

The complete Web Developer in 2018

Git and GitHub Guide

Part-2

Branching

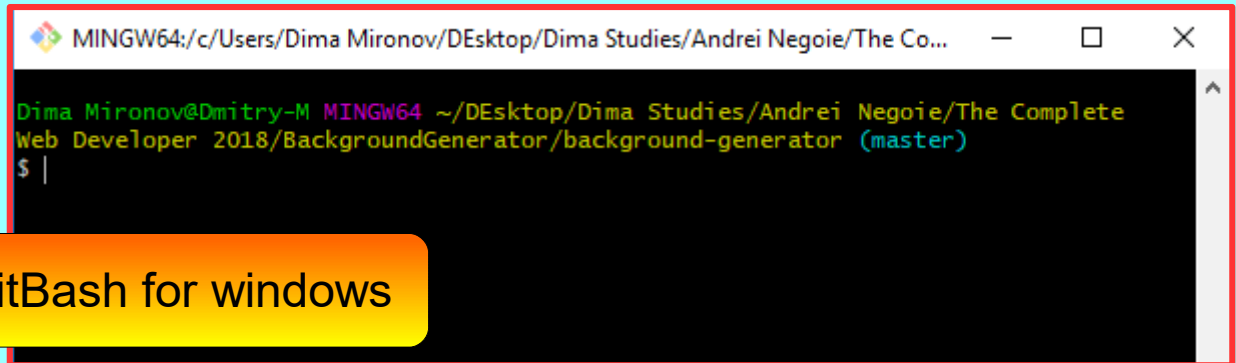
Info Page

For all the windows users.

You can use Gitbash terminal, or a Visual Studio Code terminal.

Download it here <https://git-scm.com/downloads>

GitBash for windows

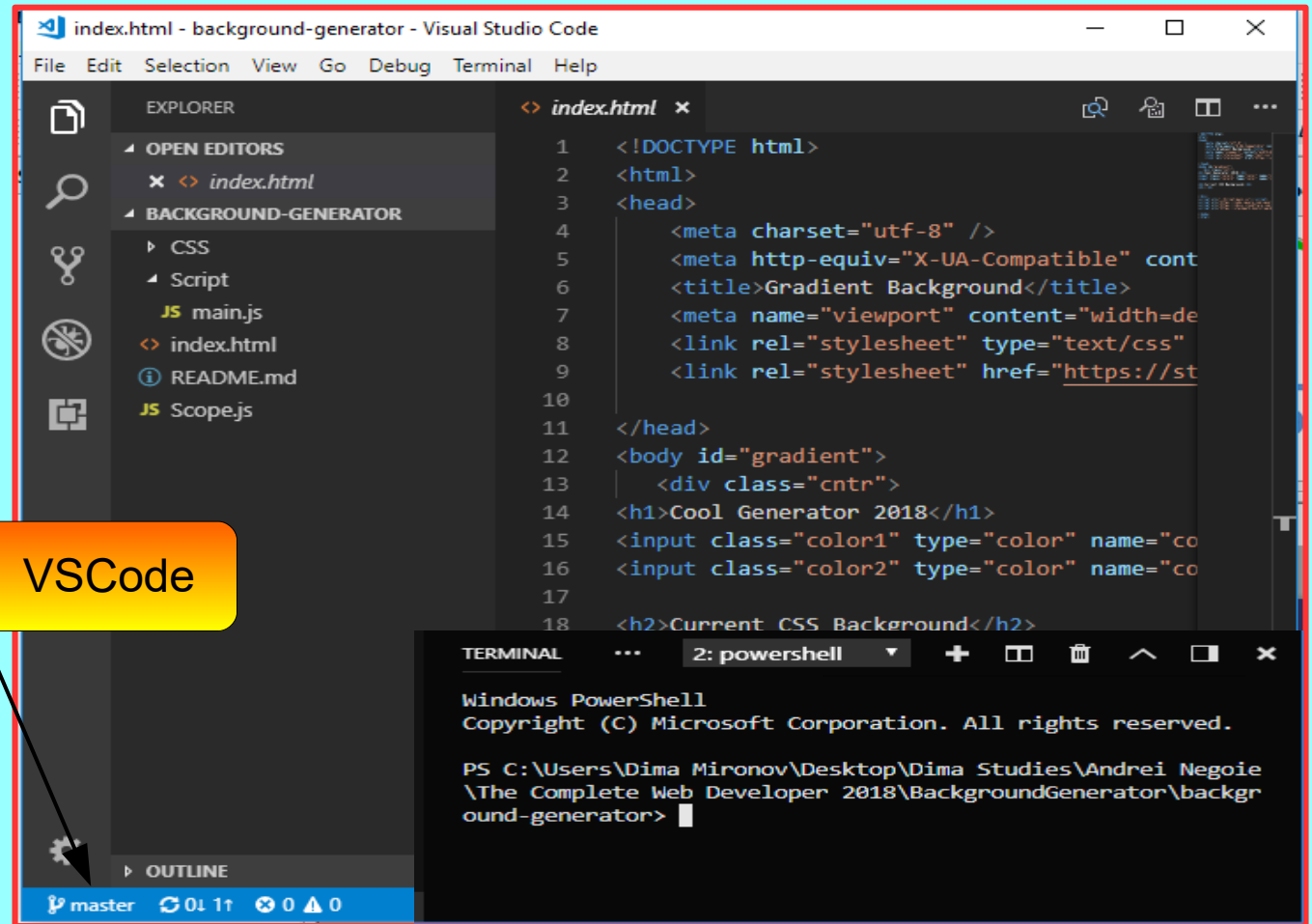


Visual Studio Code

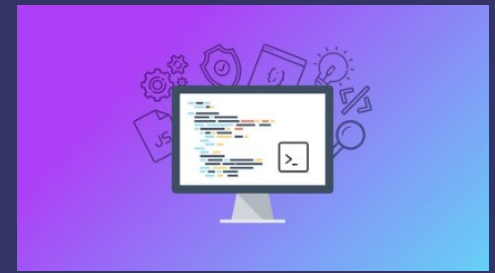
VS Code display a branch name
It Doesn't display a branch name
In the PowerShell.

I suggest you stick with the Git Bash
terminal.

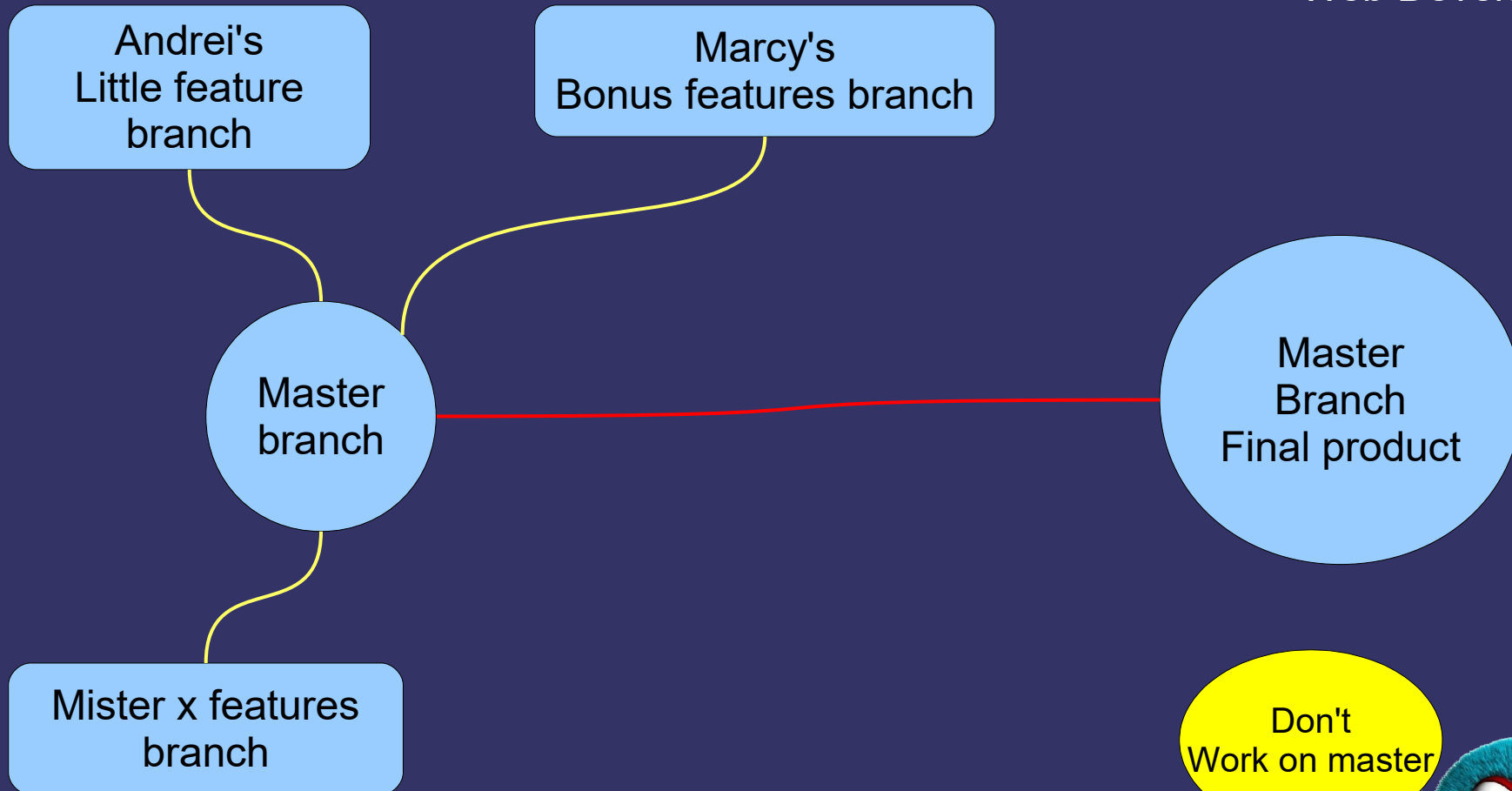
VSCode



Refers to Section 15, video #145
Git and GitHub Guide Part-2 Branching



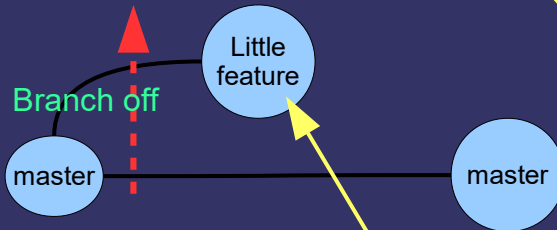
The complete
Web Developer in 2018



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

```
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)  
$
```

4. We successfully switched to a new branch

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 the background</h2>
3. Save all, then do the following procedure...
4. `git add index.html`
5. `git commit -m"changing title"`
6. `git push`

```
12 <body id="gradient">
13 |   <div class="cntr">
14 |     <h1>Cool Generator 2018</h1>
15 |     <input class="color1" type="color" name="color1" value="#00f
16 |     <input class="color2" type="color" name="color2" value="#ff6
17 |
18 |     <h2>This is the background</h2>
19 |     <h3></h3>
20 |
21 |
```

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`

```
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

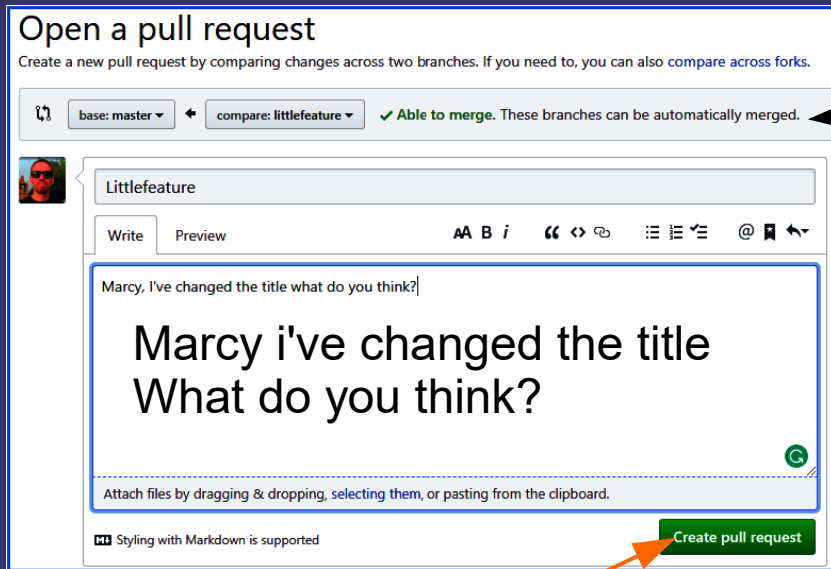
    git push --set-upstream origin littlefeature

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>

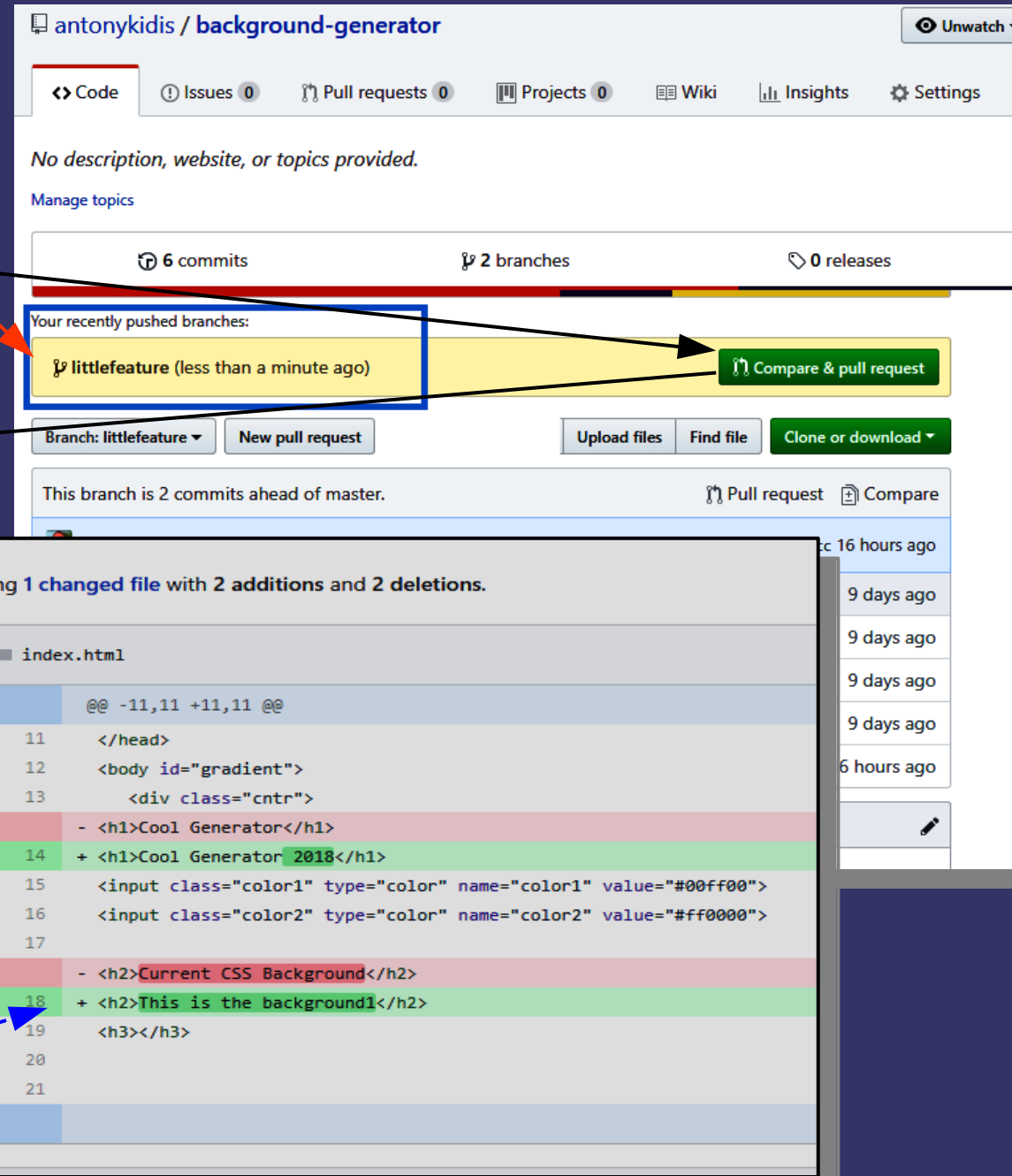
1. We now see a yellow notification
You recently pushed branches

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



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

We've made a change to index.html!
Marcy will decide approve 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
Do something about it



 Compare & pull request

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.

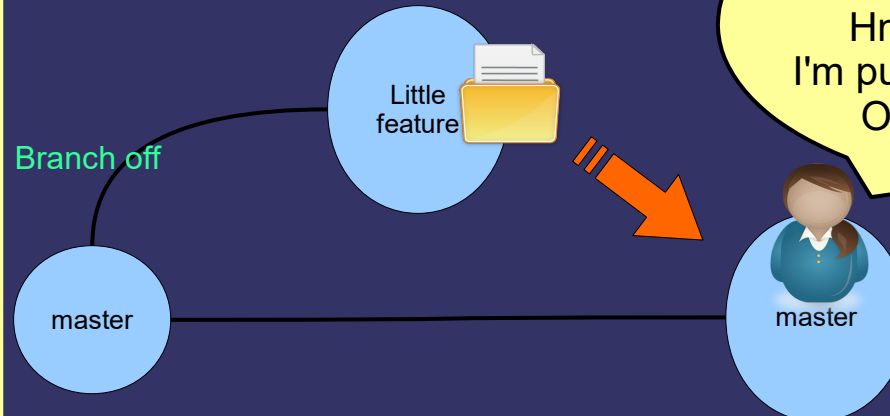
Marcy is an administrator and she will open your pushed files on a littlefeature branch.
She will compare them to the previous one, and decide whether to prove these changes or not

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.

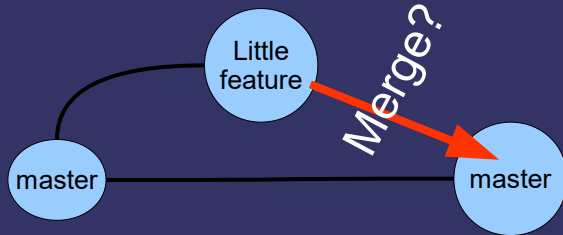


Hi Andrei!
Let's compare your changes
Hmm..okay looks good!
I'm pulling this into the master
Okay it's merged now!

We successfully created a pull request (PR)

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

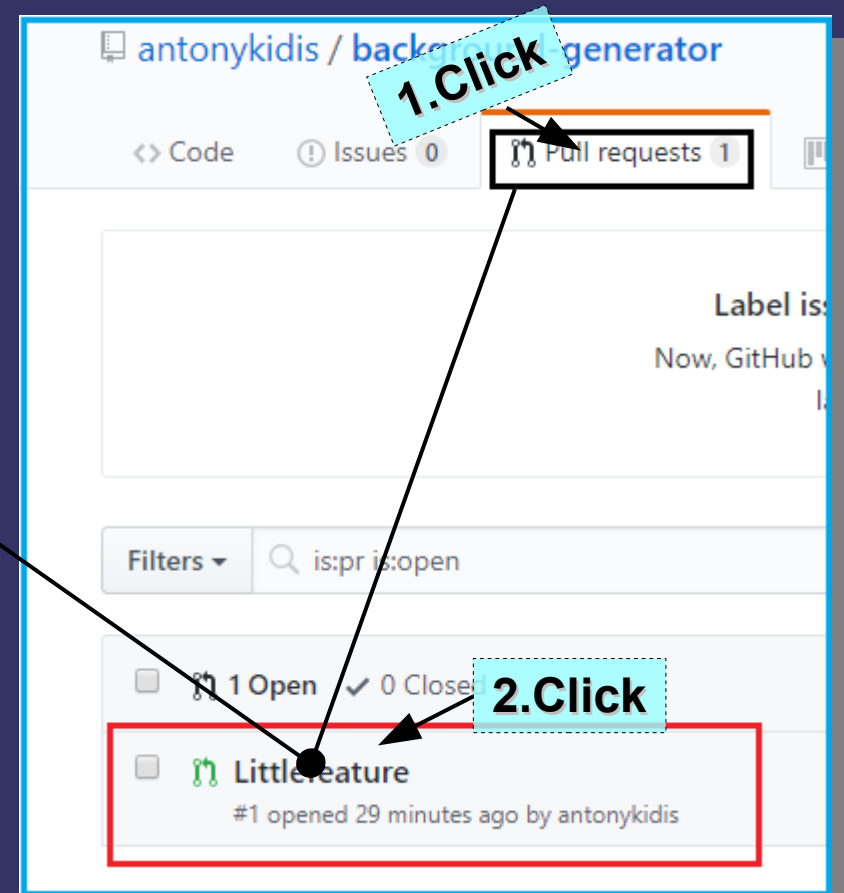
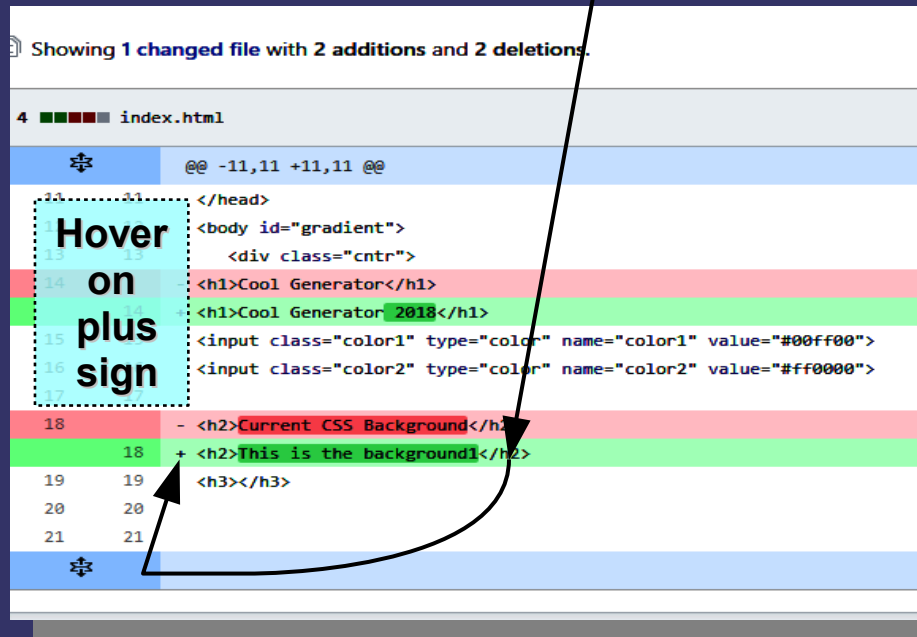
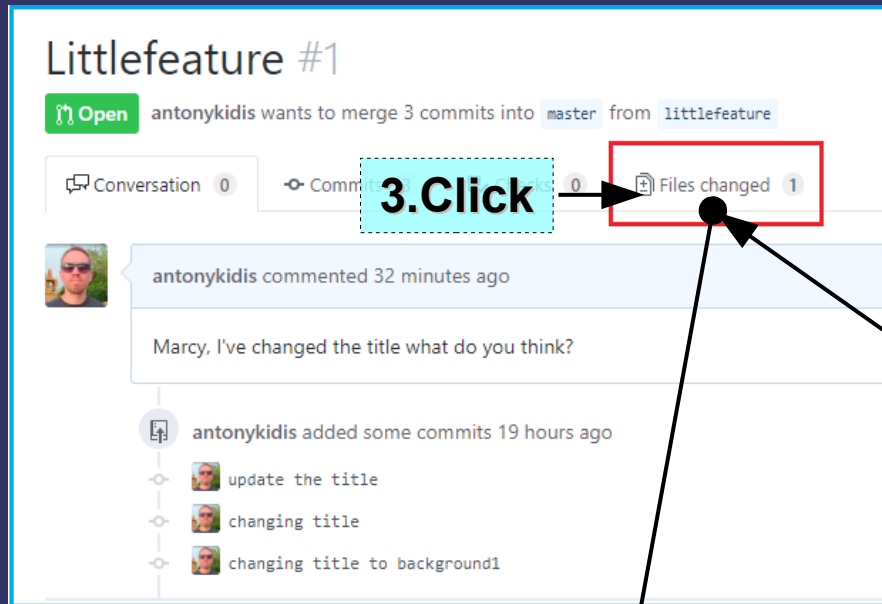
Hey Marcy I did a little Change on little feature, is it okay for me to Merge it to master?



Let's see what Marcy will probably do..

The screenshot shows a GitHub Pull Request titled "Littlefeature #1". The header indicates that antonykidis wants to merge 3 commits into master from littlefeature. Below the header, there are tabs for Conversation (0), Commits (3), Checks (0), and Files changed (1). A comment from antonykidis, posted 4 minutes ago, asks Marcy for feedback on a title change. Below the comment, a list of commits is shown, with a box highlighting the first three commits: "1 commit", "2 commit", and "3 commit". The commits are: "update the title" (122ade7), "changing title" (ef61fcc), and "changing title to background1" (52de4ef). A message at the bottom states: "Add more commits by pushing to the littlefeature branch on antonykidis/background-generator." Below this, a green box contains a message: "Continuous integration has not been set up. Several apps are available to automatically catch bugs and enforce style." Below that, a green checkmark indicates: "This branch has no conflicts with the base branch. Merging can be performed automatically." At the bottom, there is a green button labeled "Merge pull request" and a dropdown menu. A footer section shows a "Write" tab, a "Preview" tab, and a "Leave a comment" input field.

Marcy will Open-up a pull request link



Hey everything Looks good!

(Click a plus sign)

1. Start 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 @@

```
11 11 </head>
12 12 <body id="gradient">
13 13 <div class="cntr">
14 - <h1>Cool Generator</h1>
14 + <h1>Cool Generator 2018</h1>
15 15 <input class="color1" type="color" name="color1" value="#00ff00">
16 16 <input class="color2" type="color" name="color2" value="#ff0000">
17 17
18 - <h2>Current CSS Background</h2>
18 + <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 Submit 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.

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

Merge pull request #1 from antonykidis/littlefeature

Littlefeature

Confirm merge Cancel

Pull request successfully merged and closed

You're all set—the **littlefeature** branch can be safely deleted.

Delete branch

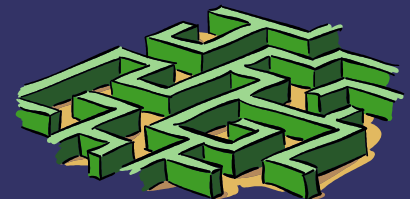
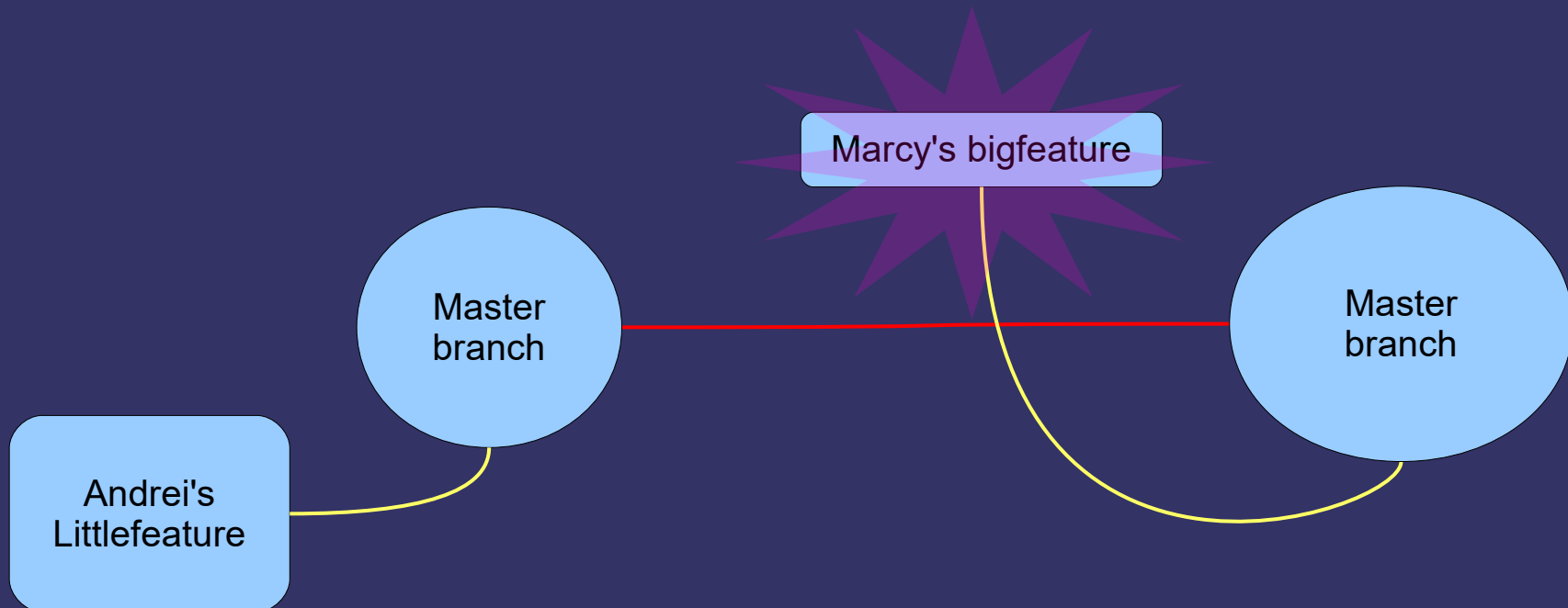
Write Preview

Styling with Markdown is supported

Merge Conflicts 07:45 (time)



What if Marcy wants to create her own feature!?





I want to create my own „bigfeature“ branch.

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

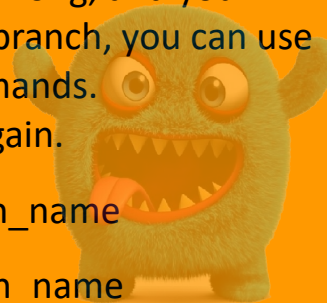
She will then do a regular procedure, to save the changes.

- 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`



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

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

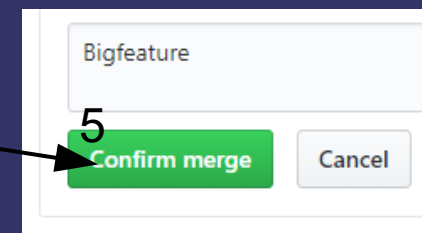
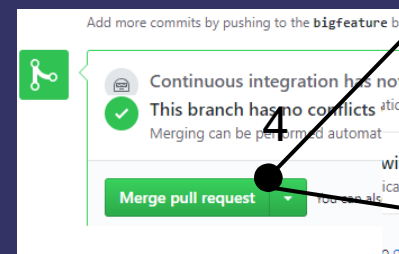
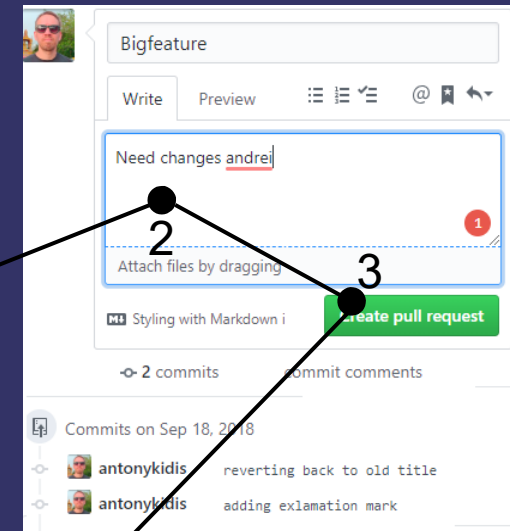
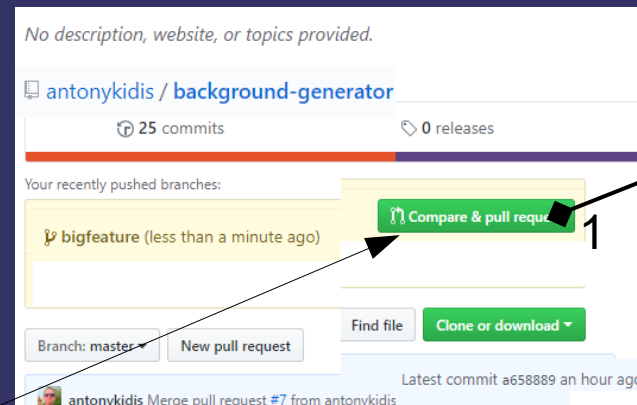
Marcy adds exclamation mark

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

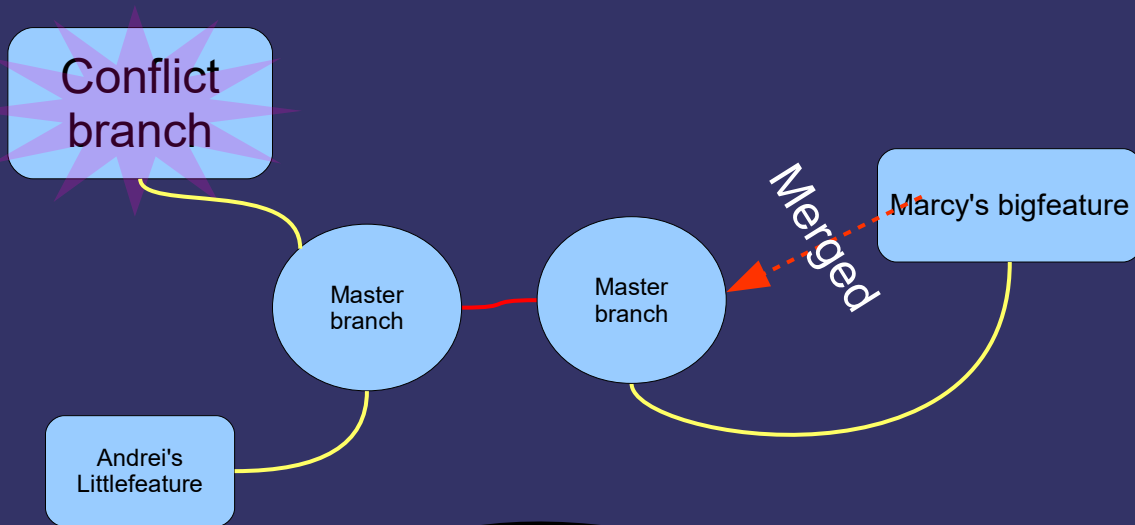
- 1 git add index.html
- 2 git commit -m "adding exclamation mark"
- 3 git push

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 will enter the following commands:

1. `git branch conflict`
2. `git checkout conflict`

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

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

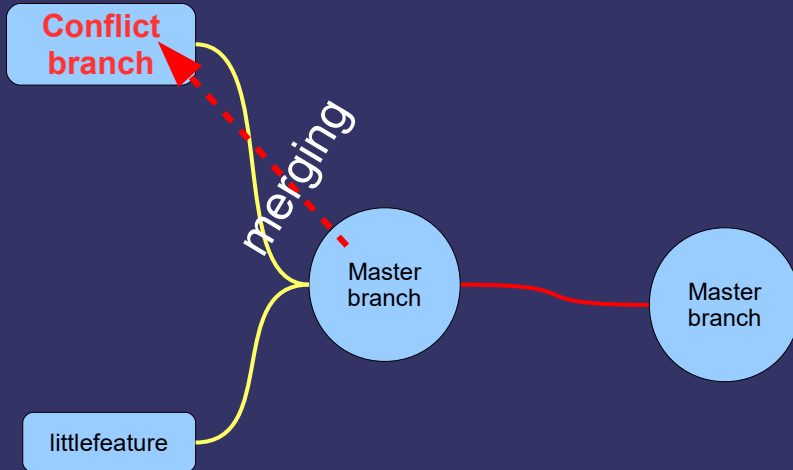
5. `git add index.html`
6. `git commit -m"deleting title"`
7. `git checkout master`
8. `git pull`
9. `git checkout conflict`
10. `.merge master into the conflict branch.`
11. type `git merge master`

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

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

Index.html

```
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.

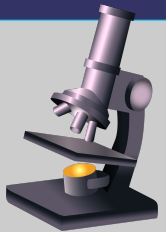
Dima Mironov@Dmitry-M MINGW64 ~/Desktop/Dima Studies/Andrei Negoie/The Complete
Web Developer 2018/backgroundGenerator/background-generator (conflict|MERGING)
$
  
```

Important to read!

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

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 friendly 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.

```

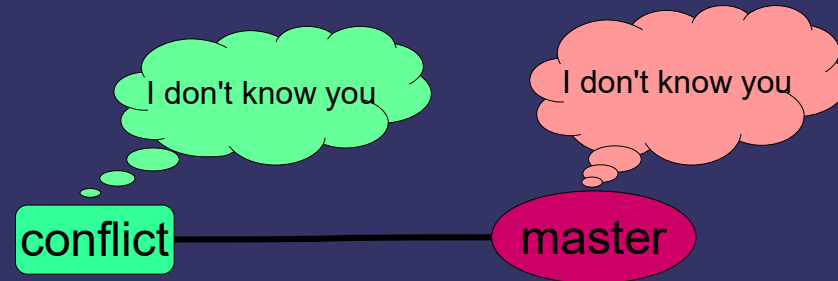
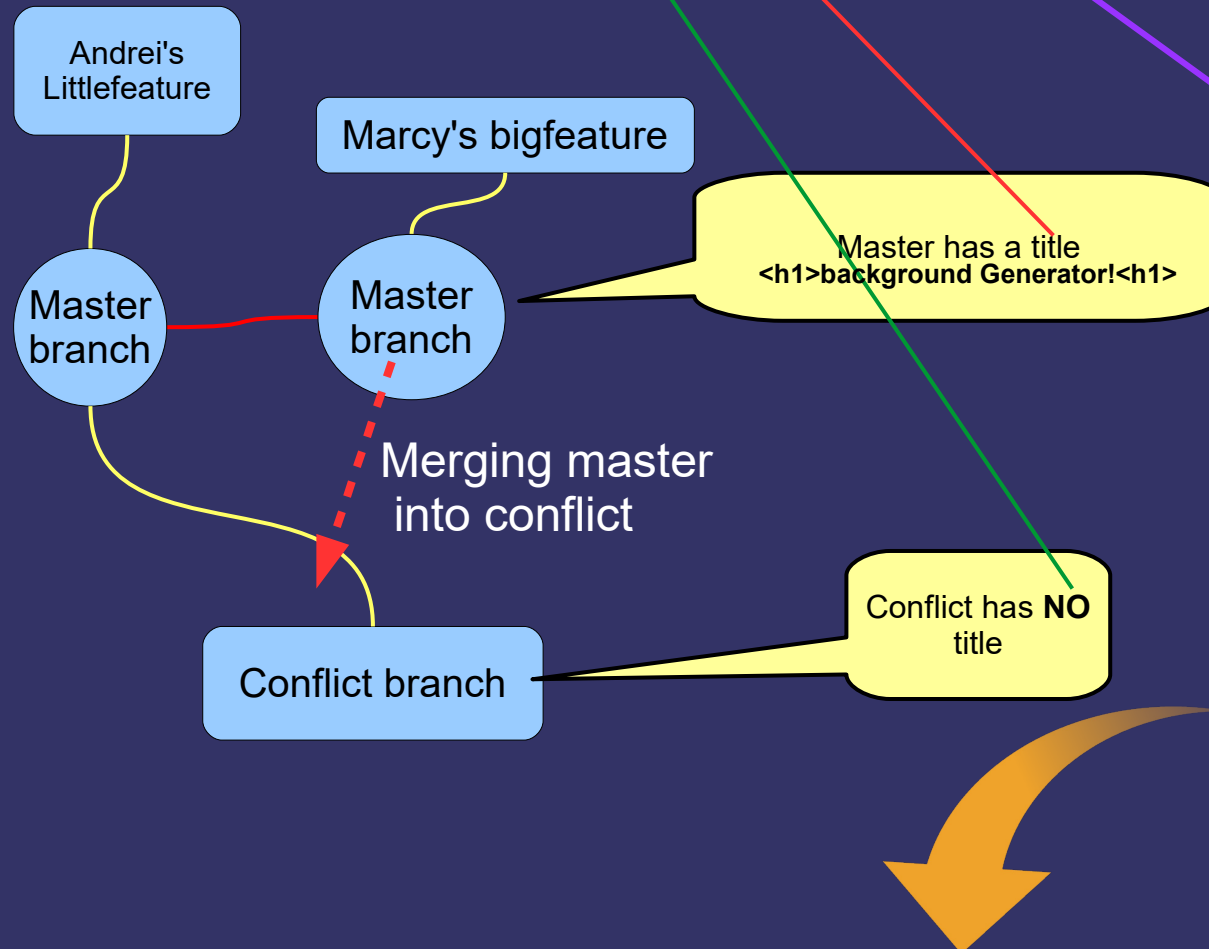
1  </head>
2  <body id="gradient">
3    <div class="cntr">
4  <<<<<<< HEAD (Current Change)
5  =====
6  <h1>Background Generator!</h1>
7  >>>>>>> master (Incoming Change)
8  <input class="color1" type="color" name="color1" value="#0
9  <input class="color2" type="color" name="color2" value="#f
10 <input class="color2" type="color" name="color2" value="#f
1
  
```




```
1 </head>
2 <body id="gradient">
3   <div class="cntr">
4     Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
5     <<<<<<< HEAD (Current Change)
6     =====
7     <h1>Background Generator!</h1>
8     >>>>>>> master (Incoming Change)
9     <input class="color1" type="color" name="color1" value="#00ff00">
10    <input class="color2" type="color" name="color2" value="#ff0000">
11    <input class="color2" type="color" name="color2" value="#ff0000">
```

<<<Head This is **conflict** brach

>>>master This is **master** branch



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 manually

Finally use the following commands

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

That's it for now! :-)



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

