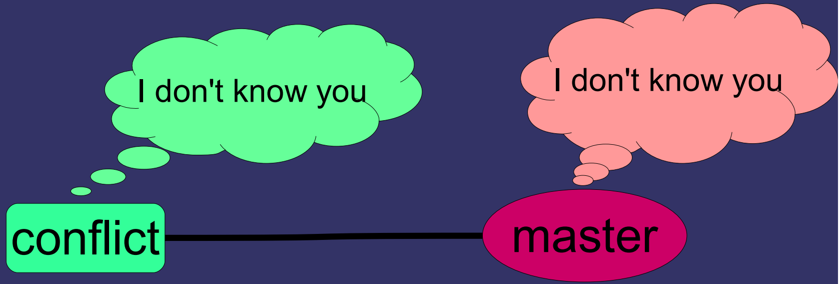
1. We Made a conflict brach.
2. Then we removed the title from HTML.
3. Then Andrei did **git add .** Then **git commit -m „deleting title“**
But before pushing these changes back to a GitHub, he decided to Merge whatever in the master, into his conflict branch. Yes, he could push the title-less index.html back to a GitHub. But he came up with another idea: He said let's see what is in master these days....
Let's see what is inside the master now. Let's pull it intothe conflict just to be updated. Yes sure, he didn't know that master is now different from a conflict, because he did not pulled nothing for a while (just an example)

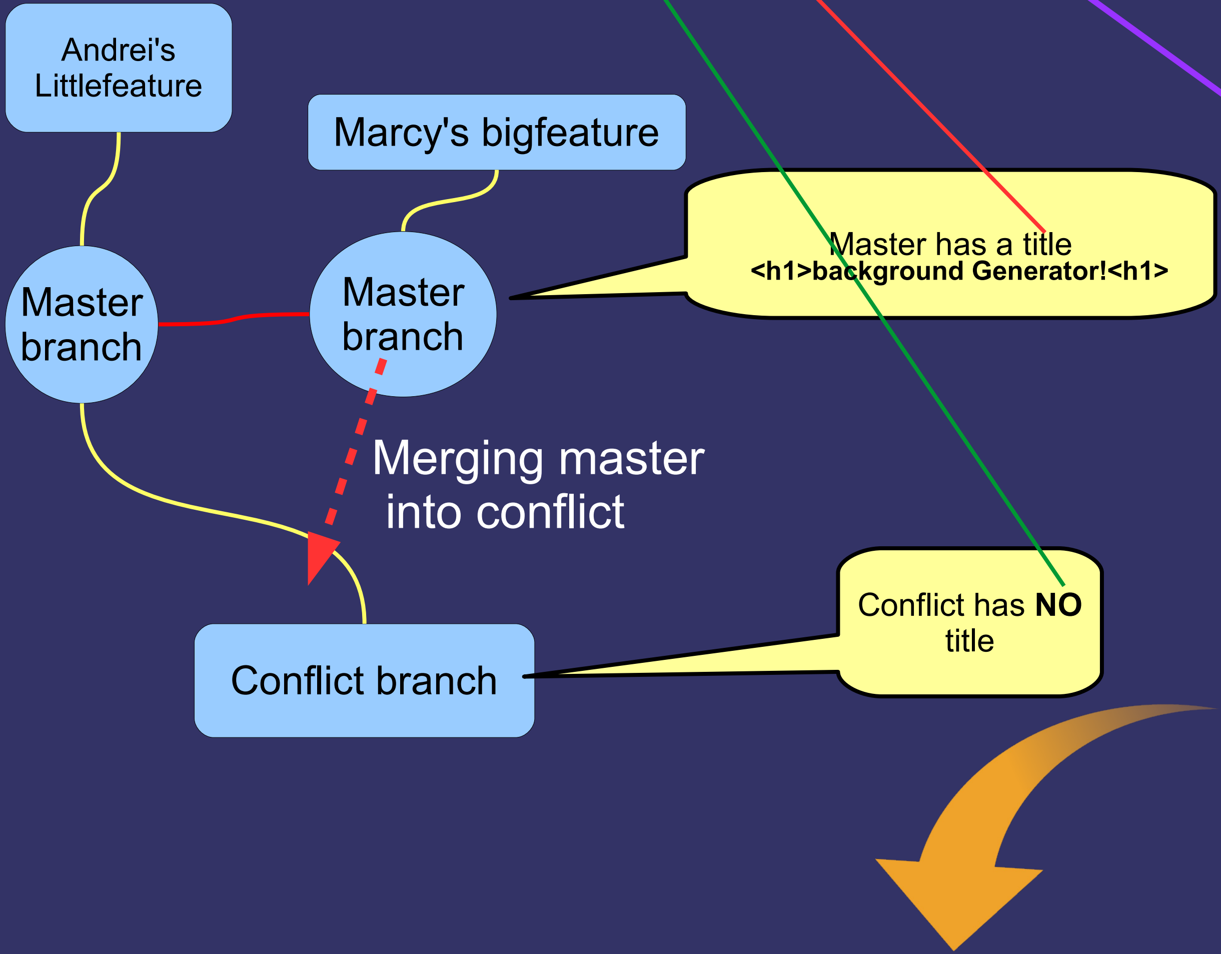
4. So we got back into the master branch.
5. We then pulled master's **current version** (to a local computer)
6. Then we merged the master into the conflict. Just to be sure the conflict is synchronized with the master. But we have invoked an error. Because master contains a title, but a conflict branch doesn't have a title.
So we get an intuitive user frendly notification from VS code or sublime.
Saying that there is a changes to be made, and your files different from master.
So!?...Where is the catch? Why we all need this??? you say....
Okay, You worked on some branch, so once in a while you want to update Your branch with a latest changes. So it is easier to merge a master into Your **OWN** branch, rather then download files manually, copy all of its contents, then Paste all of these into your own branch. **You see the difference??? :-)**
What if a master branch contains thousands of files? You will get nuts until you copy them one by one into your branch. So We simply merging the master, and it does all the dirty work for us. without worrying we missed something.



```
</head>
<body id="gradient">
  <div class="cntr">
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
<<<<<<< HEAD (Current Change)
=======
    <h1>Background Generator!</h1>
>>>>>>> master (Incoming Change)
<input class="color1" type="color" name="color1" value="#00ff00">
<input class="color2" type="color" name="color2" value="#ff0000">
<input class="color2" type="color" name="color2" value="#ff0000">
```

<<<Head This is conflict brach
>>>master This is master branch





Andrei's Littlefeature
Marcy's bigfeature
Master branch
Master branch
Master has a title <h1>background Generator!<h1>
Merging master into conflict
Conflict branch
Conflict has **NO** title

Ask Marcy first if she want to leave the title.
If yes, we can accept the Incoming changes by clicking the Accept incoming link,Or delete manually the following:

<<<<<<<<<<< Head Current change
==================
>>>>>>>>>>> Master Incoming change

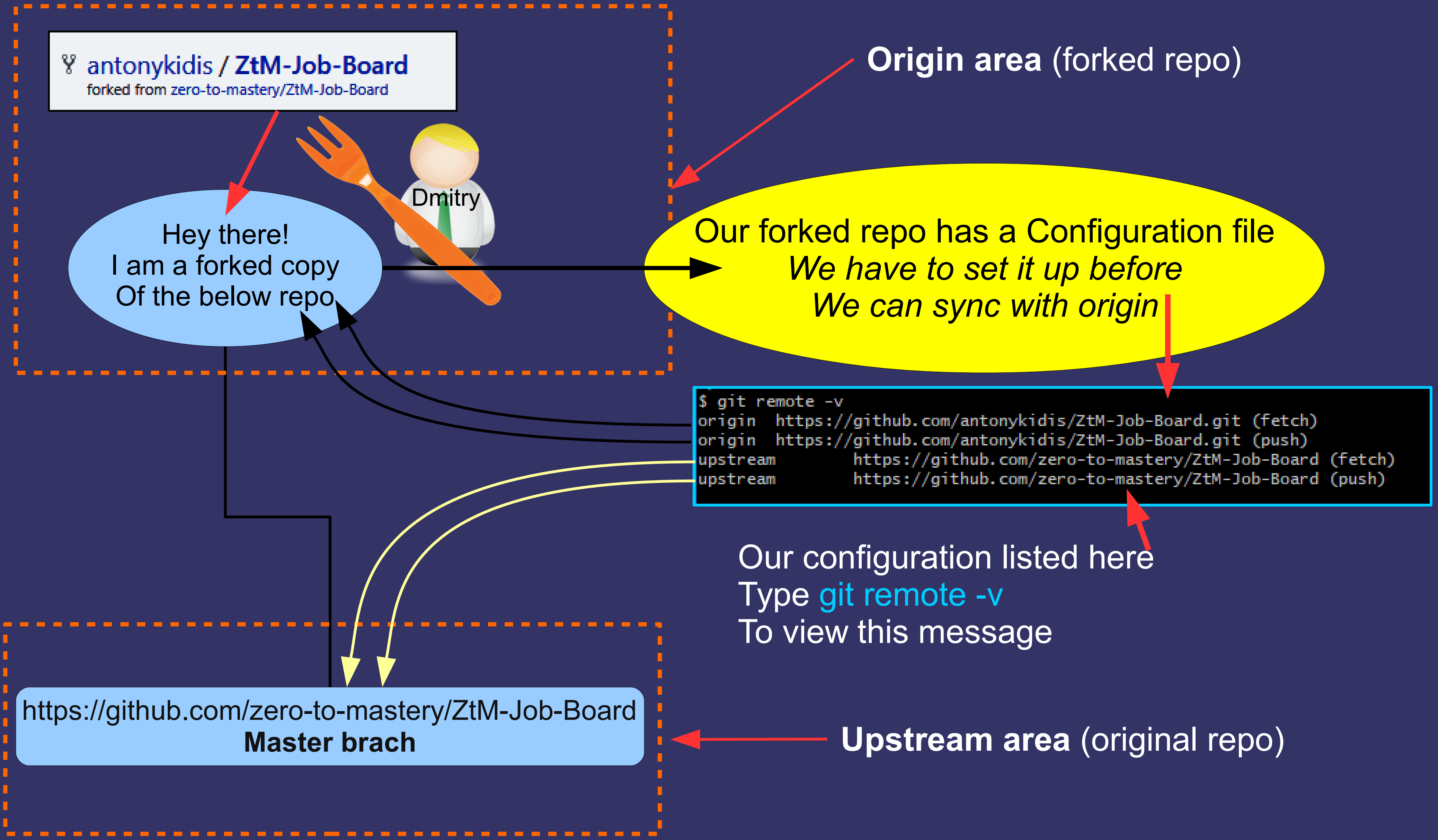
Leaving the
<h1>background Generator</h1>
In place
If you edit the file manualy

Finally use the following commands

1	git add index.html
2	git commit -m “Okay we leaving the title as is“
3	git push

In the next section We'll see
how to keep your fork up to date

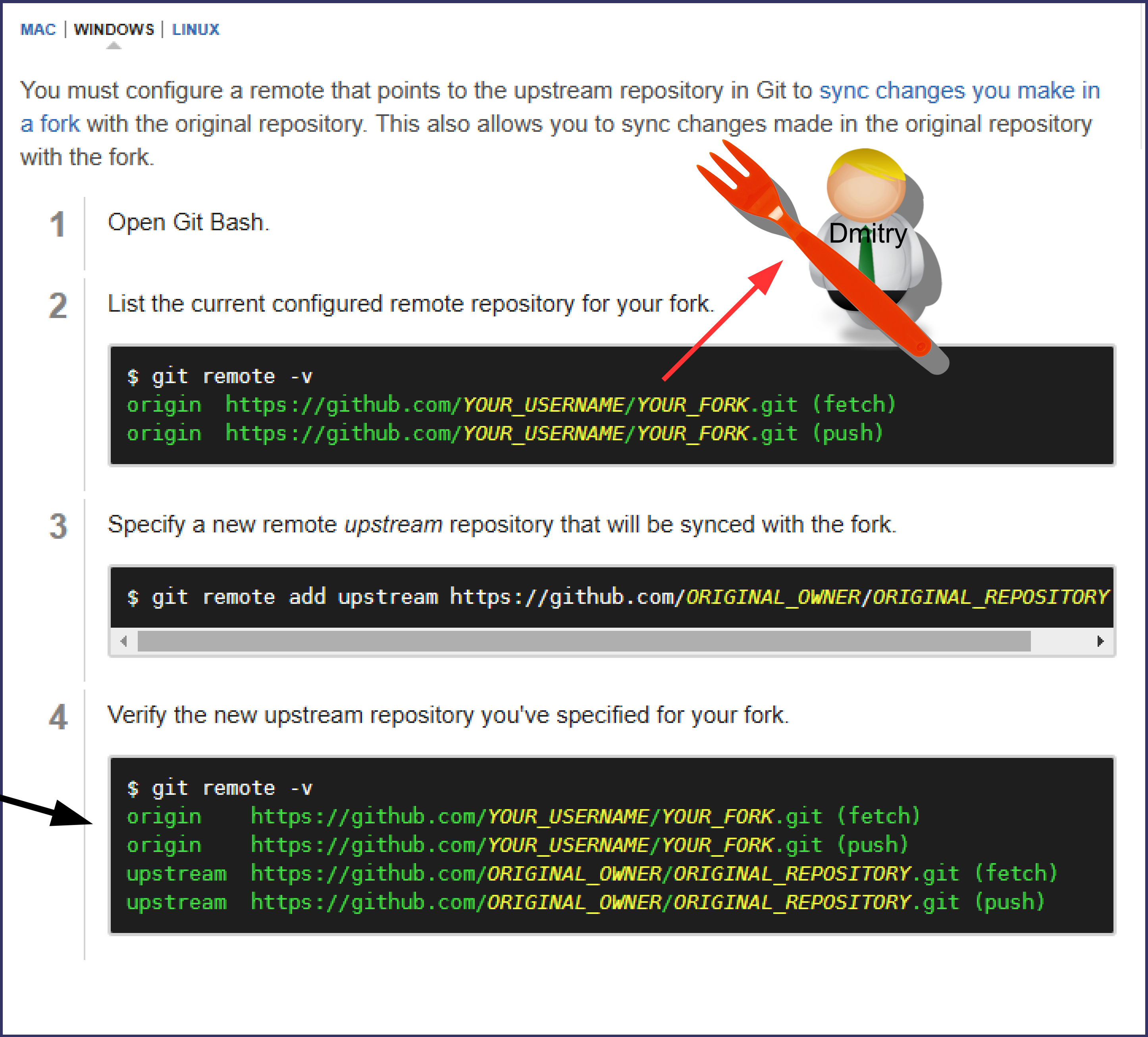
keep your fork up to date


antonykidis / ZtM-Job-Board
forked from zero-to-mastery/ZtM-Job-Board
Dmitry
Hey there!
I am a forked copy
Of the below repo
Origin area (forked repo)
Our forked repo has a Configuration file
We have to set it up before
We can sync with origin
Our configuration listed here
Type git remote -v
To view this message
https://github.com/zero-to-mastery/ZtM-Job-Board
Master brach
Upstream area (original repo)

```
$ git remote -v
origin  https://github.com/antonykidis/ZtM-Job-Board.git (fetch)
origin  https://github.com/antonykidis/ZtM-Job-Board.git (push)
upstream        https://github.com/zero-to-mastery/ZtM-Job-Board (fetch)
upstream        https://github.com/zero-to-mastery/ZtM-Job-Board (push)
```

#Step1 Configuring a remote for a fork

Fetch == pull
Push == push



MAC | WINDOWS | LINUX

You must configure a remote that points to the upstream repository in Git to sync changes you make in a fork with the original repository. This also allows you to sync changes made in the original repository with the fork.

1 Open Git Bash.

2 List the current configured remote repository for your fork.

```
$ git remote -v
origin  https://github.com/YOUR_USERNAME/YOUR_FORK.git (fetch)
origin  https://github.com/YOUR_USERNAME/YOUR_FORK.git (push)
```

3 Specify a new remote *upstream* repository that will be synced with the fork.

```
$ git remote add upstream https://github.com/ORIGINAL_OWNER/ORIGINAL_REPOSITORY
```

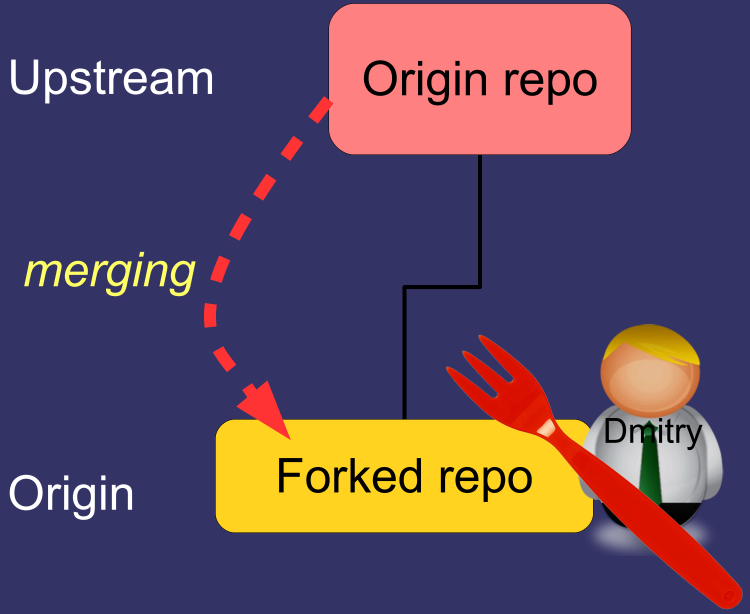
4 Verify the new upstream repository you've specified for your fork.

```
$ git remote -v
origin    https://github.com/YOUR_USERNAME/YOUR_FORK.git (fetch)
origin    https://github.com/YOUR_USERNAME/YOUR_FORK.git (push)
upstream  https://github.com/ORIGINAL_OWNER/ORIGINAL_REPOSITORY.git (fetch)
upstream  https://github.com/ORIGINAL_OWNER/ORIGINAL_REPOSITORY.git (push)
```

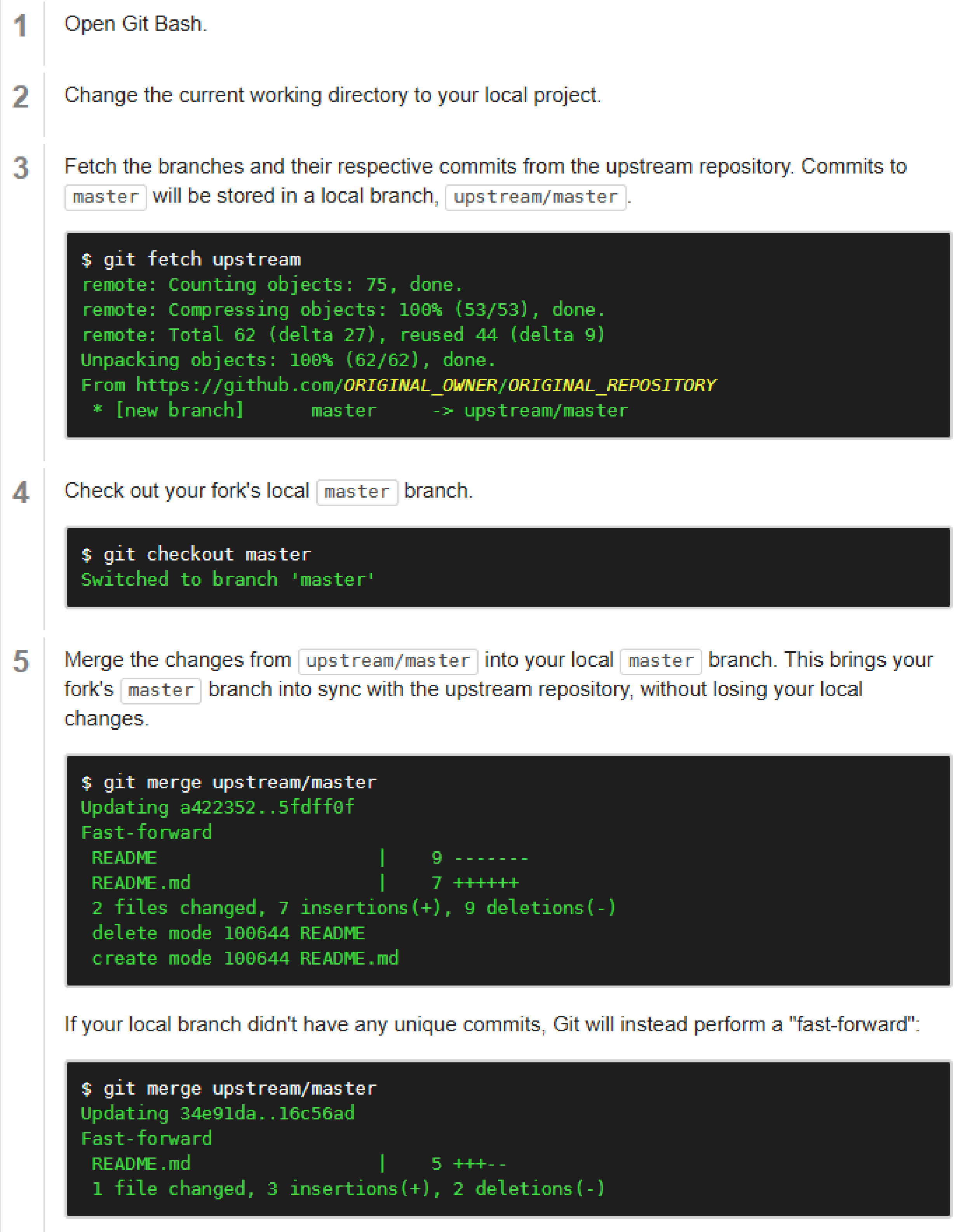
We all set up now

But we still have to pull
From upstream, and
Merge with origin.

step 2
Let's Pull, and merge



Done!
We now synced with the original repository



1 Open Git Bash.

2 Change the current working directory to your local project.

3 Fetch the branches and their respective commits from the upstream repository. Commits to `master` will be stored in a local branch, `upstream/master`.

```
$ git fetch upstream
remote: Counting objects: 75, done.
remote: Compressing objects: 100% (53/53), done.
remote: Total 62 (delta 27), reused 44 (delta 9)
Unpacking objects: 100% (62/62), done.
From https://github.com/ORIGINAL_OWNER/ORIGINAL_REPOSITORY
 * [new branch]      master     -> upstream/master
```

4 Check out your fork's local `master` branch.

```
$ git checkout master
Switched to branch 'master'
```

5 Merge the changes from `upstream/master` into your local `master` branch. This brings your fork's `master` branch into sync with the upstream repository, without losing your local changes.

```
$ git merge upstream/master
Updating a422352..5fdff0f
Fast-forward
 README                    |    9 -------
 README.md                 |    7 ++++++
 2 files changed, 7 insertions(+), 9 deletions(-)
 delete mode 100644 README
 create mode 100644 README.md
```

If your local branch didn't have any unique commits, Git will instead perform a "fast-forward":

```
$ git merge upstream/master
Updating 34e91da..16c56ad
Fast-forward
 README.md                 |    5 +++--
 1 file changed, 3 insertions(+), 2 deletions(-)
```

What the hell Was that?
Okay let's try „explanation for dummies“
As I had a real hard time to understand this the easy way

Without
My config file
I am a real dumbass

My forked repo

.Git/Config.file

Tell me how should I
Connect a forked repo
with GitHub's origin
repo, and keep it up to
date?

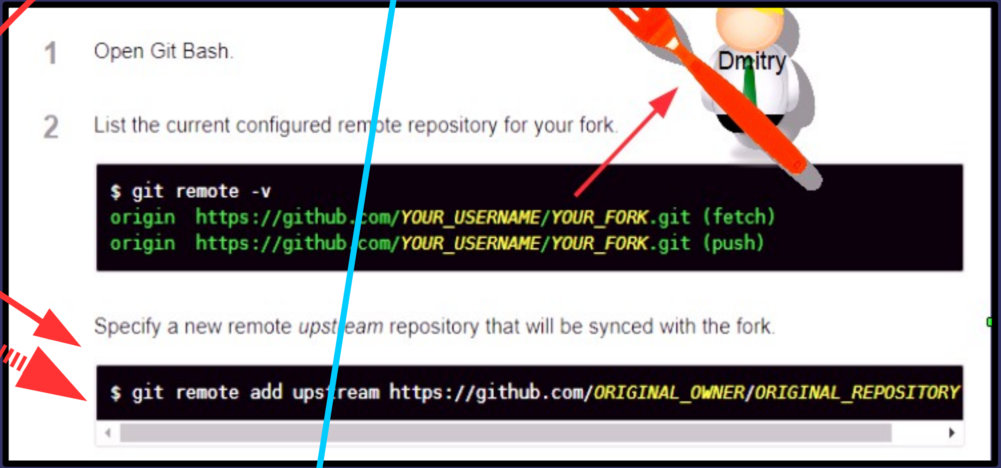
Awaiting your
instructions here.....

Git remote add upstream....

I am a „Configuration“ file.
I can do some tricks.
But tell me what should I do.

Okay seems like you want
To setup your fork, so it could
Comunicate with upstream repository.

Aha...I can now sync all the changes
You made in original repo with the fork.



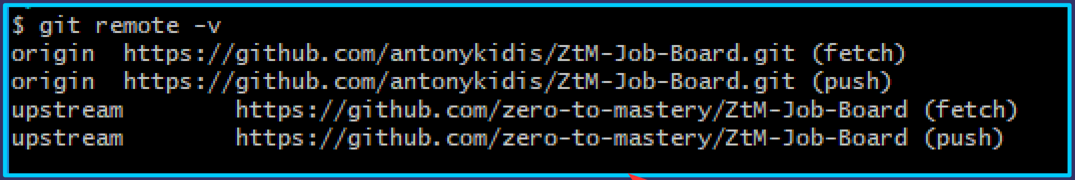
1 Open Git Bash.

2 List the current configured remote repository for your fork.

```
$ git remote -v
origin  https://github.com/YOUR_USERNAME/YOUR_FORK.git (fetch)
origin  https://github.com/YOUR_USERNAME/YOUR_FORK.git (push)
```

Specify a new remote upstream repository that will be synced with the fork.

```
$ git remote add upstream https://github.com/ORIGINAL_OWNER/ORIGINAL_REPOSITORY
```



```
$ git remote -v
origin  https://github.com/antonykidis/ZtM-Job-Board.git (fetch)
origin  https://github.com/antonykidis/ZtM-Job-Board.git (push)
upstream        https://github.com/zero-to-mastery/ZtM-Job-Board (fetch)
upstream        https://github.com/zero-to-mastery/ZtM-Job-Board (push)
```

I forgot to tell you that
I have a special configuration file
In my **.git folder**
Non of your friends thold you of
Its existence?
….I Kill you!

GitHub

Some repo is here

A simple representation of config file
As seen in gitbash terminal.

To sync a fork with the original repo you have to
set up comunication rules between these two in the
Forked repository configuration file (in .git) folder

TO BE CONTINUED
IN PART 3 (Contribute to the open source)