

Opera Clase #3.pdf

Code School - Try Git

try.github.io/levels/1/challenges/2

TryGit—1143x310

Press enter to submit commands

```
> git init

Initialized empty Git repository in ./git/
Success!
```

\$ |

My Octobox Repository

.git

Advice

The .git directory

On the left you'll notice a .git directory. It's usually hidden but we're showing it to you for convenience.

If you click it you'll notice it has all sorts of directories and files inside it. You'll rarely ever need to do anything inside here but it's the guts of Git, where all the magic happens.

This screenshot shows the initial step of creating a new Git repository. The terminal window displays the command 'git init' being run, which successfully initializes an empty repository in the current directory. Below the terminal is a file browser window titled 'My Octobox Repository' showing a single entry: '.git'. To the right of the browser is an 'Advice' panel with a cartoon octopus icon. It explains that the '.git' directory contains the core Git metadata and rarely needs to be interacted with directly.

Opera Clase #3.pdf

Code School - Try Git

Sign In - Code School

try.github.io/levels/1/challenges/3

TryGit—1143x310

```
Initialized empty Git repository in ./git/
Success!
$ git status

# On branch master
#
# Initial commit
#
nothing to commit (create/copy files and use "git add" to track)
Success!
$
```

My Octobox Repository

.git

octocat.txt

Advice

Tip:

It's healthy to run `git status` often. Sometimes things change and you don't notice it.

This screenshot shows the repository after adding a file named 'octocat.txt'. The terminal window now shows the output of 'git status', indicating that there is nothing to commit but files have been created. The file browser window shows both the '.git' directory and the 'octocat.txt' file. The 'Advice' panel provides a tip about regularly running 'git status' to stay updated with changes in the repository.

Opera | Clase #3.pdf | Code School - Try Git | Sign In - Code School | Download The Legend of | +

try.github.io/levels/1/challenges/4

TryGit—1143x310

```
Success!
$ git status

# On branch master
# Initial commit
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       octocat.txt
nothing added to commit but untracked files present (use "git add" to track)

Success!
$ |
```

My Octobox Repository

- .git
- octocat.txt

Advice

staged:
Files are ready to be committed.

unstaged:
Files with changes that have not been prepared to be committed.

untracked:
Files aren't tracked by Git yet. This usually indicates a newly created file.

deleted:
File has been deleted and is waiting to be

Opera | Clase #3.pdf | Code School - Try Git | Sign In - Code School | Download The Legend of | +

try.github.io/levels/1/challenges/5

TryGit—1143x310

```
# Initial commit
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       octocat.txt
nothing added to commit but untracked files present (use "git add" to track)

Success!
$ git add octocat.txt

Nice job, you've added octocat.txt to the Staging Area
$ |
```

My Octobox Repository

- .git
- octocat.txt

Advice

add all:
You can also type `git add -A` . where the dot stands for the current directory, so everything in and beneath it is added. The `-A` ensures even file deletions are included.

git reset:
You can use `git reset <filename>` to remove a file or files from the staging area.

Opera | Clase #3.pdf | Code School - Try Git | Sign In - Code School | Download The Legend of | +

try.github.io/levels/1/challenges/6

Nice job, you've added octocat.txt to the Staging Area

```
$ git status
```

```
# On branch master
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file:   octocat.txt
#
Success!
```

```
$
```

TryGit—1143x310

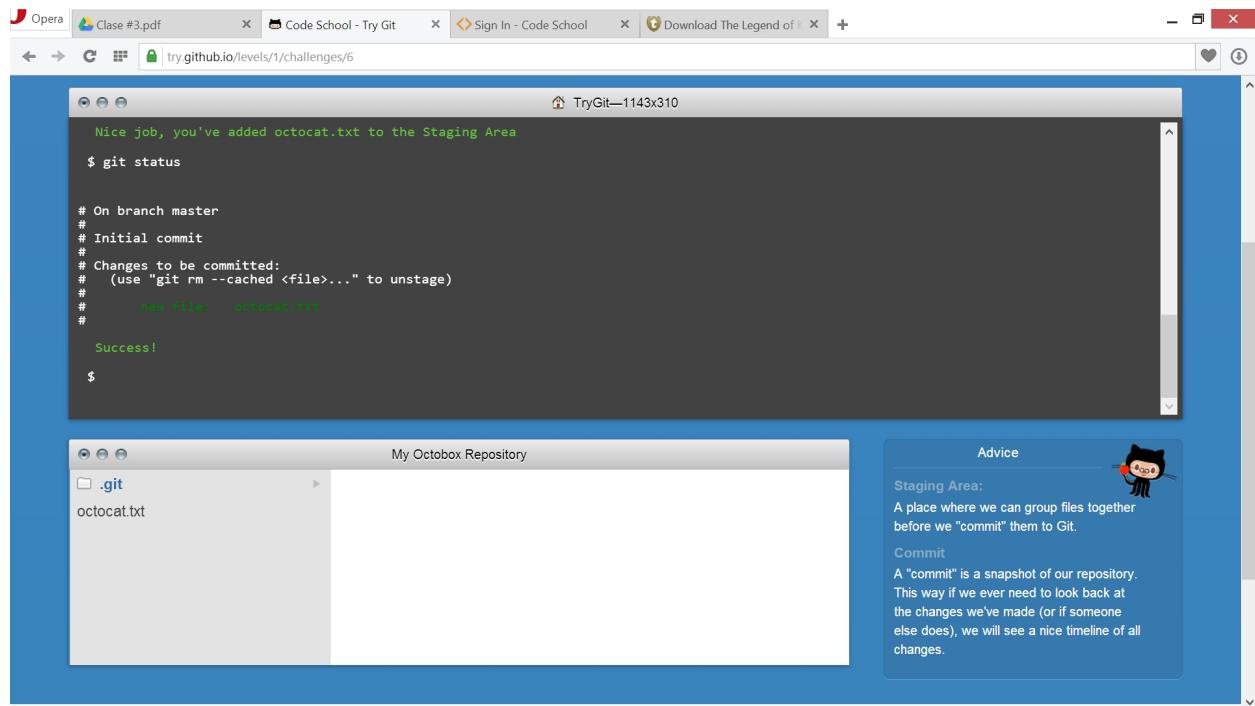
My Octobox Repository

- .git
- octocat.txt

Advice

Staging Area: A place where we can group files together before we "commit" them to Git.

Commit: A "commit" is a snapshot of our repository. This way if we ever need to look back at the changes we've made (or if someone else does), we will see a nice timeline of all changes.



Opera | Clase #3.pdf | Code School - Try Git | Sign In - Code School | Download The Legend of | +

try.github.io/levels/1/challenges/7

TryGit—1143x310

```
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file:   octocat.txt
#
Success!
```

```
$ git commit -m "Add cute octocat story"
```

```
[master (root-commit) 0db1a86] Add cute octocat story
1 file changed, 1 insertion(+)
create mode 100644 octocat.txt
Success!
```

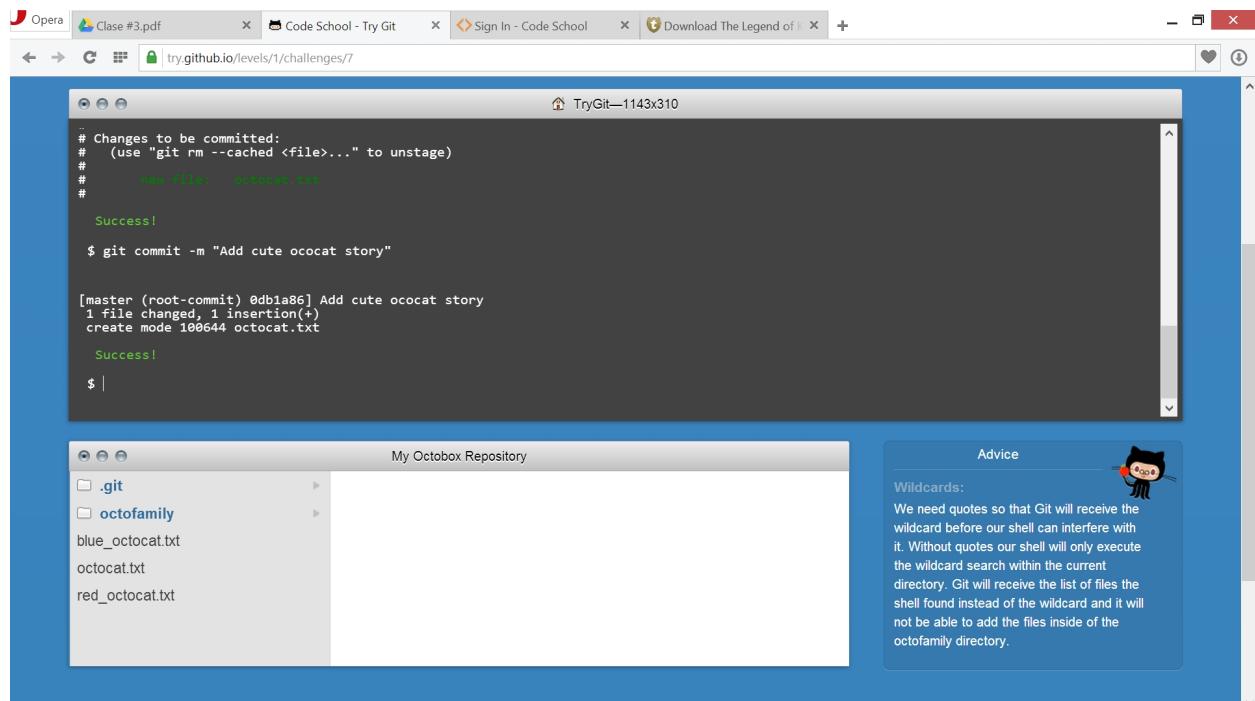
```
$ |
```

My Octobox Repository

- .git
- octofamily
- blue_octocat.txt
- octocat.txt
- red_octocat.txt

Advice

Wildcards: We need quotes so that Git will receive the wildcard before our shell can interfere with it. Without quotes our shell will only execute the wildcard search within the current directory. Git will receive the list of files the shell found instead of the wildcard and it will not be able to add the files inside of the octofamily directory.



Opera Clase #3.pdf Code School - Try Git Sign In - Code School Download The Legend of +

try.github.io/levels/1/challenges/8

TryGit—1143x310

```
$ git commit -m "Add cute octocat story"

[master (root-commit) 0db1a86] Add cute octocat story
1 file changed, 1 insertion(+)
create mode 100644 octocat.txt
Success!
$ git add *.txt

Success!
$ |
```

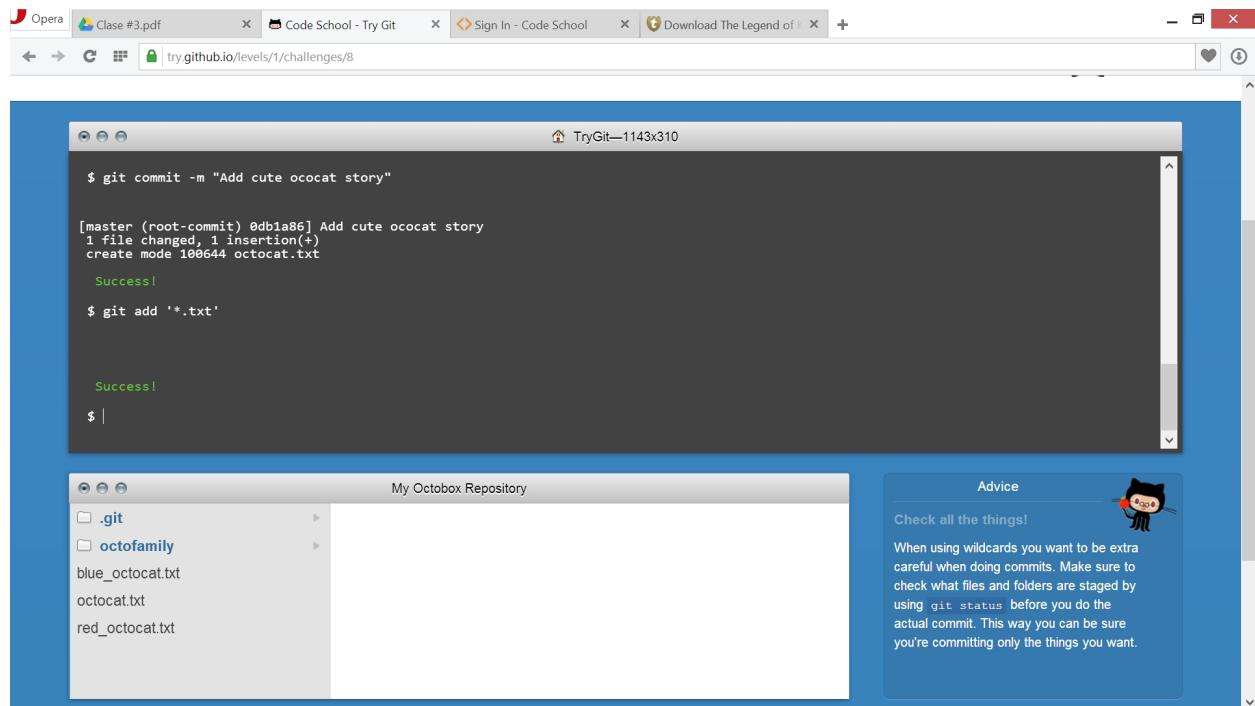
My Octobox Repository

- .git
- octofamily
 - blue_octocat.txt
 - octocat.txt
 - red_octocat.txt

Advice

Check all the things!

When using wildcards you want to be extra careful when doing commits. Make sure to check what files and folders are staged by using `git status` before you do the actual commit. This way you can be sure you're committing only the things you want.



Opera Clase #3.pdf Code School - Try Git Sign In - Code School Download The Legend of +

try.github.io/levels/1/challenges/9

TryGit—1143x310

```
Success!
$ git commit -m 'Add all the octocat txt files'

[master 3852b4d] Add all the octocat txt files
4 files changed, 4 insertions(+)
create mode 100644 blue_octocat.txt
create mode 100644 octofamily/baby_octocat.txt
create mode 100644 octofamily/momma_octocat.txt
create mode 100644 red_octocat.txt
Success!
$
```

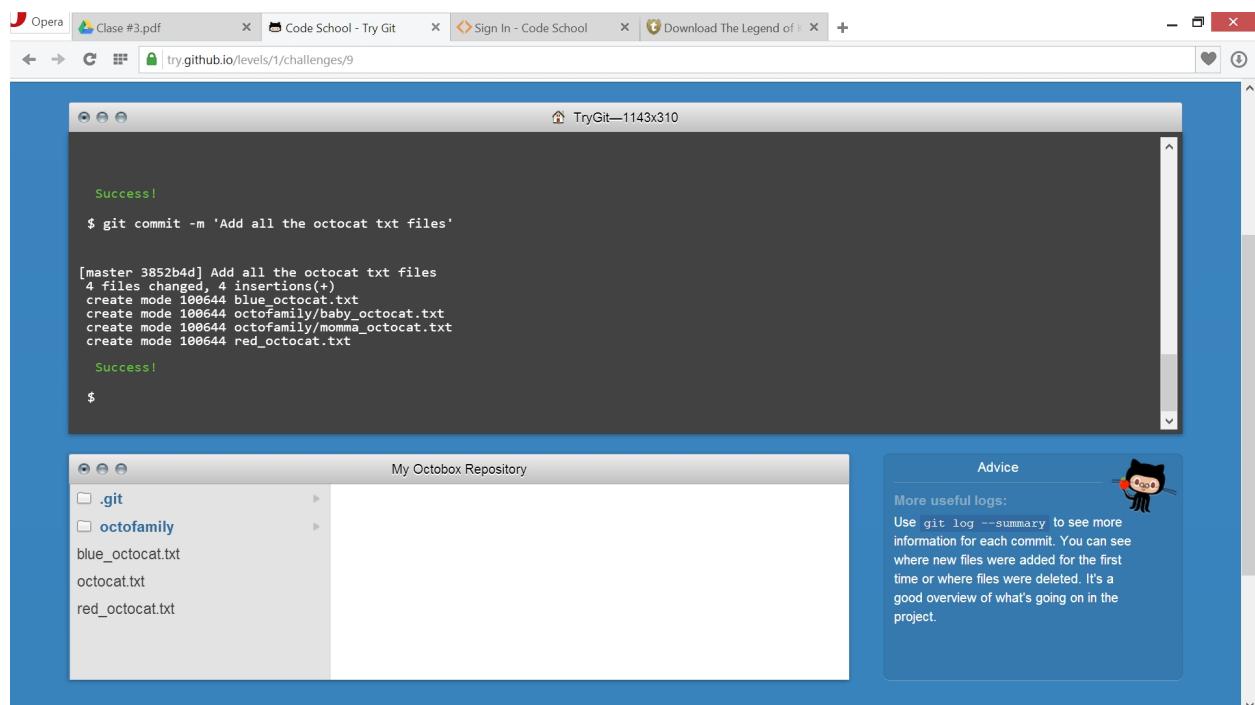
My Octobox Repository

- .git
- octofamily
 - blue_octocat.txt
 - octocat.txt
 - red_octocat.txt

Advice

More useful logs:

Use `git log --summary` to see more information for each commit. You can see where new files were added for the first time or where files were deleted. It's a good overview of what's going on in the project.



Opera Clase #3.pdf Code School - Try Git Sign In - Code School Download The Legend of +

try.github.io/levels/1/challenges/10

TryGit—1143x310

```
$ git log

commit 3852b4db1634463d0bb4d267edb7b3f9cd02ace1
Author: Try Git <try_git@github.com>
Date:   Sat Oct 10 08:30:00 2020 -0500

    Add all the octocat txt files

commit b652edfd888cd3d5e7fc857d0dabc5a0fcb5e28
Author: Try Git <try_git@github.com>
Date:   Sat Oct 10 08:30:00 2020 -0500

    Added cute octocat story

Success!
```

My Octobox Repository

- .git
- octobox

Advice

git remote:

Git doesn't care what you name your remotes, but it's typical to name your main one `origin`.

It's also a good idea for your main repository to be on a remote server like GitHub in case your machine is lost at sea during a transatlantic boat cruise or crushed by three monkey statues during an earthquake.

Opera Clase #3.pdf Code School - Try Git Sign In - Code School Download The Legend of +

try.github.io/levels/1/challenges/11

TryGit—1143x310

```
Add all the octocat txt files
commit b652edfd888cd3d5e7fc857d0dabc5a0fcb5e28
Author: Try Git <try_git@github.com>
Date:   Sat Oct 10 08:30:00 2020 -0500

    Added cute octocat story

Success!
```

```
$ git remote add origin https://github.com/try-git/try_git.git

Success!
```

My Octobox Repository

- .git
- octofamily

blue_octocat.txt
octocat.txt
red_octocat.txt

Advice

Cool Stuff:

When you start to get the hang of git you can do some really cool things with `hooks` when you push.

For example, you can upload directly to a webserver whenever you push to your master remote instead of having to upload your site with an `ftp` client. Check out [Customizing Git - Git Hooks](#) for more information.

Opera Clase #3.pdf

Code School - Try Git

Sign In - Code School

Download The Legend of

try.github.io/levels/1/challenges/12

TryGit—1143x310

```
Success!
$ git remote add origin https://github.com/try-git/try_git.git

Success!
$ git push -u origin master

Branch master set up to track remote branch master from origin.
Success!
$
```

My Octobox Repository

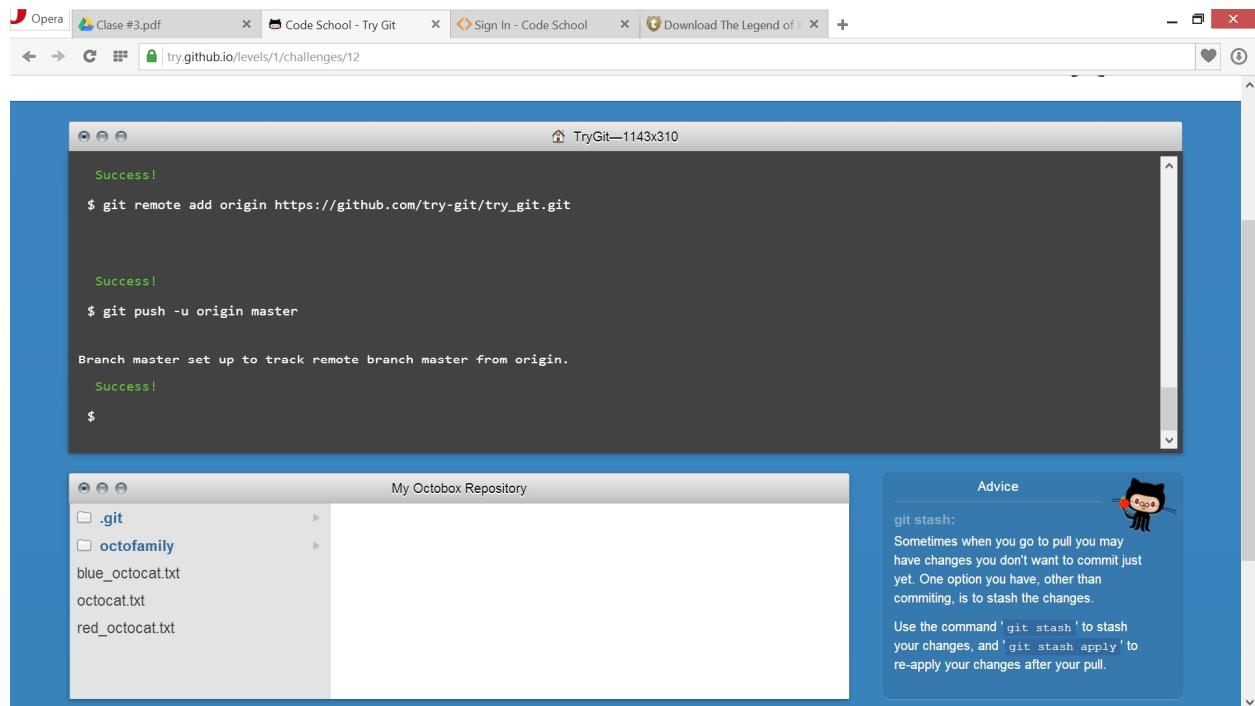
- .git
- octofamily
 - blue_octocat.txt
 - octocat.txt
 - red_octocat.txt

Advice

git stash:

Sometimes when you go to pull you may have changes you don't want to commit just yet. One option you have, other than committing, is to stash the changes.

Use the command 'git stash' to stash your changes, and 'git stash apply' to re-apply your changes after your pull.



Opera Clase #3.pdf

Code School - Try Git

Sign In - Code School

Download The Legend of

try.github.io/levels/1/challenges/13

TryGit—1143x310

```
Branch master set up to track remote branch master from origin.
Success!
$ git pull origin master

Updating 3852b4d..3e70b0f
Fast-forward
 yellow_octocat.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 yellow_octocat.txt
Success!
$
```

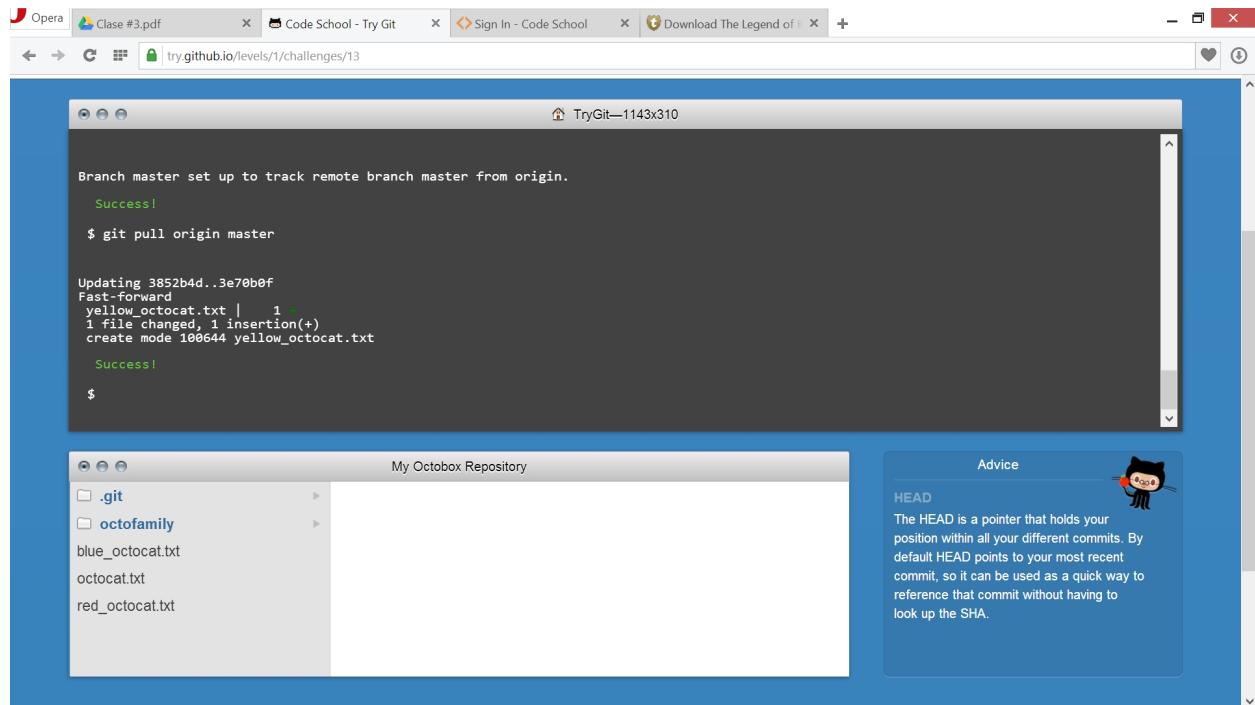
My Octobox Repository

- .git
- octofamily
 - blue_octocat.txt
 - octocat.txt
 - red_octocat.txt

Advice

HEAD

The HEAD is a pointer that holds your position within all your different commits. By default HEAD points to your most recent commit, so it can be used as a quick way to reference that commit without having to look up the SHA.



Opera Clase #3.pdf

Code School - Try Git

Sign In - Code School

Download The Legend of

try.github.io/levels/1/challenges/14

TryGit—1143x310

```
create mode 100644 yellow_octocat.txt
Success!
$ git diff HEAD

diff --git a/octocat.txt b/octocat.txt
index 7d8d808..e725ef6 100644
--- a/octocat.txt
+++ b/octocat.txt
@@ -1 +1 @@
-A Tale of Two Octocats
+An OctoCat's Tale of Two Octocats and an Octodog
Success!
$ |
```

My Octobox Repository

- .git
- octofamily
 - blue_octocat.txt
 - octocat.txt
 - red_octocat.txt

Advice

Commit Etiquette:

You want to try to keep related changes together in separate commits. Using 'git diff' gives you a good overview of changes you have made and lets you add files or directories one at a time and commit them separately.

Opera Clase #3.pdf

Code School - Try Git

Sign In - Code School

Download The Legend of

try.github.io/levels/1/challenges/15

TryGit—1143x310

```
diff --git a/octocat.txt b/octocat.txt
index 7d8d808..e725ef6 100644
--- a/octocat.txt
+++ b/octocat.txt
@@ -1 +1 @@
-A Tale of Two Octocats
+An OctoCat's Tale of Two Octocats and an Octodog
Success!
$ git add octofamily/octodog.txt

Success!
$
```

My Octobox Repository

- .git
- octofamily
 - blue_octocat.txt
 - octocat.txt
 - red_octocat.txt

Advice

Commit Etiquette:

You want to try to keep related changes together in separate commits. Using 'git diff' gives you a good overview of changes you have made and lets you add files or directories one at a time and commit them separately.

Opera | Clase #3.pdf | Code School - Try Git | Sign In - Code School | Download The Legend of | +

try.github.io/levels/1/challenges/16

TryGit—1143x310

```
Success!
$ git diff --staged

diff --git a/octofamly/octodog.txt b/octofamily/octodog.txt
new file mode 100644
index 0000000..cfbc74a
--- /dev/null
+++ b/octofamily/octodog.txt
@@ -0,0 +1 @@
Success!
$
```

My Octobox Repository

- 📁 .git
- 📁 octofamly
 - blue_octocat.txt
 - octocat.txt
 - red_octocat.txt

Advice

Commit Etiquette:



You want to try to keep related changes together in separate commits. Using 'git diff' gives you a good overview of changes you have made and lets you add files or directories one at a time and commit them separately.

Opera | Clase #3.pdf | Code School - Try Git | Sign In - Code School | Download The Legend of | +

try.github.io/levels/1/challenges/16

TryGit—1143x310

```
diff --git a/octofamily/octodog.txt b/octofamily/octodog.txt
new file mode 100644
index 0000000..cfbc74a
--- /dev/null
+++ b/octofamily/octodog.txt
@@ -0,0 +1 @@
Success!
$ git reset octofamily/octodog.txt

Success!
$
```

My Octobox Repository

- 📁 .git
- 📁 octofamly
 - blue_octocat.txt
 - octocat.txt
 - red_octocat.txt

Advice

The '…'



So you may be wondering, why do I have to use this '…' thing? `git checkout` seems to work fine without it. It's simply promising the command line that there are no more options after the '…'. This way if you happen to have a branch named `octocat.txt`, it will still revert the file, instead of switching to the branch of the same name.

Opera Clase #3.pdf Code School - Try Git Sign In - Code School Download The Legend of +

try.github.io/levels/1/challenges/18

TryGit—1143x310

```
$ git reset octofamily/octodog.txt
Success!
$ git checkout -- octocat.txt
Success!
$ |
```

My Octobox Repository

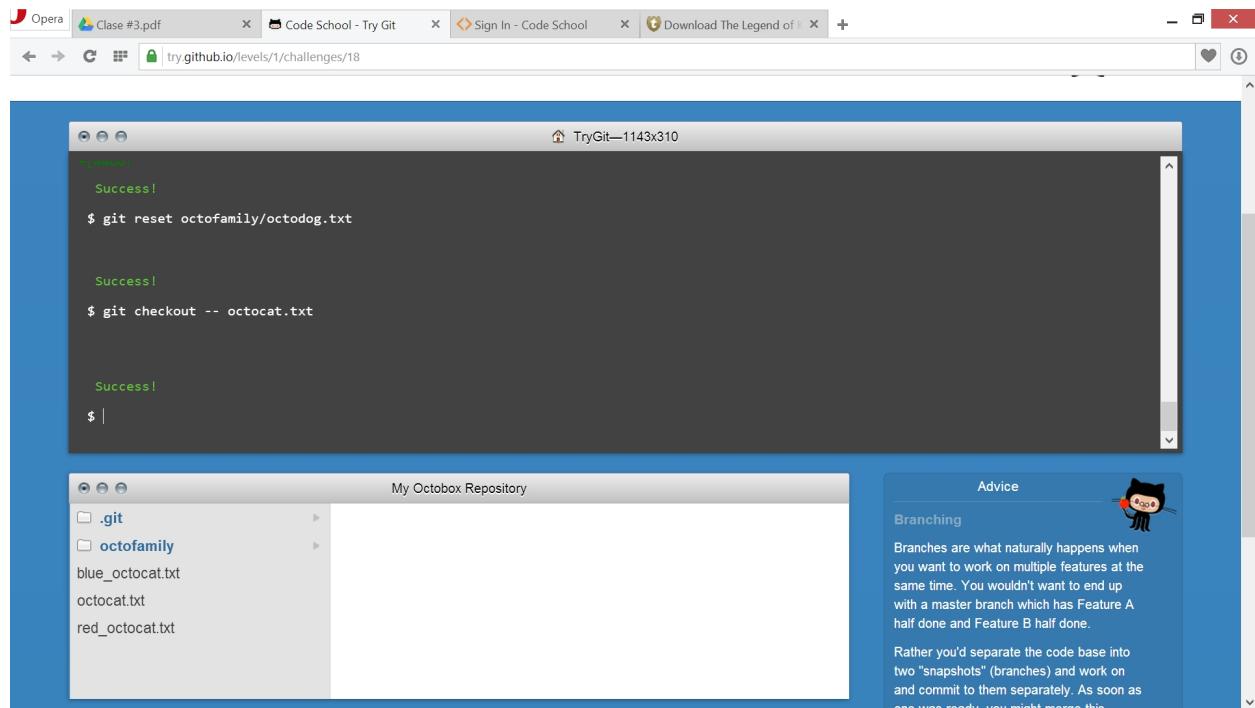
- .git
- octofamily
 - blue_octocat.txt
 - octocat.txt
 - red_octocat.txt

Advice

Branching

Branches are what naturally happens when you want to work on multiple features at the same time. You wouldn't want to end up with a master branch which has Feature A half done and Feature B half done.

Rather you'd separate the code base into two "snapshots" (branches) and work on and commit to them separately. As soon as one was ready, you might merge this.



Opera Clase #3.pdf Code School - Try Git Sign In - Code School Download The Legend of +

try.github.io/levels/1/challenges/19

TryGit—1143x310

```
$ git checkout -- octocat.txt
Success!
$ git branch clean_up
Success!
$ |
```

My Octobox Repository

- .git
- octofamily
 - blue_octocat.txt
 - octocat.txt
 - red_octocat.txt

Advice

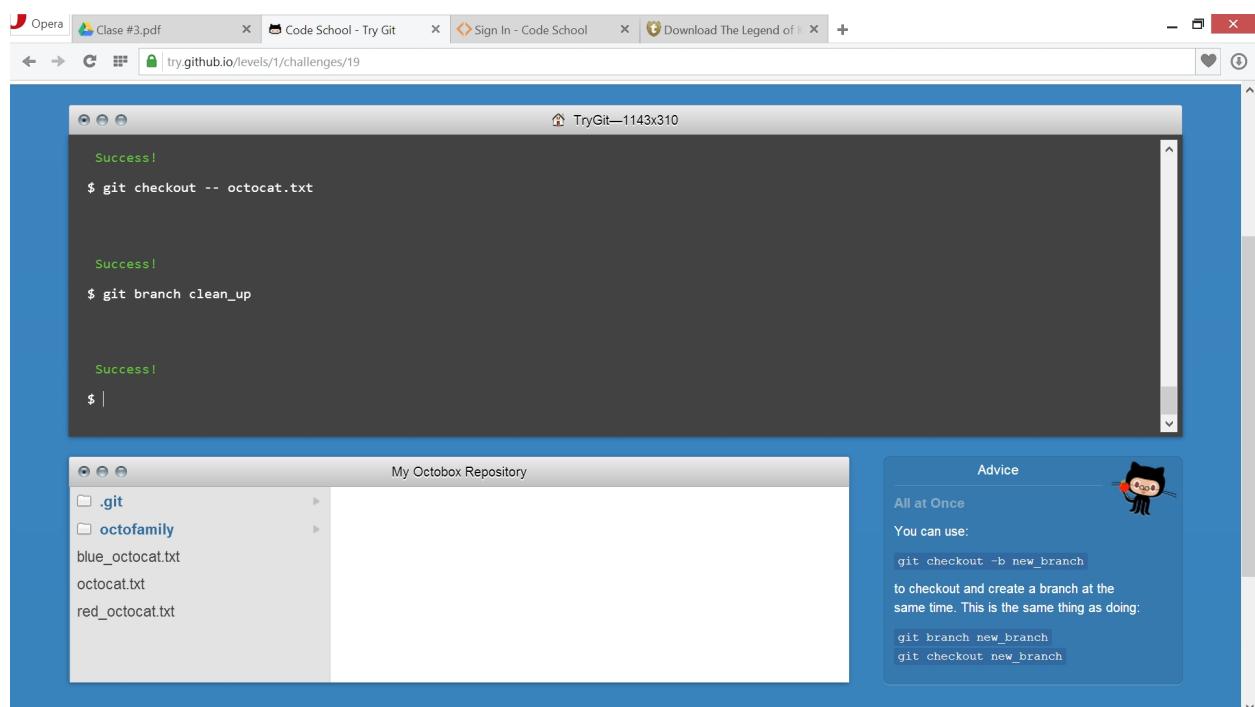
All at Once

You can use:

```
git checkout -b new_branch
```

to checkout and create a branch at the same time. This is the same thing as doing:

```
git branch new_branch
git checkout new_branch
```



Opera Clase #3.pdf

Code School - Try Git

Sign In - Code School

Download The Legend of

try.github.io/levels/1/challenges/20

TryGit—1143x310

```
Success!
$ git branch clean_up

Success!
$ git checkout clean_up

Switched to branch 'clean_up'
Success!
$
```

My Octobox Repository

- .git
- octofamily
 - blue_octocat.txt
 - octocat.txt
 - red_octocat.txt

Advice

Remove all the things!

Removing one file is great and all, but what if you want to remove an entire folder? You can use the recursive option on `git rm`.

`git rm -r folder_of_cats`

This will recursively remove all folders and files from the given directory.

Opera Clase #3.pdf

Code School - Try Git

Sign In - Code School

Download The Legend of

try.github.io/levels/1/challenges/21

TryGit—1143x310

```
Switched to branch 'clean_up'
Success!
$ git rm *.txt

rm 'blue_octocat.txt'
rm 'octocat.txt'
rm 'octofamily/baby_octocat.txt'
rm 'octofamily/momma_octocat.txt'
rm 'red_octocat.txt'

Success!
$ |
```

My Octobox Repository

- .git
- octofamily

Advice

The '-a' option

If you happen to delete a file without using `'git rm'` you'll find that you still have to `'git rm'` the deleted files from the working tree. You can save this step by using the `'-a'` option on `'git commit'`, which auto removes deleted files with the commit.

`git commit -am "Delete stuff"`

Opera | Clase #3.pdf | Code School - Try Git | Sign In - Code School | Download The Legend of | +

try.github.io/levels/1/challenges/22

git checkout master

TryGit—1143x310

```
rm 'red_octocat.txt'
Success!
$ git commit -m "Remove all the cats"

[clean_up 63540fe] Remove all the cats
5 files changed, 5 deletions(-)
 delete mode 100644 blue_octocat.txt
 delete mode 100644 octocat.txt
 delete mode 100644 octofamily/baby_octocat.txt
 delete mode 100644 octofamily/momma_octocat.txt
 delete mode 100644 red_octocat.txt
Success!
$
```

My Octobox Repository

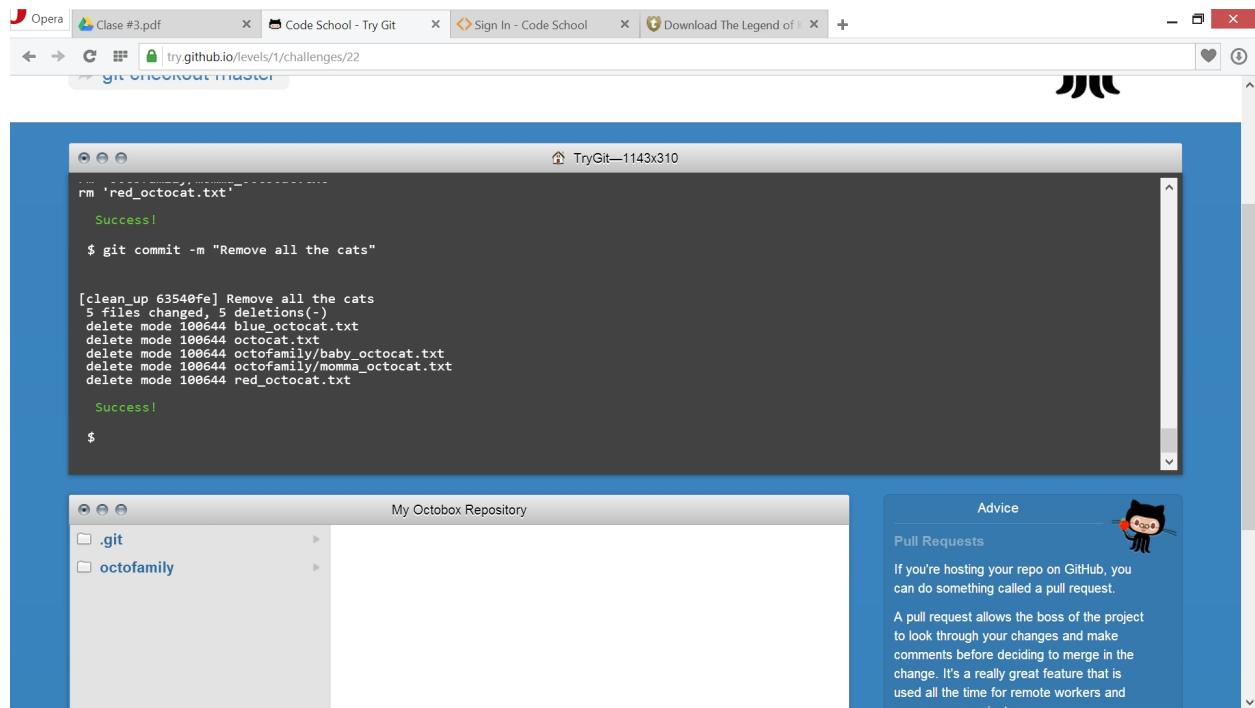
- .git
- octofamily

Advice

Pull Requests

If you're hosting your repo on GitHub, you can do something called a pull request.

A pull request allows the boss of the project to look through your changes and make comments before deciding to merge in the change. It's a really great feature that is used all the time for remote workers and open source projects.



Opera | Clase #3.pdf | Code School - Try Git | Sign In - Code School | Download The Legend of | +

try.github.io/levels/1/challenges/23

git checkout master

TryGit—1143x310

```
[clean_up 63540fe] Remove all the cats
5 files changed, 5 deletions(-)
 delete mode 100644 blue_octocat.txt
 delete mode 100644 octocat.txt
 delete mode 100644 octofamily/baby_octocat.txt
 delete mode 100644 octofamily/momma_octocat.txt
 delete mode 100644 red_octocat.txt
Success!
$ git checkout master

Switched to branch 'master'
Success!
$ |
```

My Octobox Repository

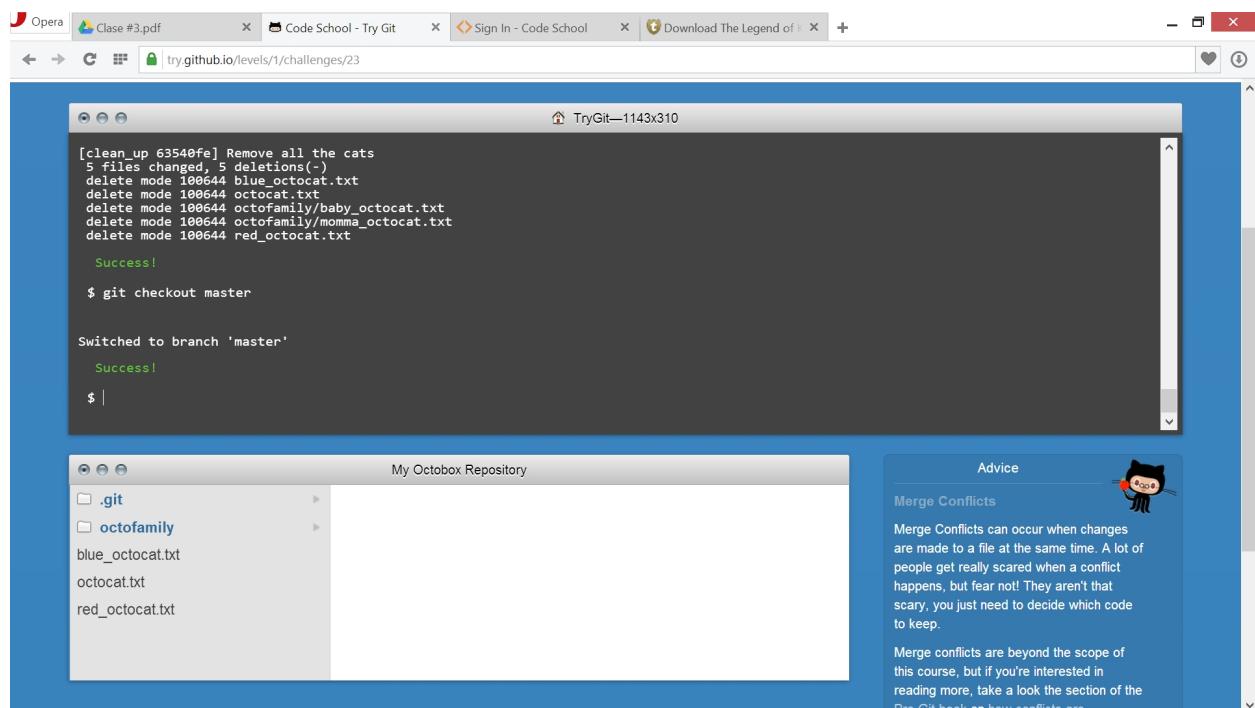
- .git
- octofamily
- blue_octocat.txt
- octocat.txt
- red_octocat.txt

Advice

Merge Conflicts

Merge Conflicts can occur when changes are made to a file at the same time. A lot of people get really scared when a conflict happens, but fear not! They aren't that scary, you just need to decide which code to keep.

Merge conflicts are beyond the scope of this course, but if you're interested in reading more, take a look at the section of the Pro Git book on how conflicts are



Opera Clase #3.pdf Code School - Try Git Sign In - Code School Download The Legend of +

try.github.io/levels/1/challenges/24

TryGit—1143x310

```
$ git merge clean_up

Updating 3852b4d..ec6888b
Fast-forward
 blue_octocat.txt      | 1 -
 octocat.txt            | 1 -
 octofamily/baby_octocat.txt | 1 -
 octofamily/momma_octocat.txt | 1 -
 red_octocat.txt        | 1 -
5 files changed, 5 deletions(-)
 delete mode 100644 blue_octocat.txt
 delete mode 100644 octocat.txt
 delete mode 100644 octofamily/baby_octocat.txt
 delete mode 100644 octofamily/momma_octocat.txt
 delete mode 100644 red_octocat.txt

Success!
```

\$ |

My Octobox Repository

- .git
- octofamily

Advice

Force delete

If you have been working on a feature branch and you decide you really don't want this feature anymore? You might decide to delete the branch since you're scrapping the idea. You'll notice that `git branch -d bad_feature` doesn't work. This is because `-d` won't let you delete something that hasn't been merged.

You can either add the `--force (-f)`

Opera Clase #3.pdf Code School - Try Git Sign In - Code School Download The Legend of +

try.github.io/levels/1/challenges/25

TryGit—1143x310

```
red_octocat.txt      | 1 -
5 files changed, 5 deletions(-)
 delete mode 100644 blue_octocat.txt
 delete mode 100644 octocat.txt
 delete mode 100644 octofamily/baby_octocat.txt
 delete mode 100644 octofamily/momma_octocat.txt
 delete mode 100644 red_octocat.txt

Success!
```

```
$ git branch -d clean_up

Deleted branch clean_up (was ec6888b).

Success!
```

\$ |

My Octobox Repository

- .git
- octofamily

Advice

Learning more about Git

We only scratched the surface of Git in this course. There is so much more you can do with it. Check out the [Git documentation](#) for a full list of functions.

The [Pro Git book](#), by Scott Chacon, is an excellent resource to teach you the inner workings of Git.

[help.github](#) and [GitHub Training](#) are also great for anything related to Git in general

Opera Clase #3.pdf Code School - Try Git Sign In - Code School Download The Legend of +

try.github.io/levels/1/challenges/25

TryGit—1143x310

```
$ git branch -d clean_up

Deleted branch clean_up (was ec6888b).
Success!
$ git push

To https://github.com/try-git/try_git.git
  3e70b0f..6df3903  master -> master
Success!
> |
```

My Octobox Repository

- .git
- octofamily

yellow_octocat.txt

Advice

Learning more about Git



We only scratched the surface of Git in this course. There is so much more you can do with it. Check out the [Git documentation](#) for a full list of functions.

The [Pro Git book](#), by Scott Chacon, is an excellent resource to teach you the inner workings of Git.

[help.github](#) and [GitHub Training](#) are also great for anything related to Git in general