

git troubleshooting

Contents

- fixing merge conflicts
- restoring a deleted branch

On this page, you will find some help/suggestions for how to untangle the occasional knot that can arise when working with git.

fixing merge conflicts

Warning

Jupyter notebook (**.ipynb**) files contain not just the markup/code for the notebook, but also any/all outputs, records of whether each cell has been executed (run), etc.

As such, it is **VERY IMPORTANT** for you to clear all of the outputs **before** you **commit** these files.

If you don't, you are going to have a bad time trying to clean up the merge conflicts.

[Skip to main content](#)

```
CONFLICT (add/add): Merge conflict in Week2/Practical12.ipynb
Automatic merge failed; fix conflicts and then commit the result.
```

Don't panic. All this means is that the same file (or files) has been edited on multiple branches, and **git** doesn't know which version is the "correct" one.

When you see this message, you'll need to open the file in a **text** editor (e.g., NotePad, Notepad++, but **not** Microsoft Word).

You're going to be looking for lines that look like this:

```
<<<<<< HEAD
<some stuff>

=====

<some other stuff>

>>>>>> cfa26ae841c9e6d252ce80239f31e4c5b568efb6
```

All you need to do is delete either `<some stuff>` or `<some other stuff>`, depending on which version you want to keep. So, for example, with this:

```
63  ",
64  <<<<<< HEAD
65  =====
66  "import os\n",
67  >>>>>> cfa26ae841c9e6d252ce80239f31e4c5b568efb6
68  "import os\n"
```

we want to keep the "import os" statement. So, we delete this:

[Skip to main content](#)

```
=====
```

as well as this:

```
>>>>>> cfa26ae841c9e6d252ce80239f31e4c5b568efb6
```

so that we're left with the version that we want:

```
63     "\n",
64     "import os\n",
65     "import geopandas as gpd\n",
66     "import matplotlib.pyplot as plt\n",
67     "from cartopy.feature import ShapelyFeature\n",
68     "import cartopy.crs as ccrs"
```

Once you're done editing the file (note that there may be multiple conflicts, so you'll need to check the file **carefully!**), you can add it like so:

```
git add <the file>
```

followed by:

```
git status
```

If you have successfully cleared all of the conflicts, you should see something like the following:

[Skip to main content](#)

```
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:
  modified:   Week2/Practical2.ipynb
```

At this point, you can commit the merge using `git commit`:

```
git commit -m "this is a commit message that makes sense"
```

and continue working.

restoring a deleted branch

If you accidentally delete a branch before it has been fully merged into the `main` branch, don't panic.

Open up a **Command Prompt** window, and navigate to your repository (in the example below, this is

```
C:\Users\e16006469\egm722\bobtheburner
```

):

[Skip to main content](#)

```
Anaconda Prompt (egm722)
(egm722) C:\Users\e16006469\egm722\bobtheburner>git branch
* main
  week2
  week3

(egm722) C:\Users\e16006469\egm722\bobtheburner>git branch -r
origin/HEAD -> origin/main
origin/main
origin/week2
origin/week3
origin/week4
upstream/main
upstream/week2
upstream/week3
upstream/week4
upstream/week5

(egm722) C:\Users\e16006469\egm722\bobtheburner>dir
Volume in drive C is OS
Volume Serial Number is F098-DD8D

Directory of C:\Users\e16006469\egm722\bobtheburner

04/04/2022  16:27    <DIR>          .
04/04/2022  16:27    <DIR>          ..
25/02/2022  11:25                27 .gitignore
25/02/2022  11:25               220 environment.yml
25/02/2022  11:25            19,051 LICENSE
04/04/2022  16:27            3,352 README.md
04/04/2022  16:27    <DIR>        Week1
04/04/2022  16:27    <DIR>        Week2
04/04/2022  16:27    <DIR>        Week3
04/04/2022  16:27    <DIR>        Week4
               4 File(s)        22,650 bytes
               6 Dir(s)  728,676,220,928 bytes free

(egm722) C:\Users\e16006469\egm722\bobtheburner>
```

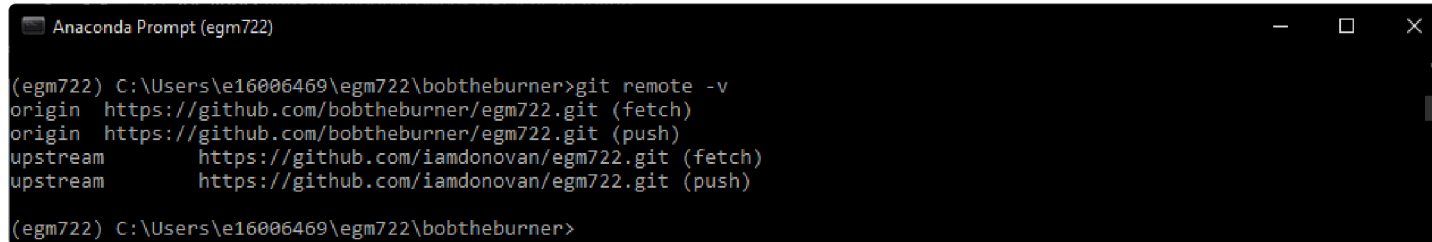
From the above, you can see from `git branch` that there is no local `week5` branch. From `git branch -r`, which lists all of the remote branches, you can see that there is no remote `week5` branch, either.

And, finally, from the output of the `dir` command, you can see that we only have the files for weeks 1-4 in our current working directory.

[Skip to main content](#)

First, though, check the list of remote repositories using `git remote -v`:

```
git remote -v
```



```
Anaconda Prompt (egm722)
(egm722) C:\Users\e16006469\egm722\bobtheburner>git remote -v
origin  https://github.com/bobtheburner/egm722.git (fetch)
origin  https://github.com/bobtheburner/egm722.git (push)
upstream https://github.com/iamdonovan/egm722.git (fetch)
upstream https://github.com/iamdonovan/egm722.git (push)
(egm722) C:\Users\e16006469\egm722\bobtheburner>
```

Note

If you don't see `upstream` as one of your remote repositories, you can add it with the following command:

```
git remote add upstream https://github.com/iamdonovan/egm722.git
```

this will add a new remote repository (`git remote add`), call it `upstream`, and point it to the iamdonovan/egm722 repository.

Now you need to make sure that you download (`fetch`) all of the remote branches, so that when you try to check something out from the new remote, git recognizes it:

```
git fetch --all
```

[Skip to main content](#)

```
Anaconda Prompt (egm722)

(egm722) C:\Users\e16006469\egm722\bobtheburner>git fetch --all
Fetching origin
Fetching upstream

(egm722) C:\Users\e16006469\egm722\bobtheburner>_
```

Now, you can check out the deleted branch from the upstream repository:

```
git checkout -b <deleted branch> upstream/<deleted branch>
```

```
Anaconda Prompt (egm722)

(egm722) C:\Users\e16006469\egm722\bobtheburner>git checkout -b week5 upstream/week5
Switched to a new branch 'week5'
Branch 'week5' set up to track remote branch 'week5' from 'upstream'.

(egm722) C:\Users\e16006469\egm722\bobtheburner>dir
Volume in drive C is OS
Volume Serial Number is F098-DD8D

Directory of C:\Users\e16006469\egm722\bobtheburner

04/04/2022  16:35    <DIR>          .
04/04/2022  16:35    <DIR>          ..
25/02/2022  11:25                27 .gitignore
25/02/2022  11:25               220 environment.yml
25/02/2022  11:25            19,051 LICENSE
04/04/2022  16:35               3,342 README.md
04/04/2022  16:35    <DIR>          Week5
                   4 File(s)        22,640 bytes
                   3 Dir(s)  728,794,054,656 bytes free

(egm722) C:\Users\e16006469\egm722\bobtheburner>_
```

And you should be able to continue working with the branch as you did before deleting it.

< [Previous](#)
[help! something went wrong!](#)

[python troubleshooting](#) > [Next](#)