Git Conflicts

Step 1: Make a clone of your git practice repository

cd ~/

git clone git@github.com:zappala/gitpractice git1

Be sure to subsitute your username above, and don't use mine. You can click on the "Clone or download" button in GitHub to get the full path name for your own repository.

You should be able to view your repository on GitHub and see this file in the repository. If you run **git status** you will see that your local repository is in sync with the remote repository on GitHub.

Step 2: Make a second clone of this repository

We're going to pretend you have a friend editing the same code. Clone a second copy of the repository:

```
cd ..
git clone git@github.com:zappala/gitpractice git2
cd git2
```

Be sure to subsitute your username above, and don't use mine. You can click on the "Clone or download" button in GitHub to get the full path name for your own repository.

You will be working in **git1** and your "friend" is working in **git2**.

Step 3: Your friend edits the file and pushes her changes

On your local machine, in the **git2** directory, edit the **test.txt** file so that it now reads as follows:

```
One
two
go
```

Commit this change and push it to the repository:

```
git add test.txt
git commit -m "Edited test file"
git push
```

Note that we had to add **test.txt** even though it was already in the repository. You can skip this step if you add the "-a" flag to **git commit**. This will automatically add any files to your commit

that have been modified.

You should be able to view this file on GitHub and see that it now has the new content. You should also be able to click on Commits and see your commits so far.

You can also view the commits on your local machine:

```
git log
```

Step 4: Conflict! You edit the file and push your changes

On your local machine, change back to the **git1** directory:

```
cd ../git1
```

In the **git1** directory, change the text file to read:

```
One
two
ready?
Go!
```

Commit this change and push it.

```
git add test.txt
commit -m "Edited the file too"
git push
```

You should get an error that looks like this:

```
error: failed to push some refs to 'git@github.com:zappala/gitpractice'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

This tells you that your friend has pushed changes to the repository. You can't push your changes yet because you need to incorporate your friend's changes first.

The standard way to do this is by using a pull:

```
git pull
```

You should now see a new message:

```
Auto-merging test.txt
CONFLICT (content): Merge conflict in test.txt
Automatic merge failed; fix conflicts and then commit the result.
```

This is known as a merge conflict. Git is trying to automatically merge your changes with your friends', and it found you both changed the same file **and** the changes were incompatible. (Sometimes you can change the same file in different places and it will automatically merge these changes.)

To resolve the conflict, you need to edit the file and make it match the version you want. If you edit the file, you should see something like this:

```
One
two
<<<<<<< HEAD
ready?
Go!
=======
go
>>>>>>> c6fe9cf1097175eedb0877f4a35e37cab1865350
```

This uses a format to show you the changes you made first (between the HEAD marker and the equal signs) and the changes your friend made before you (between the equal signs and the brackets). HEAD is a pointer to the last commit in your version of the repository, and c6fe9cf1097175eedb0877f4a35e37cab1865350 is a pointer to the last commit in her version of the repository.

Edit the file to have whatever changes you prefer, then:

```
git add test.txt
git commit -m "resolved conflict"
git push
```

This is known as *resolving a conflict*. You should now see the changes in GitHub. If you change back to your friend's repository in **git2** you can pull these changes in:

```
cd ../git2
git pull
```

The two repositories are now in sync. If you use git log in either local repository, you should see something like this:

```
commit 304a2fb55cfc713abac10bc5456e1afb6e45318a (HEAD -> master, origin/master)
Merge: 645686a c6fe9cf
Author: Daniel Zappala <daniel.zappala@gmail.com>
Date: Sat Jan 13 21:54:19 2018 -0700
```

```
Fixed conflict.

commit 645686a1fed6f867908d186930776739e32d2267
Author: Daniel Zappala <daniel.zappala@gmail.com>
Date: Sat Jan 13 21:33:13 2018 -0700

edited the file too

commit c6fe9cf1097175eedb0877f4a35e37cab1865350
Author: Daniel Zappala <daniel.zappala@gmail.com>
Date: Sat Jan 13 21:30:33 2018 -0700

Edited file

commit 9b8f6ef3dbf3d568efd26894e37e3391ac176c3d
Author: Daniel Zappala <daniel.zappala@gmail.com>
Date: Sat Jan 13 21:19:12 2018 -0700

added file
```

This shows all the commits you have made so far, most recent commit first.

✔ ⛓ ✔ 📄 (https://byu.instructure.com/courses/5724/modules/items/362832)

✔ 📝 (https://byu.instructure.com/courses/5724/modules/items/366435)

✔ 📄 (https://byu.instructure.com/courses/5724/modules/items/362834)

✔ 📝 (https://byu.instructure.com/courses/5724/modules/items/366436)

🚀 (https://byu.instructure.com/courses/5724/modules/items/362835)

✔ 📄 (https://byu.instructure.com/courses/5724/modules/items/362827)

✔ 📄 (https://byu.instructure.com/courses/5724/modules/items/362932)

📍 ✔ 📄 (https://byu.instructure.com/courses/5724/modules/items/362933)

✔ 📝 (https://byu.instructure.com/courses/5724/modules/items/362935)

📄 (https://byu.instructure.com/courses/5724/modules/items/412348)

✔ 📝 (https://byu.instructure.com/courses/5724/modules/items/366437)

✔ 📝 (https://byu.instructure.com/courses/5724/modules/items/366438)

✔ 📄 (https://byu.instructure.com/courses/5724/modules/items/415270)