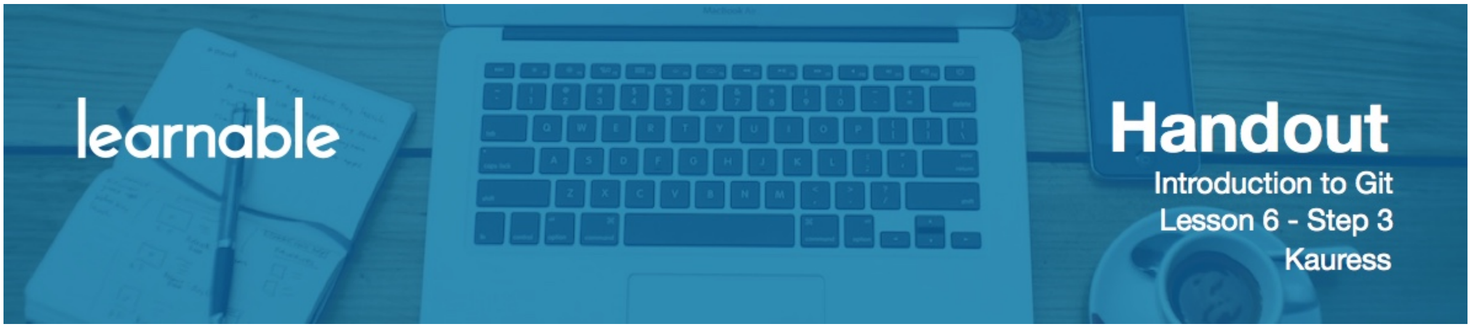

learnable

Handout

Introduction to Git
Lesson 6 - Step 3
Kauress

Introduction

In this step we'll talk a bit about a common reason why a `git push` and `pull` can be rejected and how to resolve that.

Avoid rejectec push requests

Let's imagine you and your teammate Lara both working independently on the same remote repo at the same time. We'll go to GitHub and make a change on Lara's behalf, and then we'll make our own changes locally and push them.

So first edit the *readme* file on GitHub by writing "Final edit by Lara" and commit it with message "final edit by Lara".

Now locally let's do some changes as well. Suppose you don't know about Lara's change to *readme*. Open up *css/business-casual.css* file in the code editor. Scroll down to the bottom of the file and put the following line:

```
.brand {color:#FF0000;}
```

Open up your terminal and run

```
git status
```

As expected it tells us that the *business-casual.css* file has changed. Add the file with

```
git add .
```

I'm just using the period as I don't really want to type out the whole CSS file name.

Commit the file change with

```
git commit -m "Changed the color of brand class"
```

Push our changes with

```
git push origin master
```

It says it's been rejected because the remote repo has changes that we don't have locally. So, we have to sync the local and remote repos by first pulling any changes from the remote repo.

Do that with

```
git pull origin master
```

Remember that `git pull` is a `git fetch` and `git merge` combined. Now the local copy of *readme* has the new changes.

```
git log --pretty=oneline
```

You will see our commit to change the color of the brand class, which was rejected.

Now we have to push our commit again:

```
git push origin master
```

Refresh GitHub page and you should see the commit with the file changes.