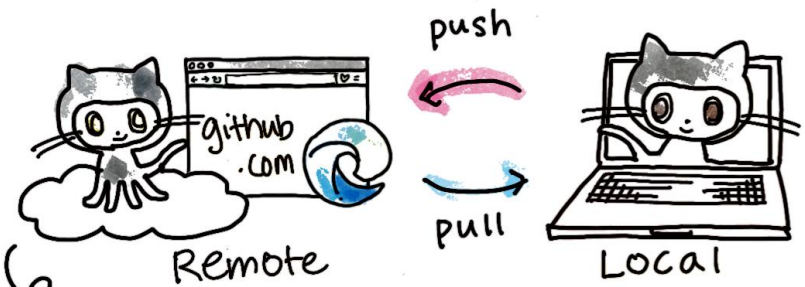
Introduction to GitHub

Git = a distributed version control system for tracking changes in your code history

GitHub = a cloud-based hosting where you can manage your Git repositories

Your first project with Git + GitHub



git-scm.com

⓪ Download + install Git

Then, set up your local Git profile

config

```
git config --global ...
```

① On github.com, Sign up / Sign in + Create a repository.

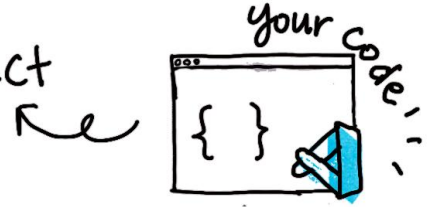
Terminal

② cd my project

③ git init

④ git status

⑤ git add .



⑥ git commit -m "first commit"

⑦ git remote add origin http://github.com/—.git

⑧ git push -u origin main

⑨ To add more changes:

- git add.
- git commit -m "typo fix"
- git push

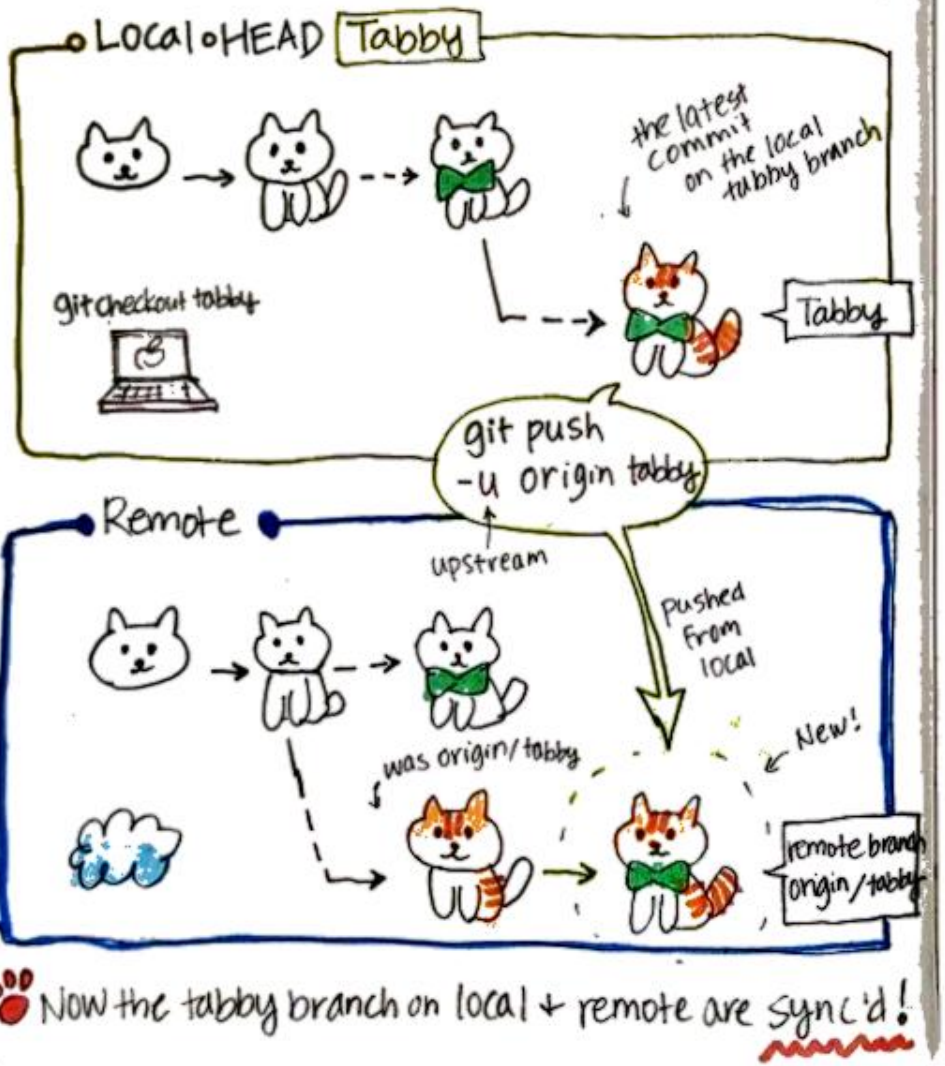
@AzureAdvocates / @girlie_mac

git Push

puss

@girlie_mac

♥ git push transfers your commits from your local repo to a remote repo!



♥ Now the tabby branch on local + remote are sync'd!

♥ git push

git push <remote> <branch>

e.g. origin e.g. tabby

push the specified branch w/ all commits

git push -u <remote> <branch>

↳ = --set-upstream

the -u flag sets up the association between your branch + the remote branch explicitly.
you don't need it once you've done!

git push -f <remote> <branch>

↳ --force

force the push, even if it results in a non-fast-foward merge.
just ignore + push!

git push --all <remote>

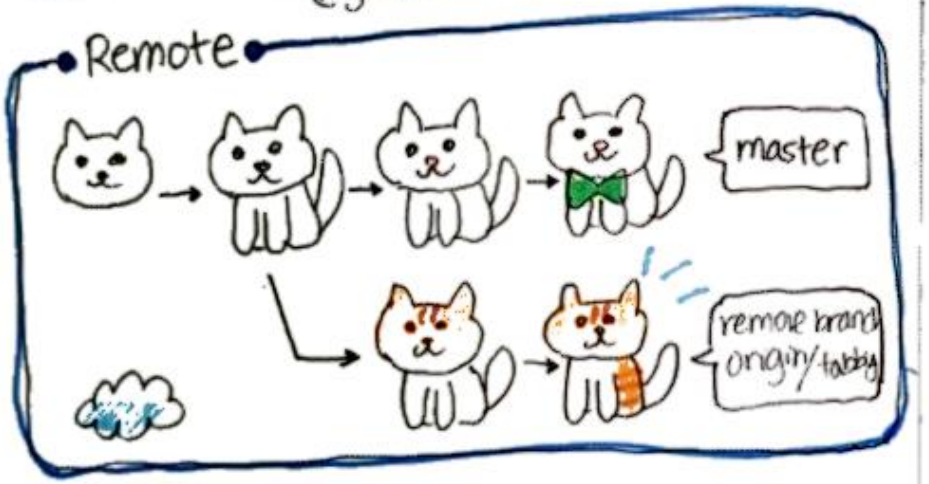
push all of your local branches!

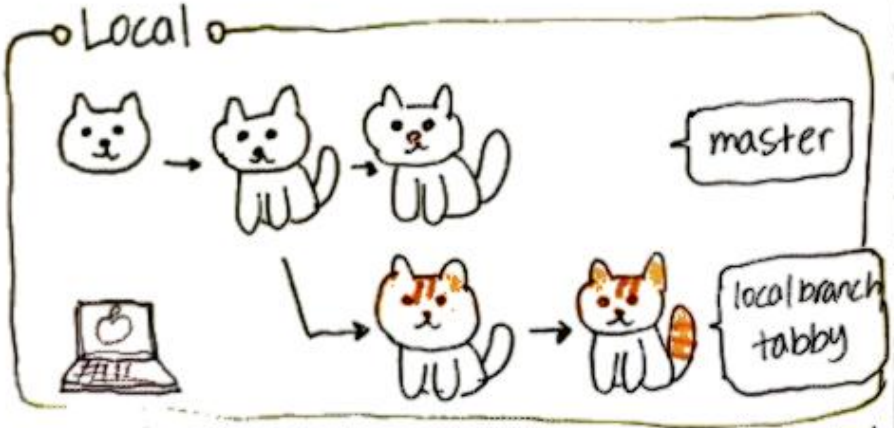
git Pull

purr

@girlie_mac

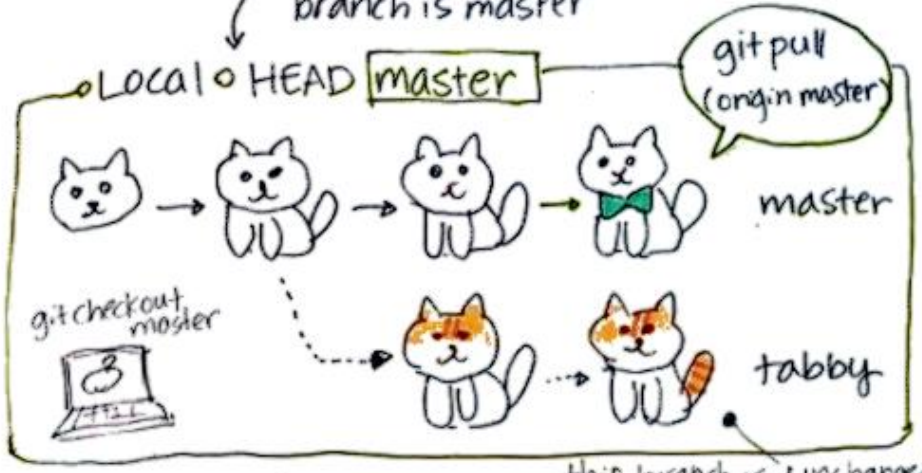
git pull updates your current HEAD branch w/ the latest Δs from remote





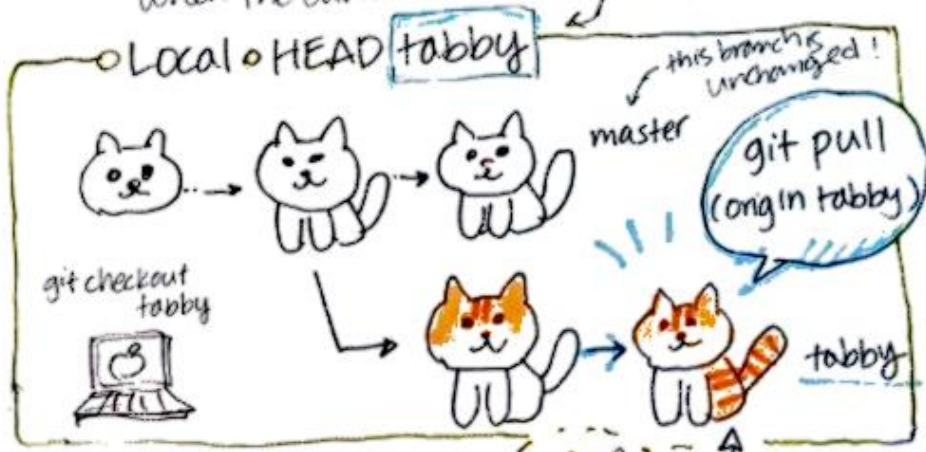
Now I'm going to 'git pull' to update a branch!

When the current branch is master



♥ This is same as:
- git fetch origin
- git merge origin/master

When the current branch is tabby



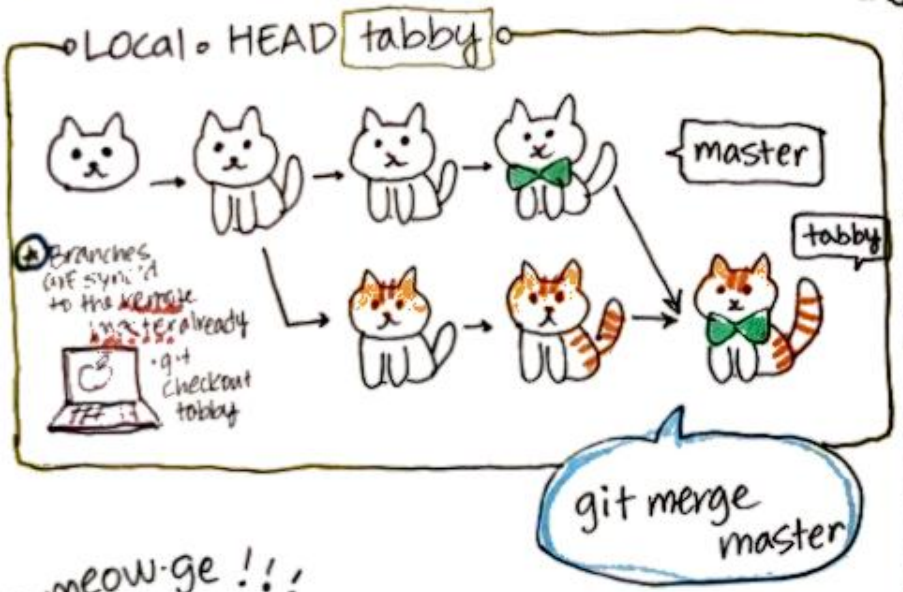
♥ Same as:
- git fetch origin
- git merge origin/tabby

git merge

meow! ge

@girlie_mac

- git merge incorporates changes into the current branch.



git meow-ge !!!

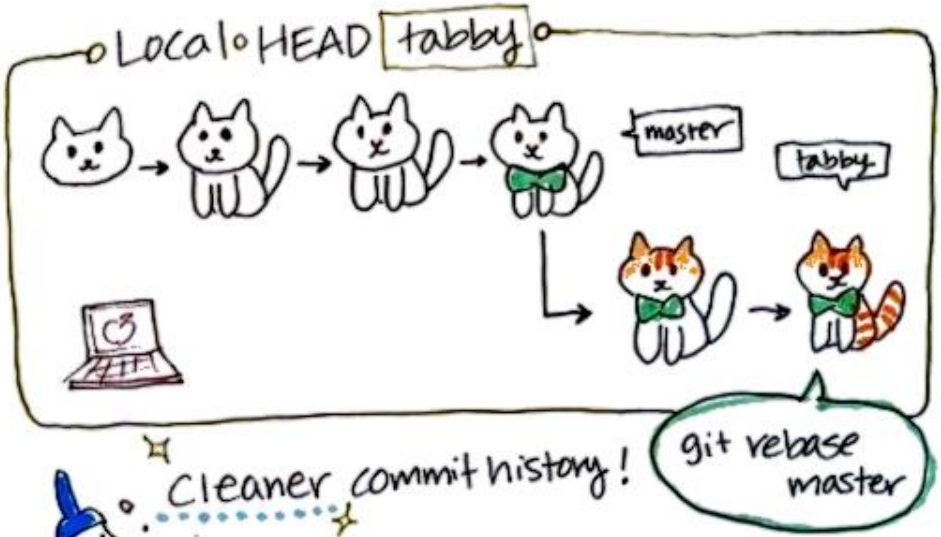
- ★ merge is like having 2 parents & 1 resulting child!
- ★ rebase adds all new Δs on top of one parent.

PAWSOME!

git rebase

@girlie_mac

- git rebase moves a branch from one commit to another.



Cleaner commit history!

If you want to rebase from the remote master instead of local. do either:

```
$ git fetch origin
$ git rebase origin/master
```

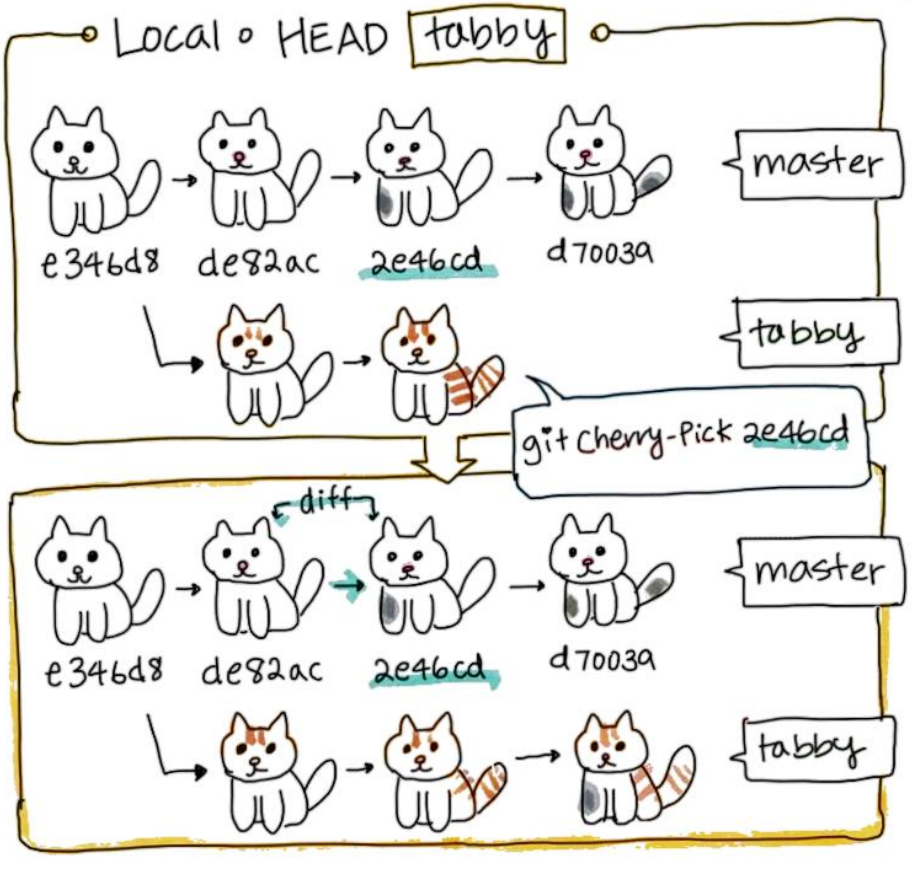
or

```
$ git pull --rebase origin master
```

git cherry-pick

@girlie_mac

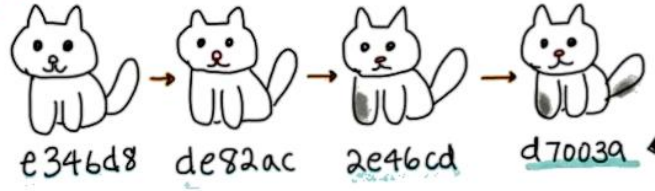
♥ git cherry-pick lets you grab some commits from one branch & apply it into another branch!



git log

♥ git log lets you view the commit history

@girlie_mac



♥ git log prints out

```
Commit e346d8
Author: Little-princess Leia <leia@kitty.cat>
Date: Sun Sept 23 17:30:42 2018 -0700
    Add body
commit de82ac
Author; Jamie <jam@kitty.cat>
Date :
....
```

git cherry-pick → this SHA into the tabby branch

Simpler log with:

♥ git log --oneline

```
e346d8 Add body
de82ac Edit nose
2e46cd Add gray dot on leg
```