ACM W

presents

VERSION CONTROL

WITH GIT AND GITHUB

WHAT CAN YOU EXPECT FROM SESSION 1?

1. What is Version Control? Why Version Control?
2. Learn to setup and use Git
3. Create local repositories
4. Make, stage and commit changes
5. Learn how to view changes and logs
6. Branch and merge

WHAT CAN YOU EXPECT FROM SESSION 2?

1. Setting up and using Github
2. Creating remote repositories and connecting it to your local repo
3. Pull. push and fetch
4. Collaborate
5. Exploit Github (Student Pack & Pages)

Session 1

What is version control??

Install & Setup Git

https://git-scm.com/downloads

Open Git Bash

```
git config --global user.name "<Your-Full-Name>"
git config --global user.email "<your-email-address>"
```

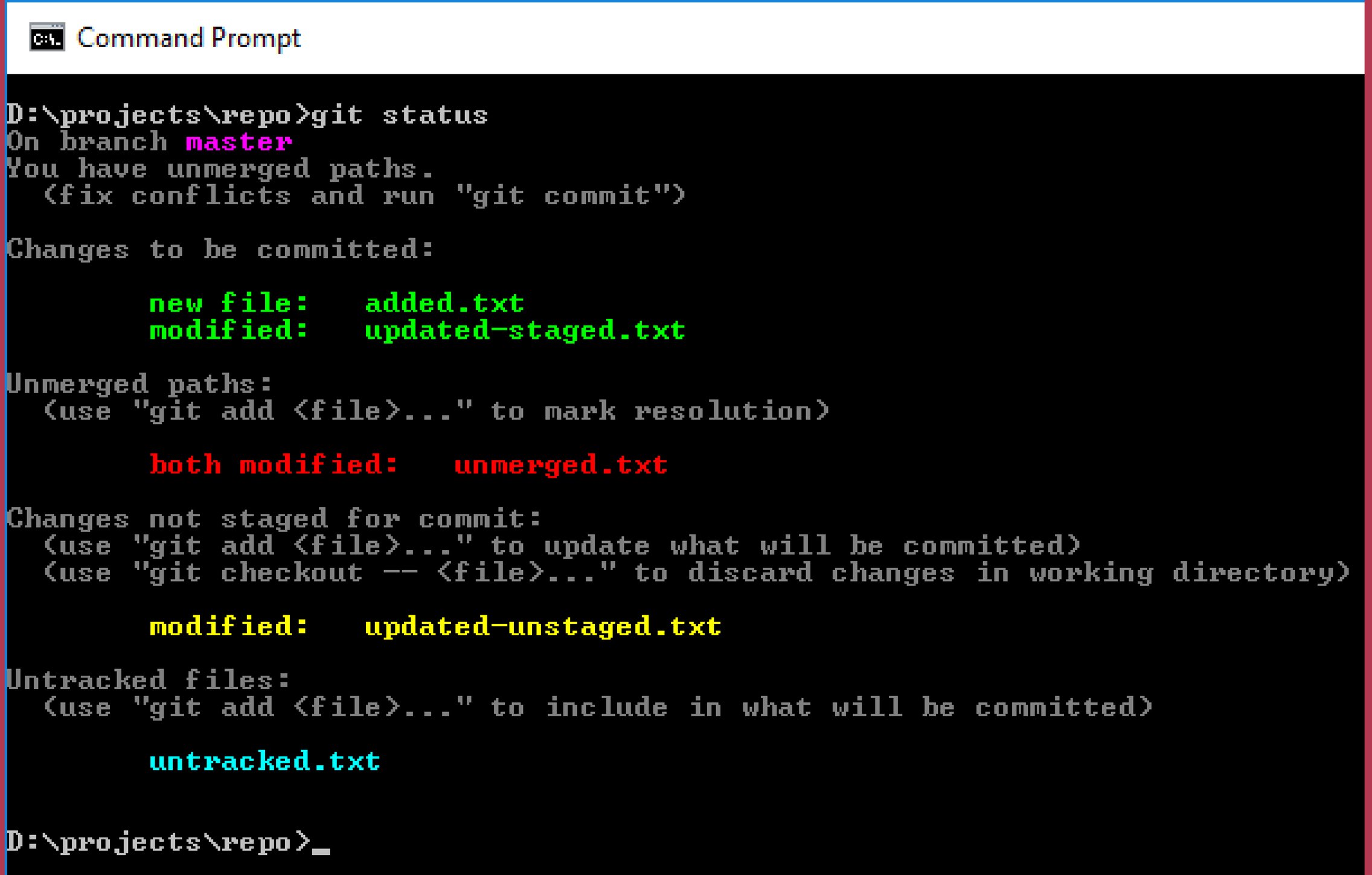
Let's create your first repository

What is Git doing?


Staging Area
Timetable.java
Working
Directory
Repository

your new best friend:

git status


Command Prompt
D:\projects\repo>git status
On branch master
You have unmerged paths.
(fix conflicts and run "git commit")
Changes to be committed:
new file: added.txt
modified: updated-staged.txt
Unmerged paths:
(use "git add <file>..." to mark resolution)
both modified: unmerged.txt
Changes not staged for commit:
(use "git add <file>..." to update what will be committed)
(use "git checkout -- <file>..." to discard changes in working directory)
modified: updated-unstaged.txt
Untracked files:
(use "git add <file>..." to include in what will be committed)
untracked.txt
D:\projects\repo>

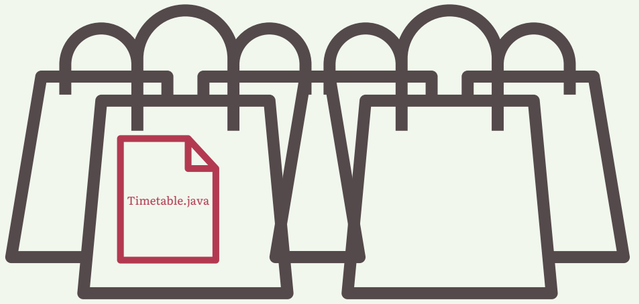
```
git add Timetable.java
```


Staging Area
Timetable.java
Working
Directory
Repository

git commit

Staging Area

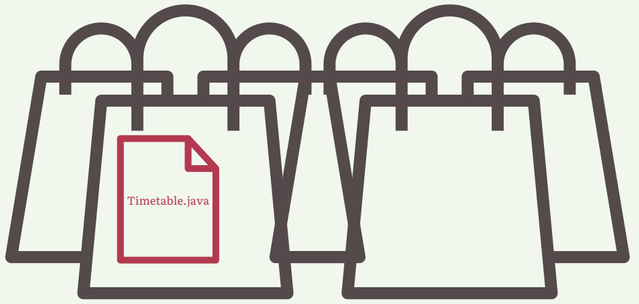
Repository

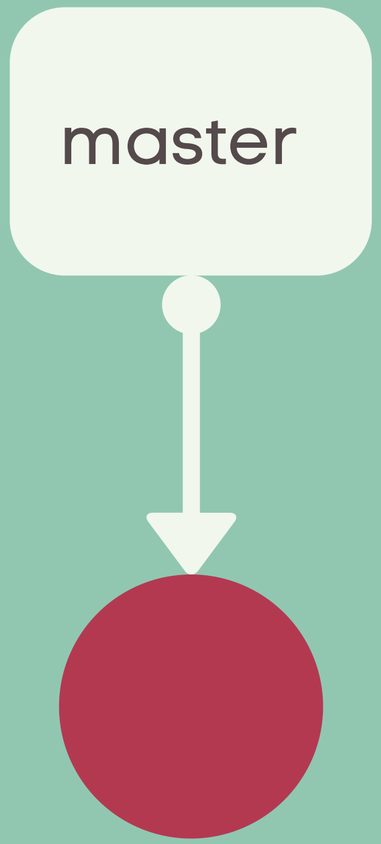


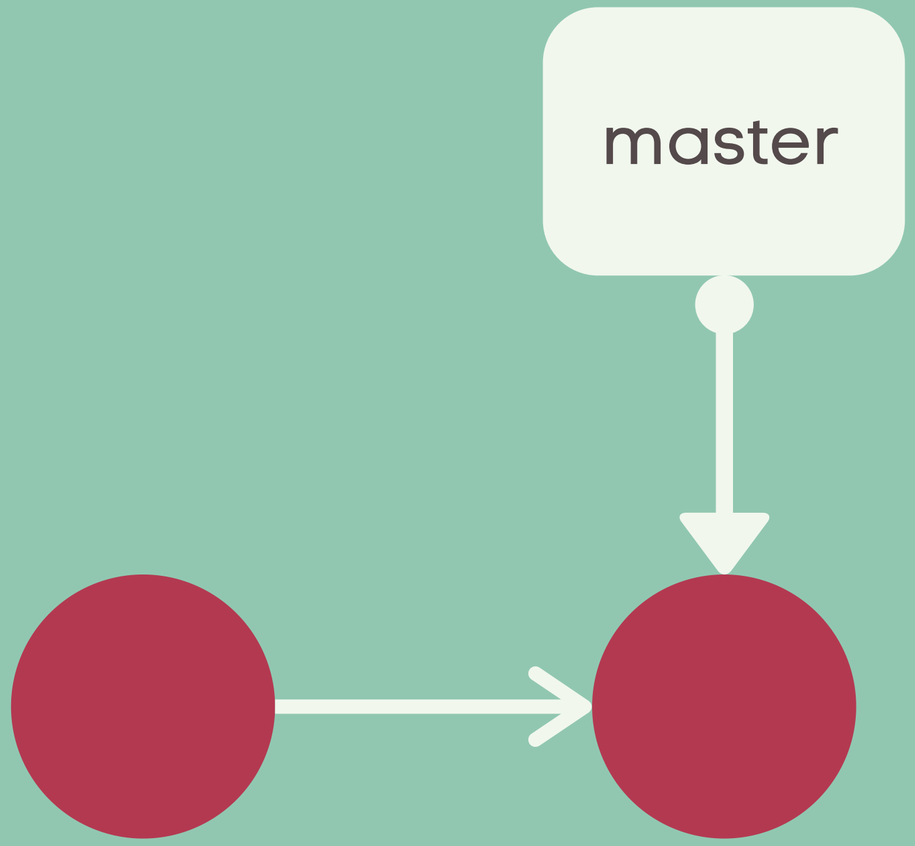
Nothing happens to the files in the working directory. Only a snapshot of them is stored in the repo.

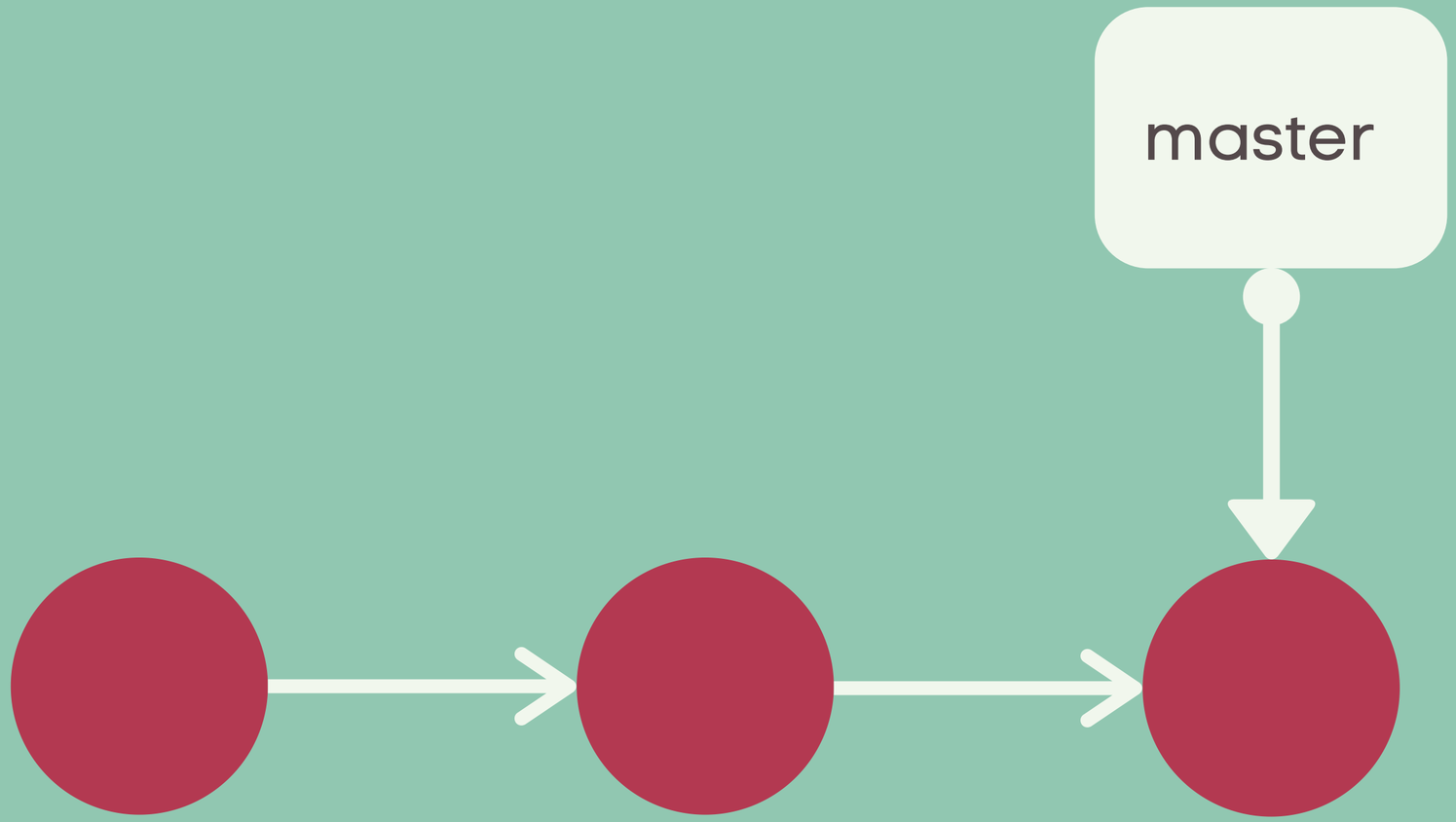
Staging Area

Repository



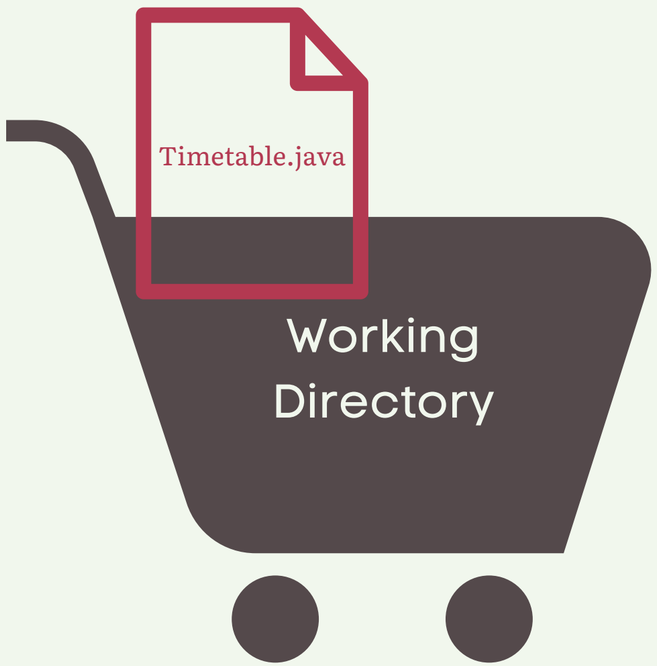

master

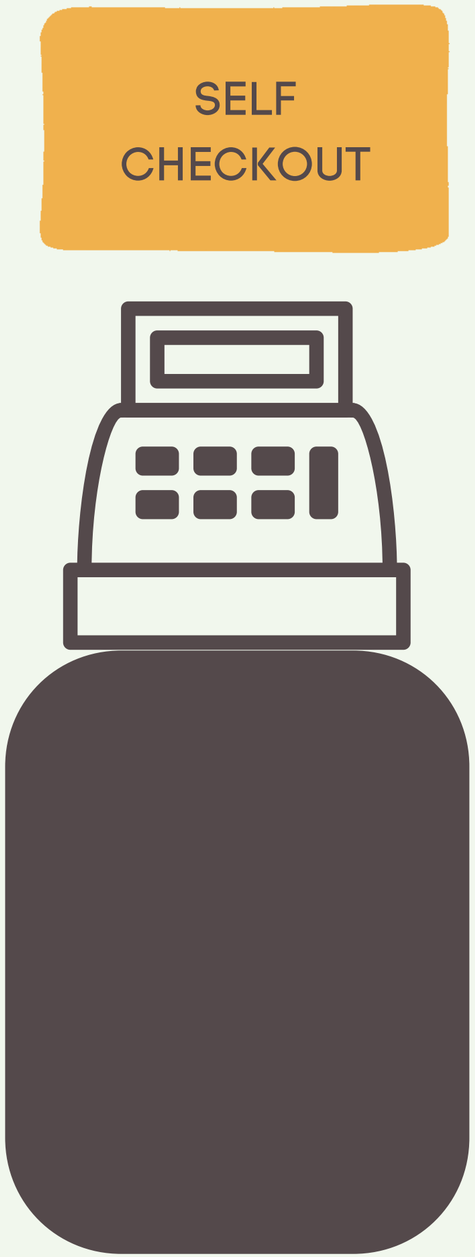

master

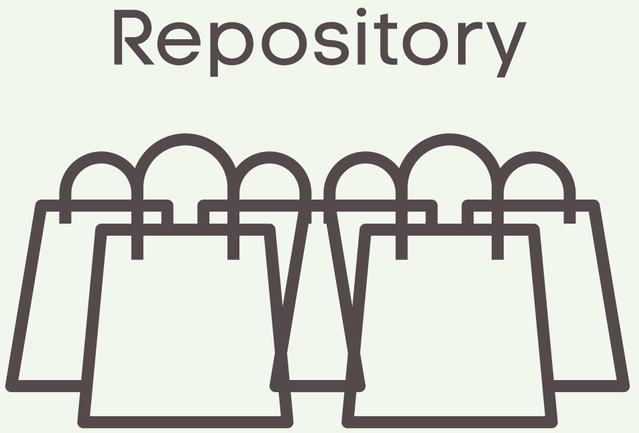

master

Commit options:

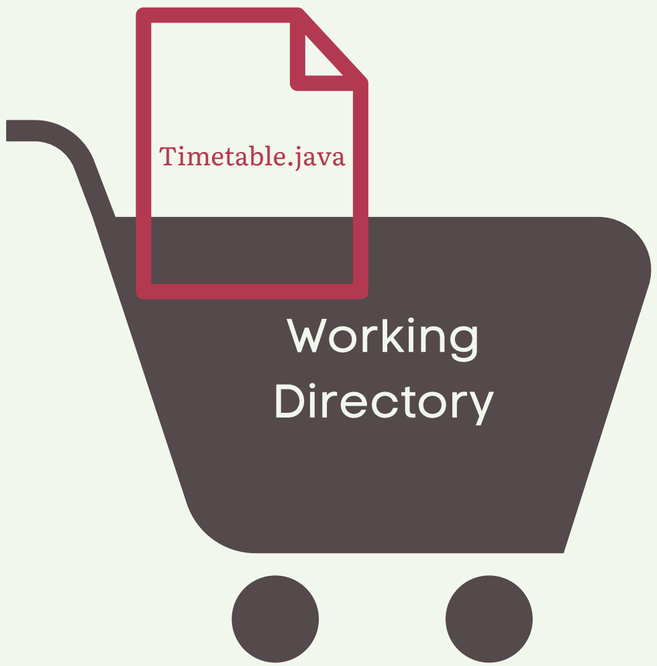
Commit options:
git commit -a

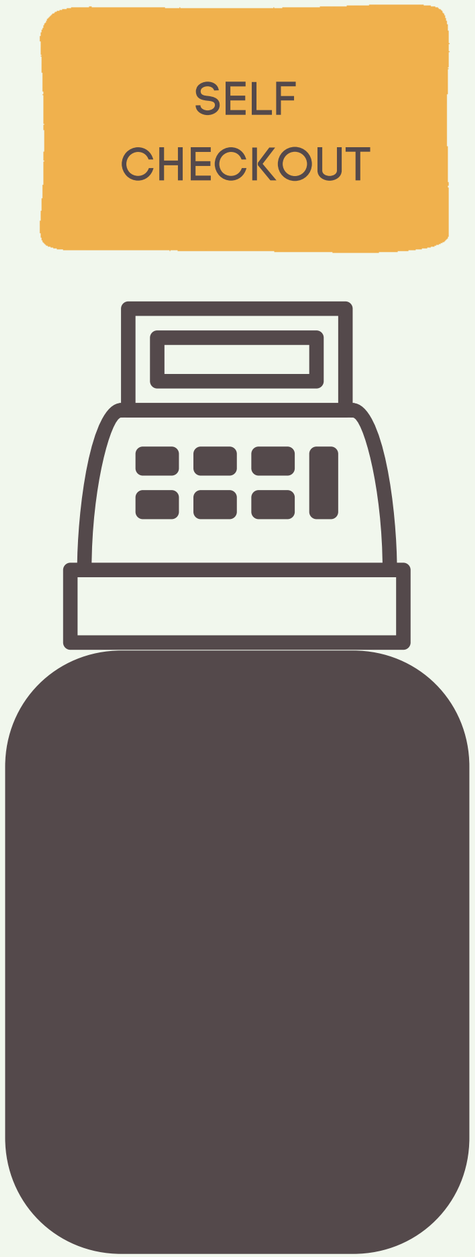


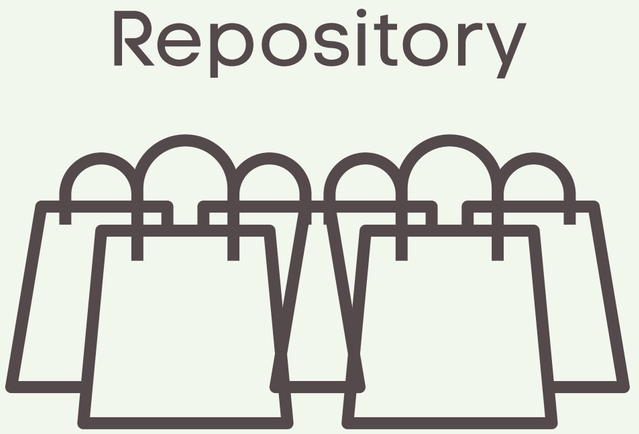




Commit options:
git commit -a







```
git commit -m "commit message here"
```

Let's complicate things

What do commits look like? How can I see what I've done?

git bash in a directory you want to place the repo

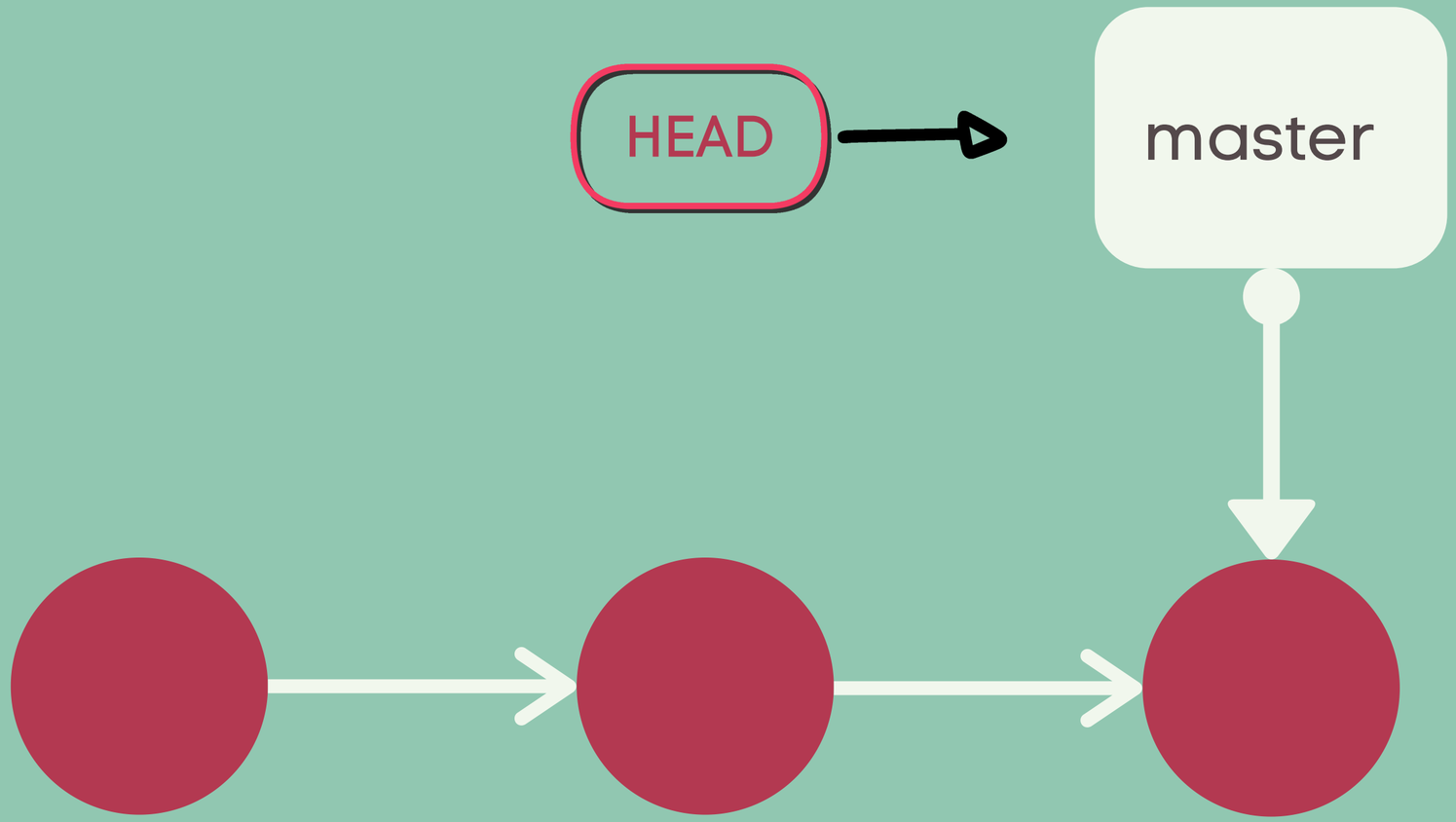
git clone

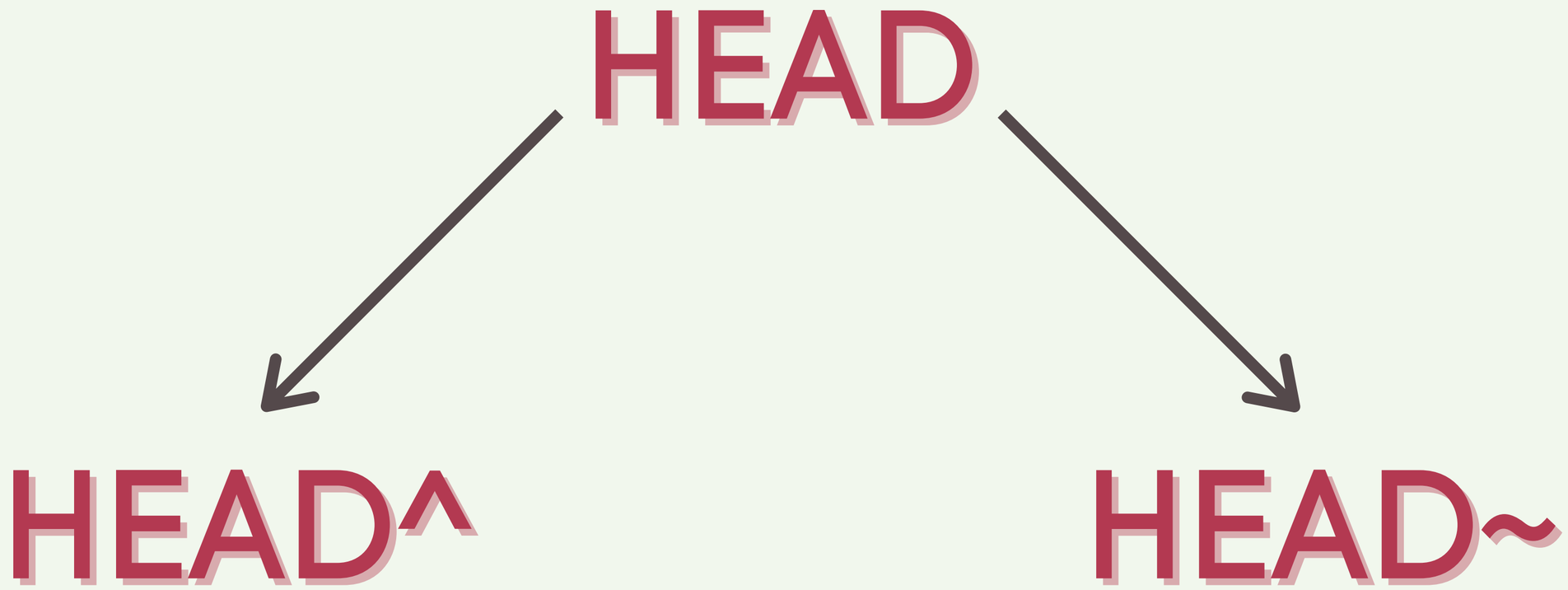
https://github.com/suhaamber/ACMW-Git-Workshop.git

jack of all trades, master of none, but better than a master of one

git log

```
git log --oneline
git log <filename>
git log --graph
git log --author="<user.name>"
git log --branch="<branchname>"
git log --branch="*"
git log --grep="<pattern>"
git log --stat
git log --patch
```


HEAD
master

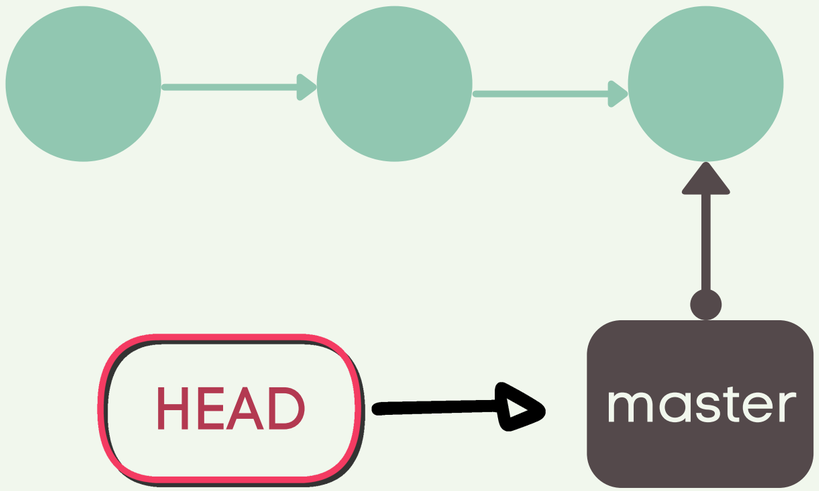

HEAD
HEAD^
HEAD~

git diff

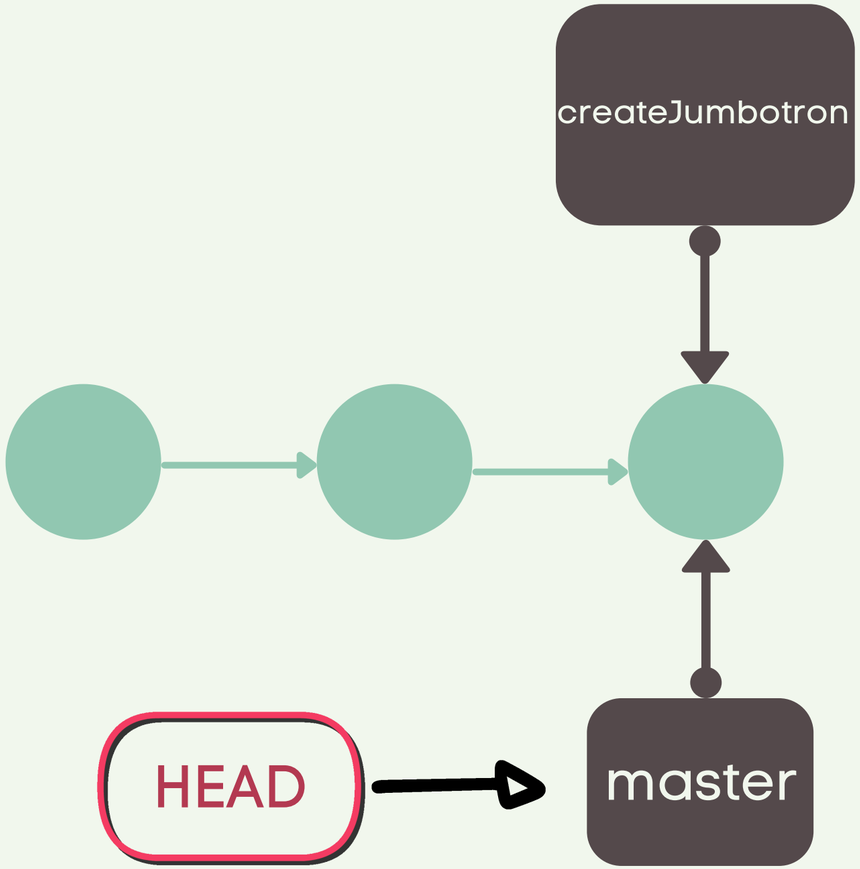
Let's complicate things...more (for the last time).

Ever written a risky code?

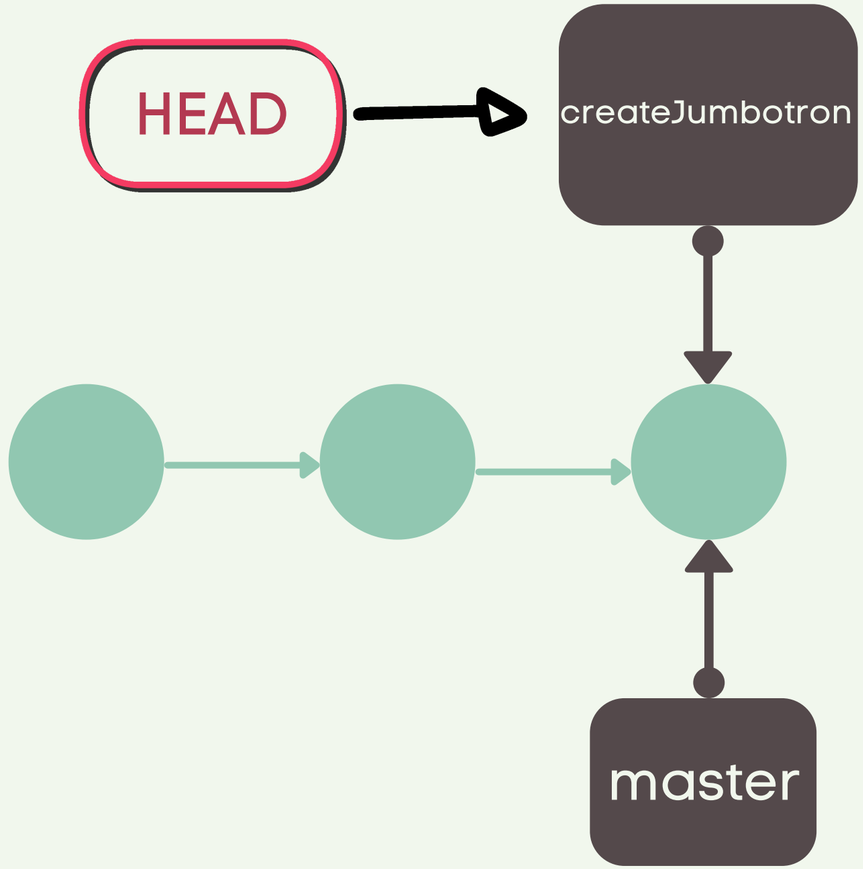
BRANCHING


HEAD
master

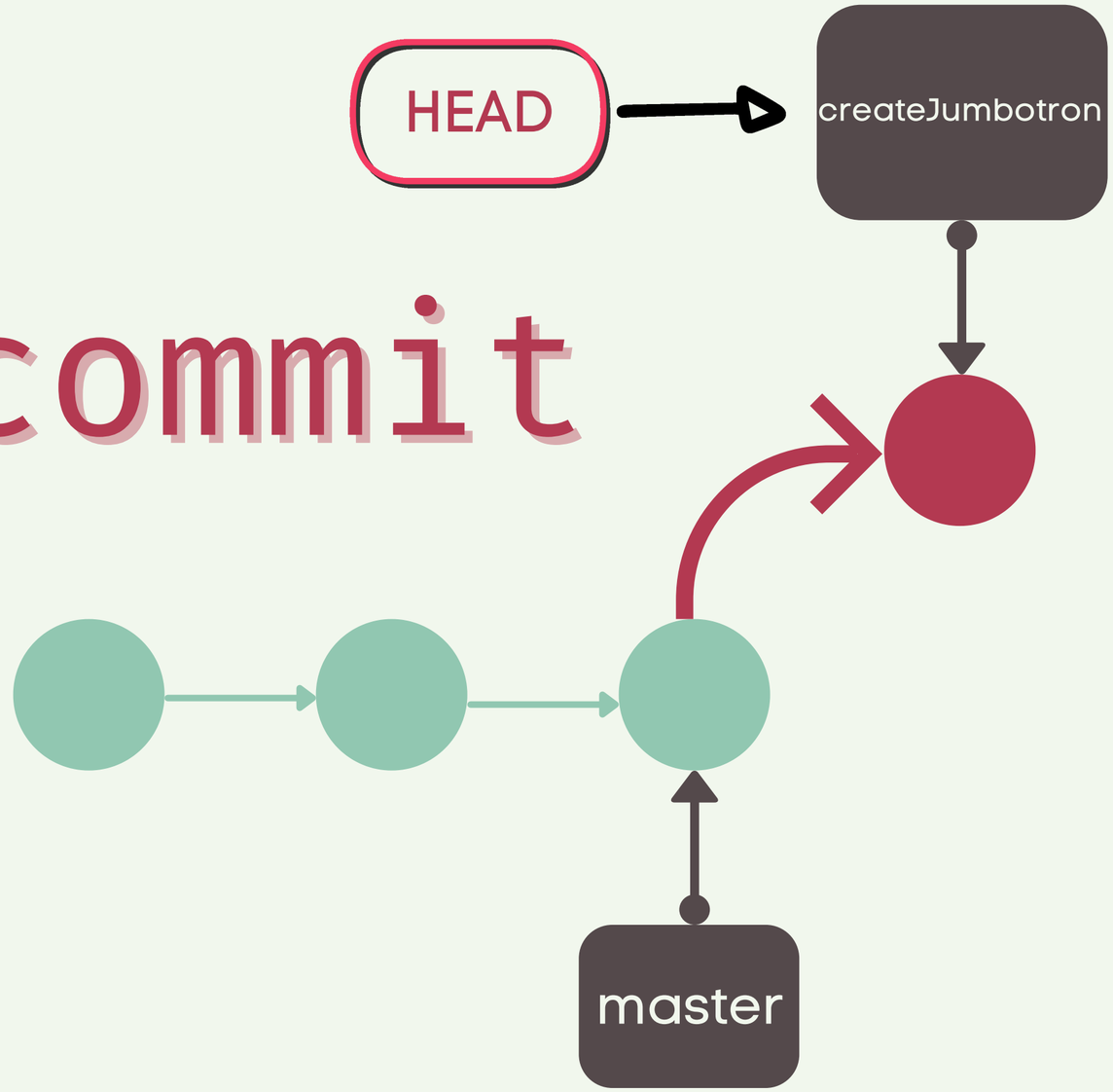
git branch createJumbotron



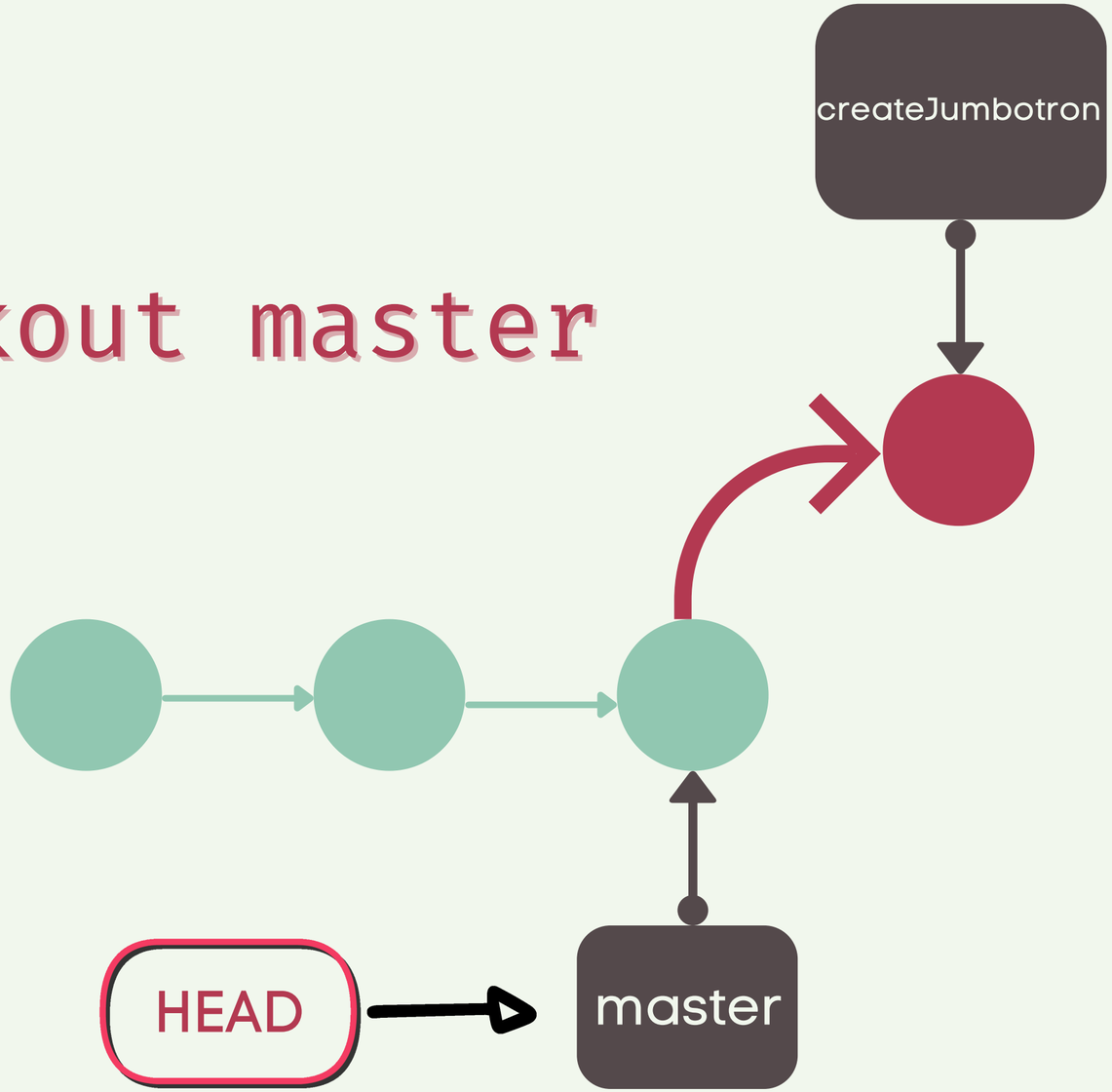
git checkout createJumbotron



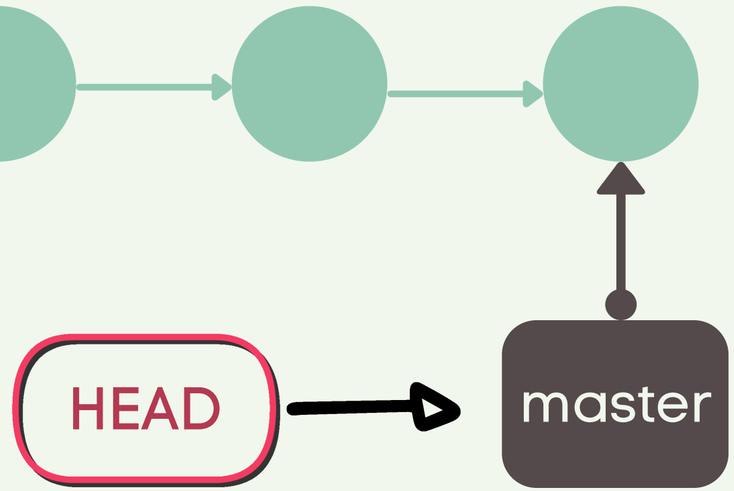
git commit



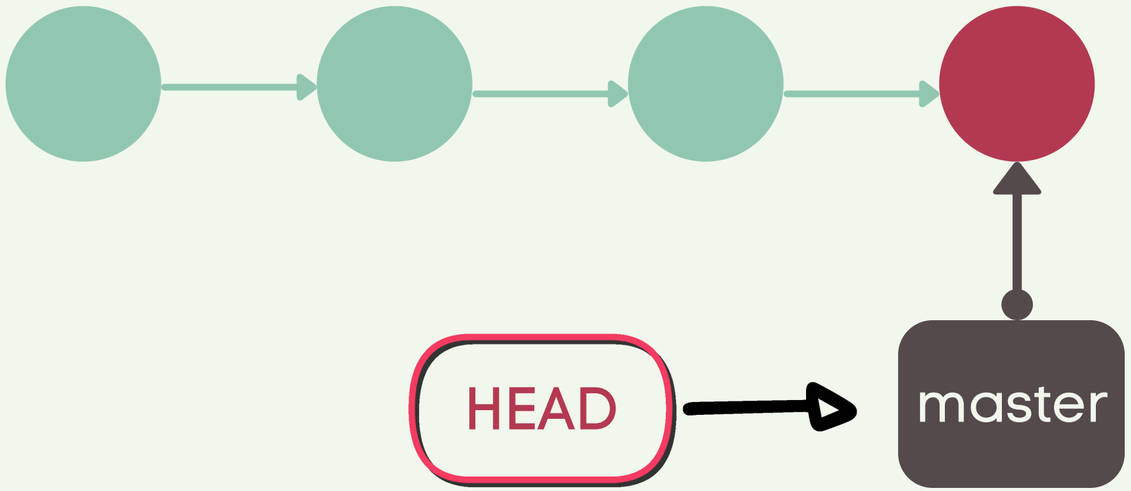
git checkout master



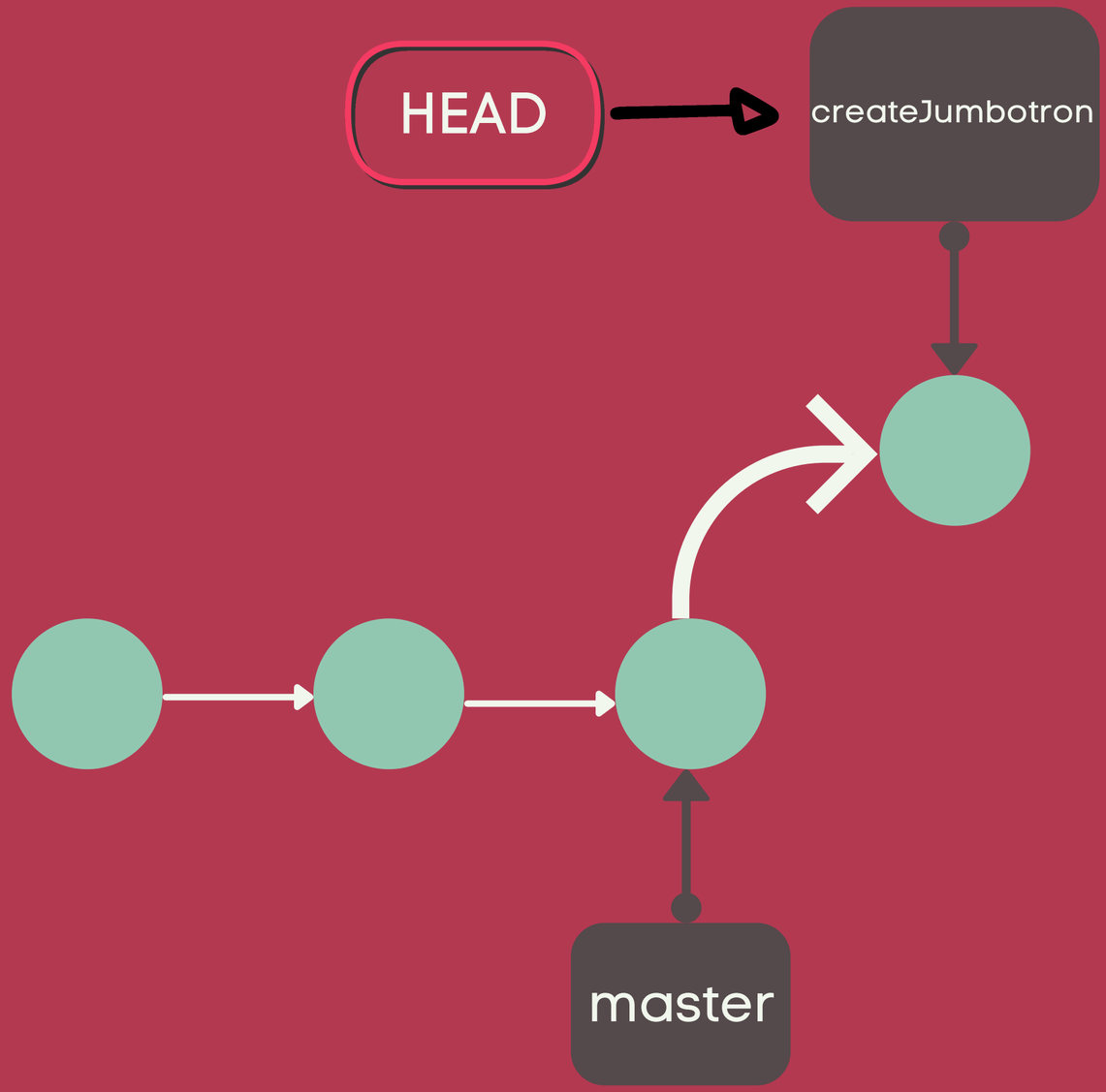
WHAT HAPPENS IF YOU DONT BRANCH

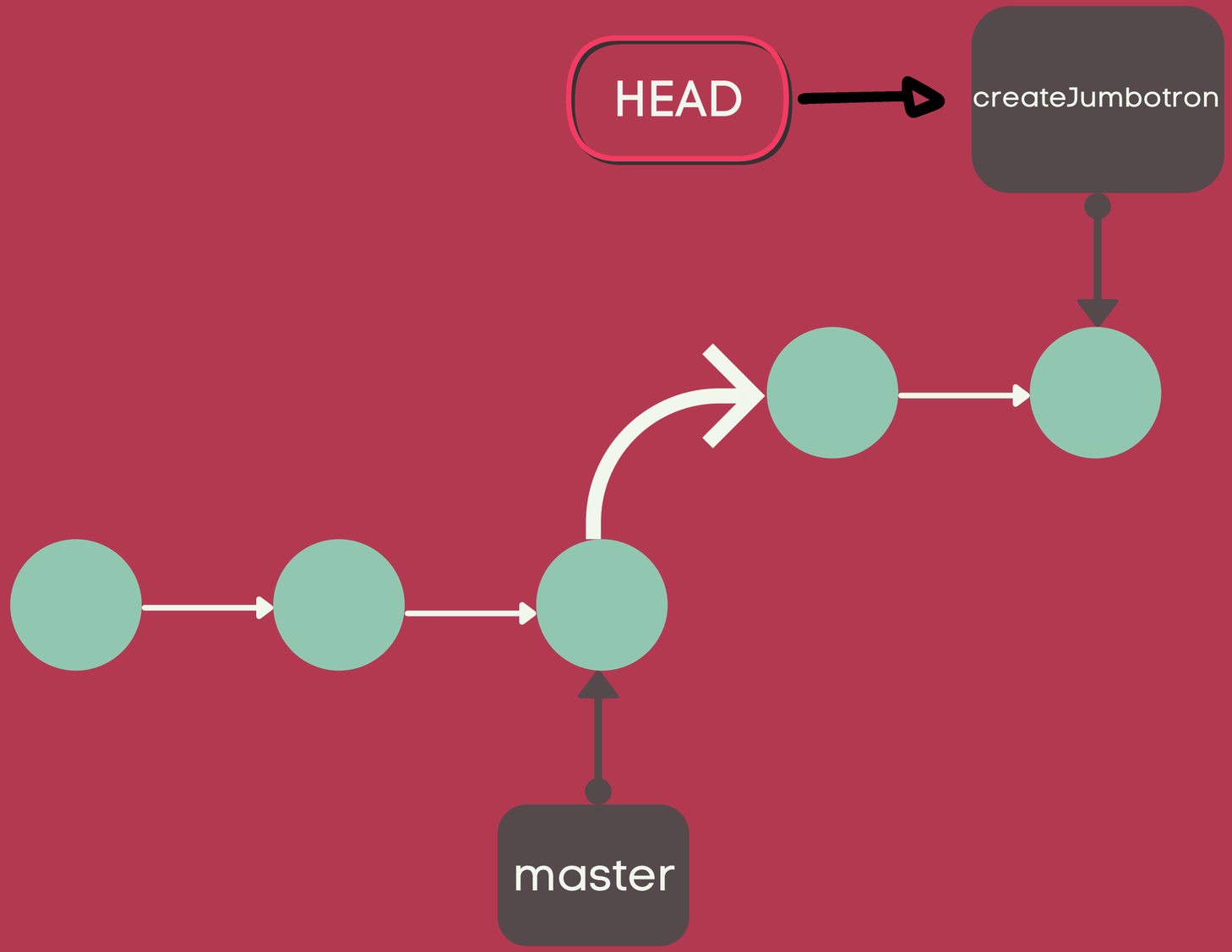


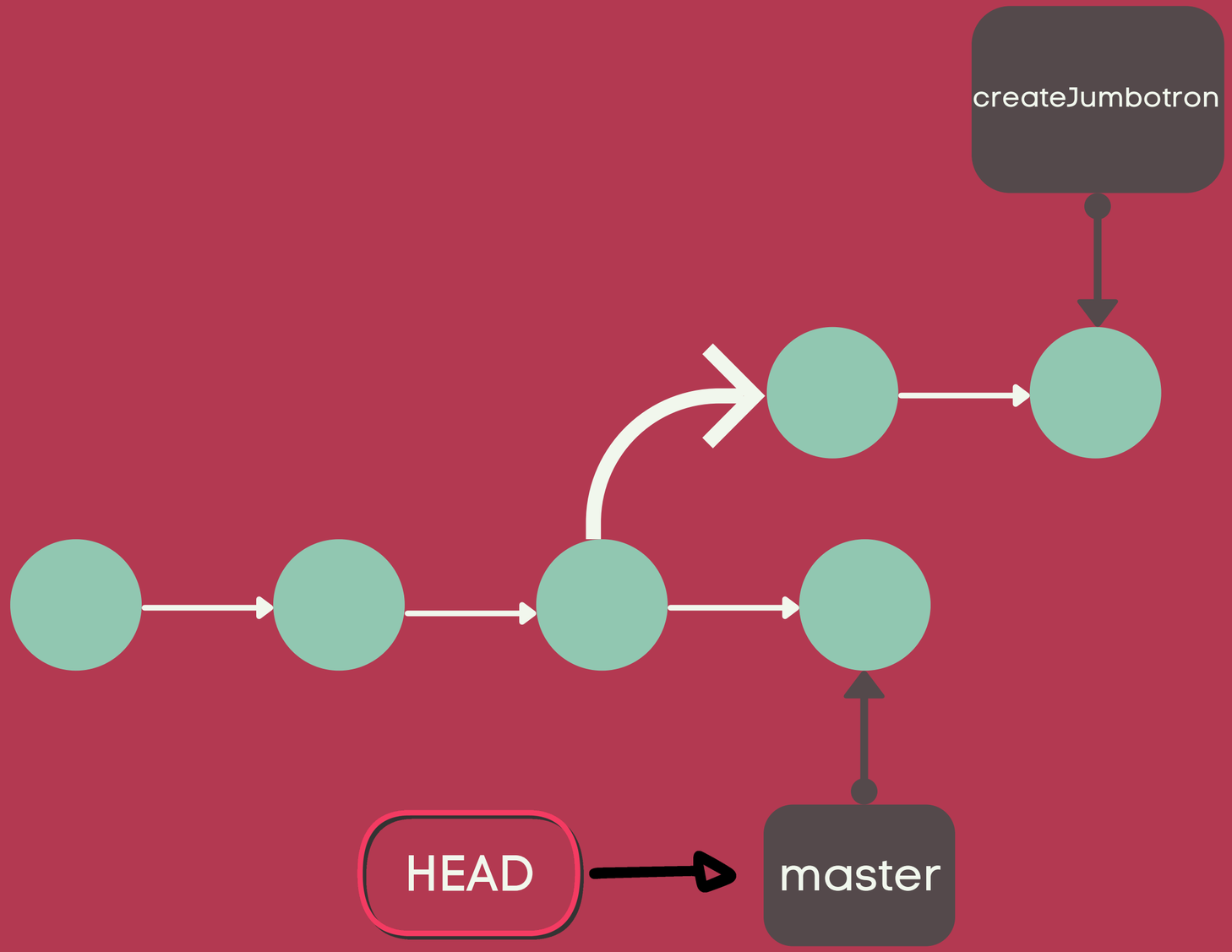
WHAT HAPPENS IF YOU DONT BRANCH



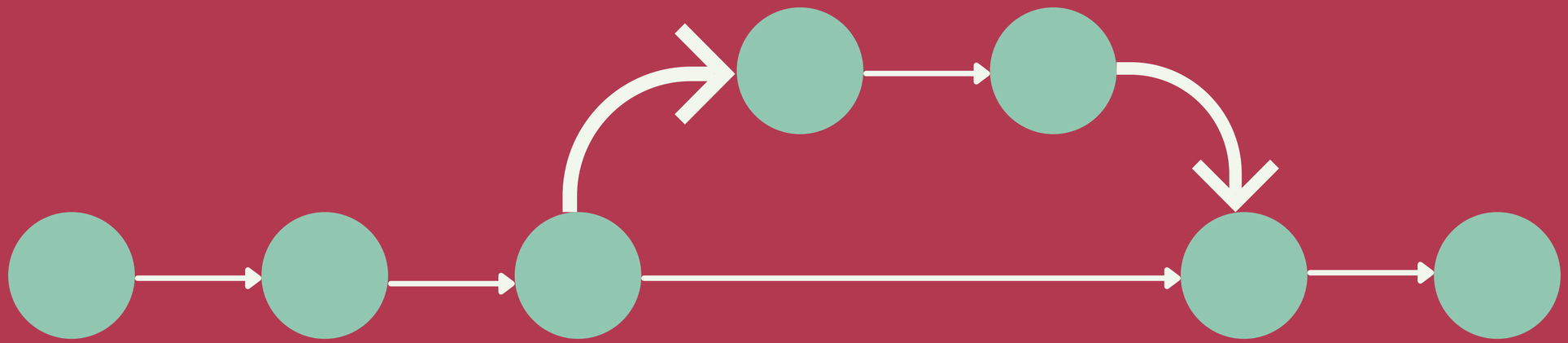
MERGING


HEAD
createJumbotron
master

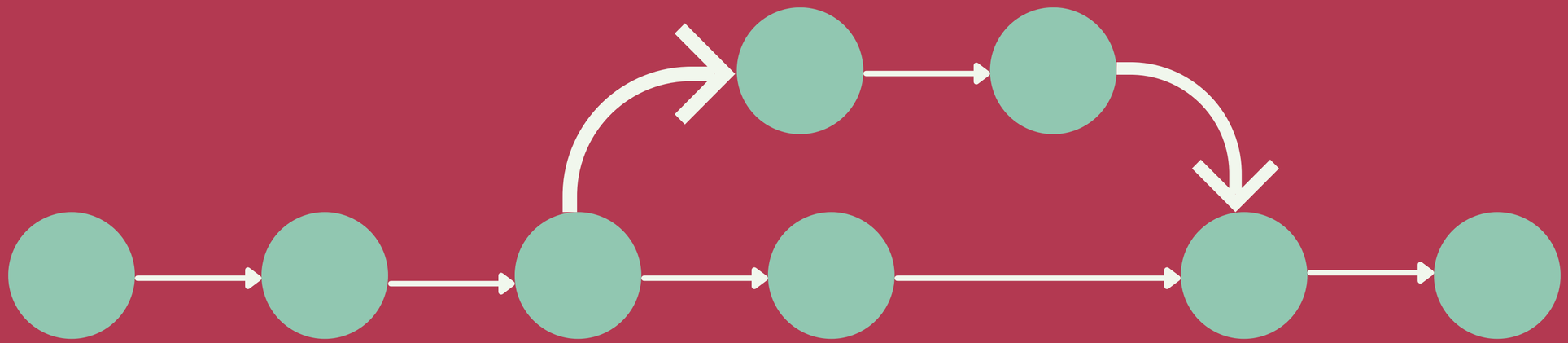

HEAD
createJumbotron
master


createJumbotron
HEAD
master

A fast-forward merge



A 3-way merge



end of Session 1

Session 2

Have you created your Github account?

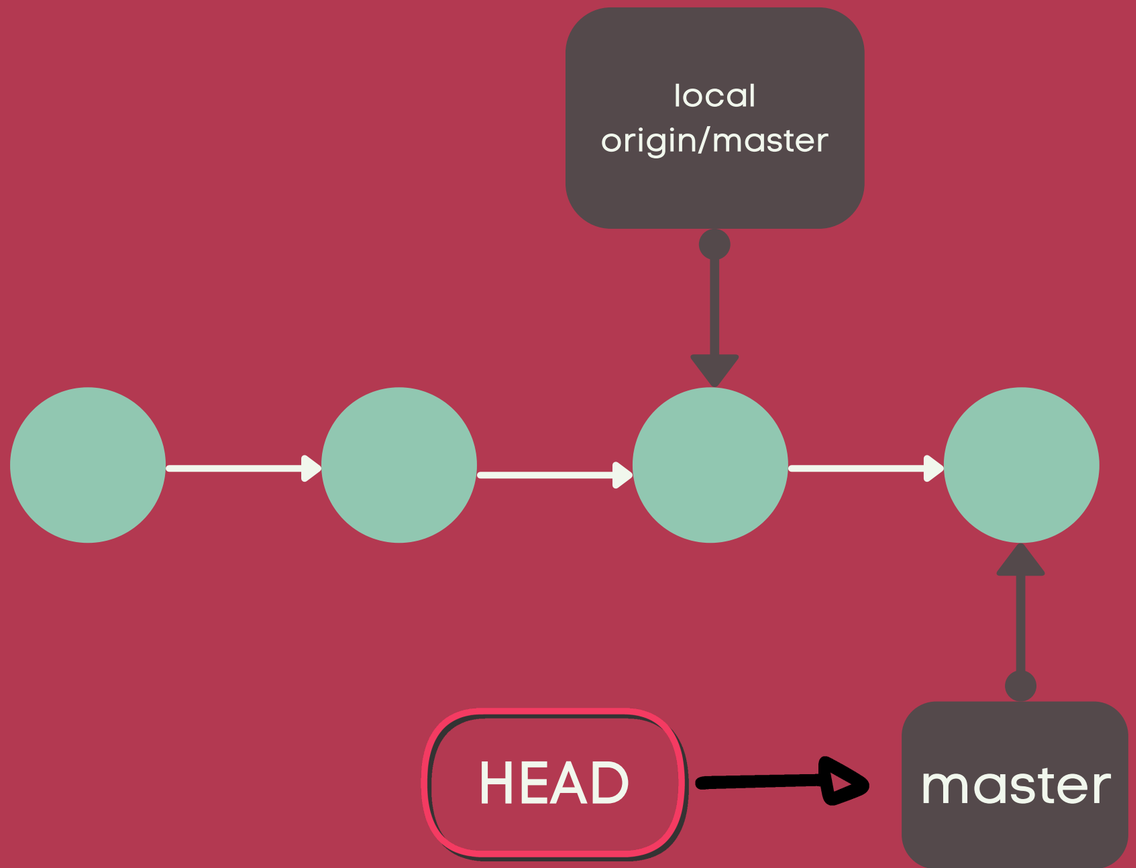
Let's get our hands dirty

clone session 2 repository

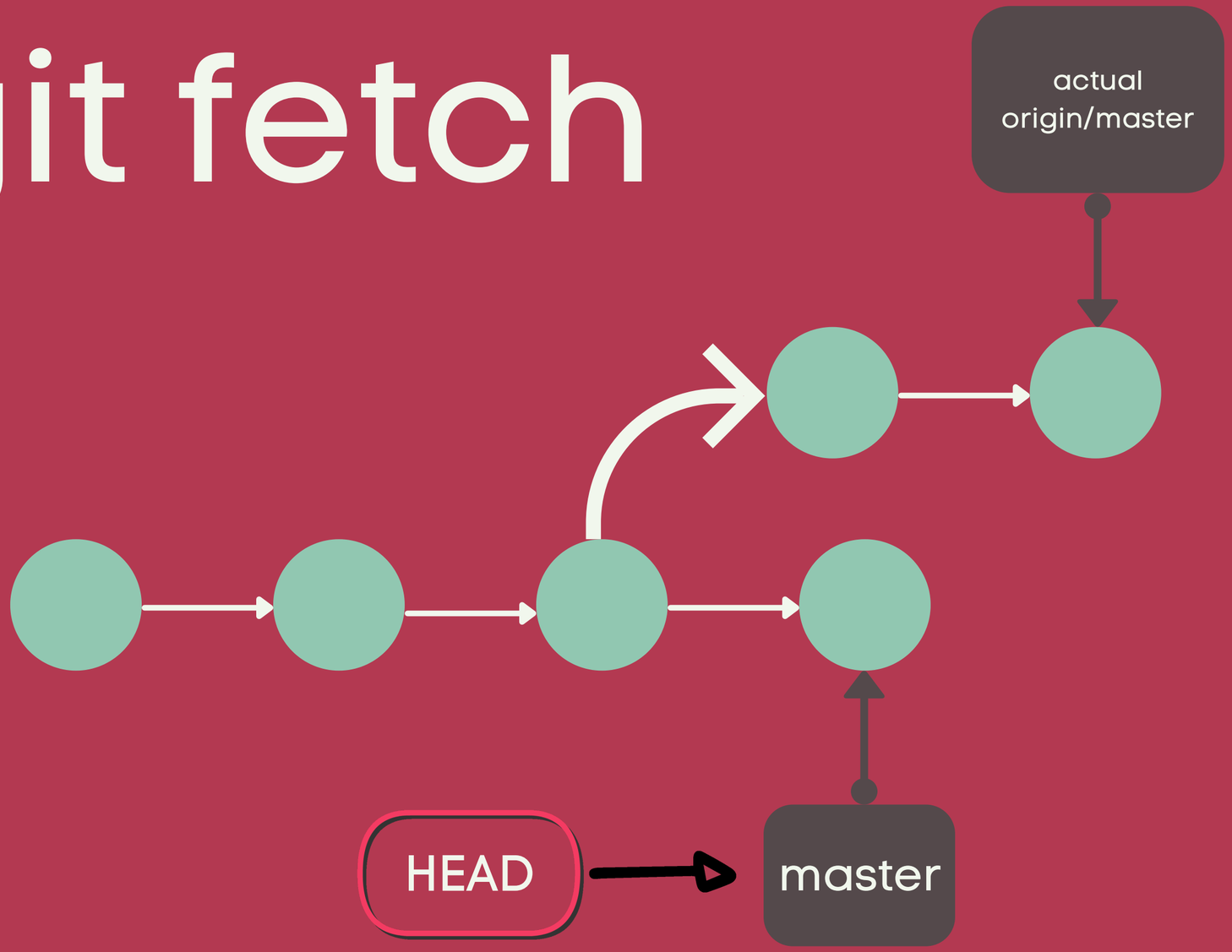
git clone https://github.com/suhaamber/TimetableBuilder.git

we've seen git remote and git push

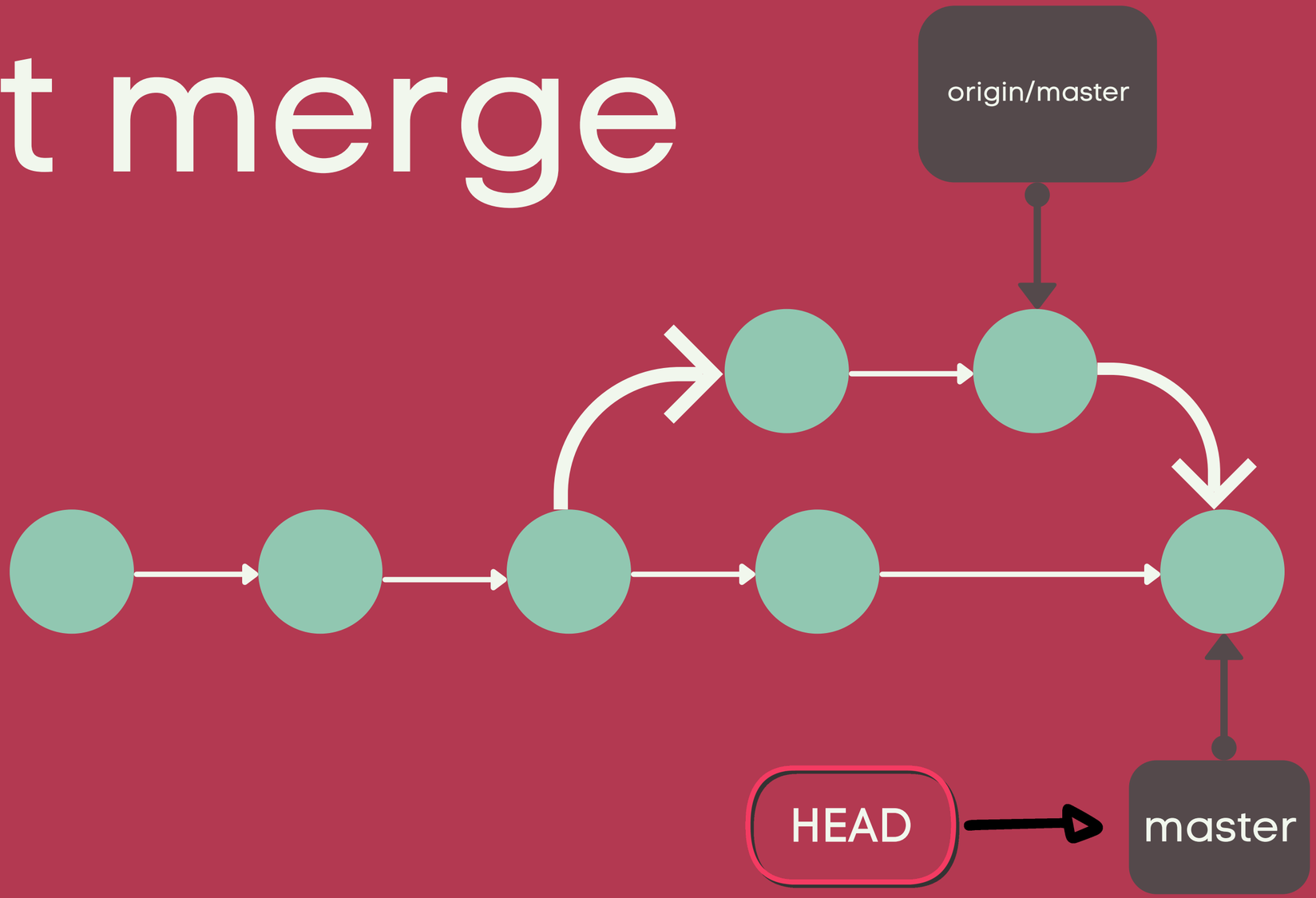
git fetch and git pull


local
origin/master
HEAD
master

git fetch

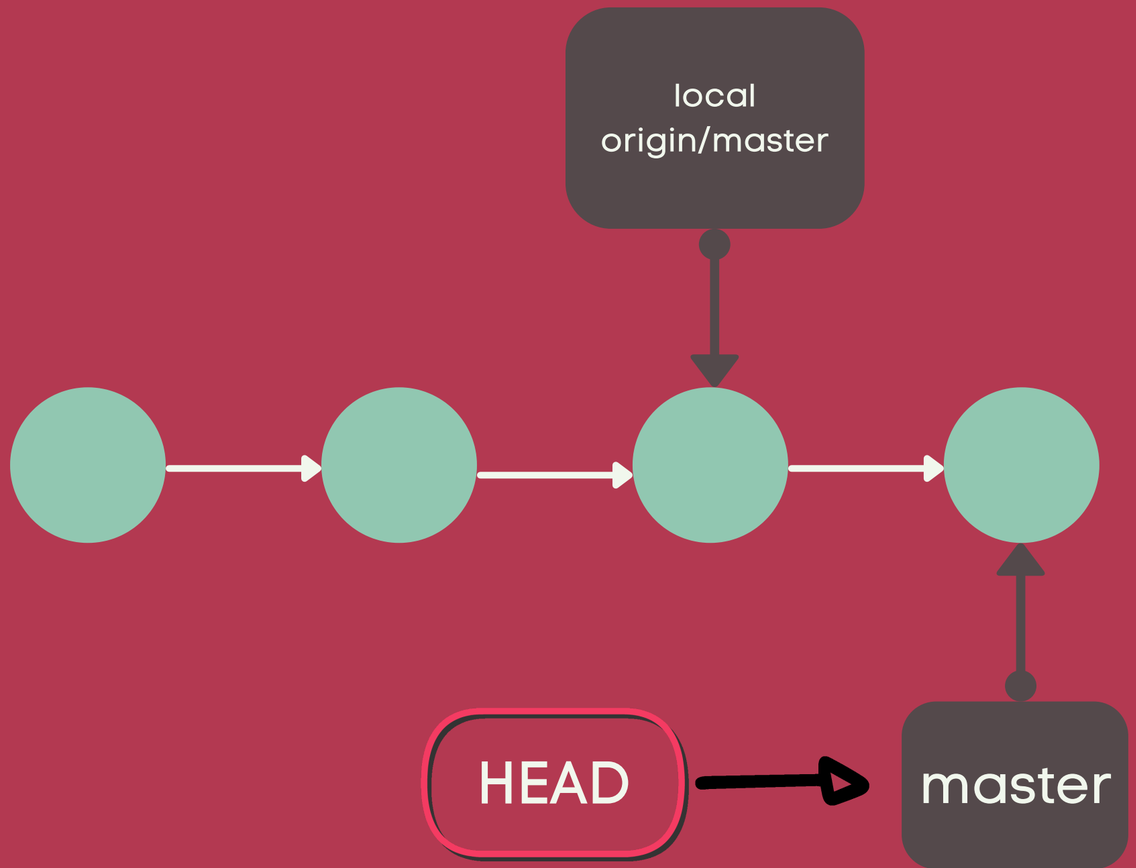


git merge

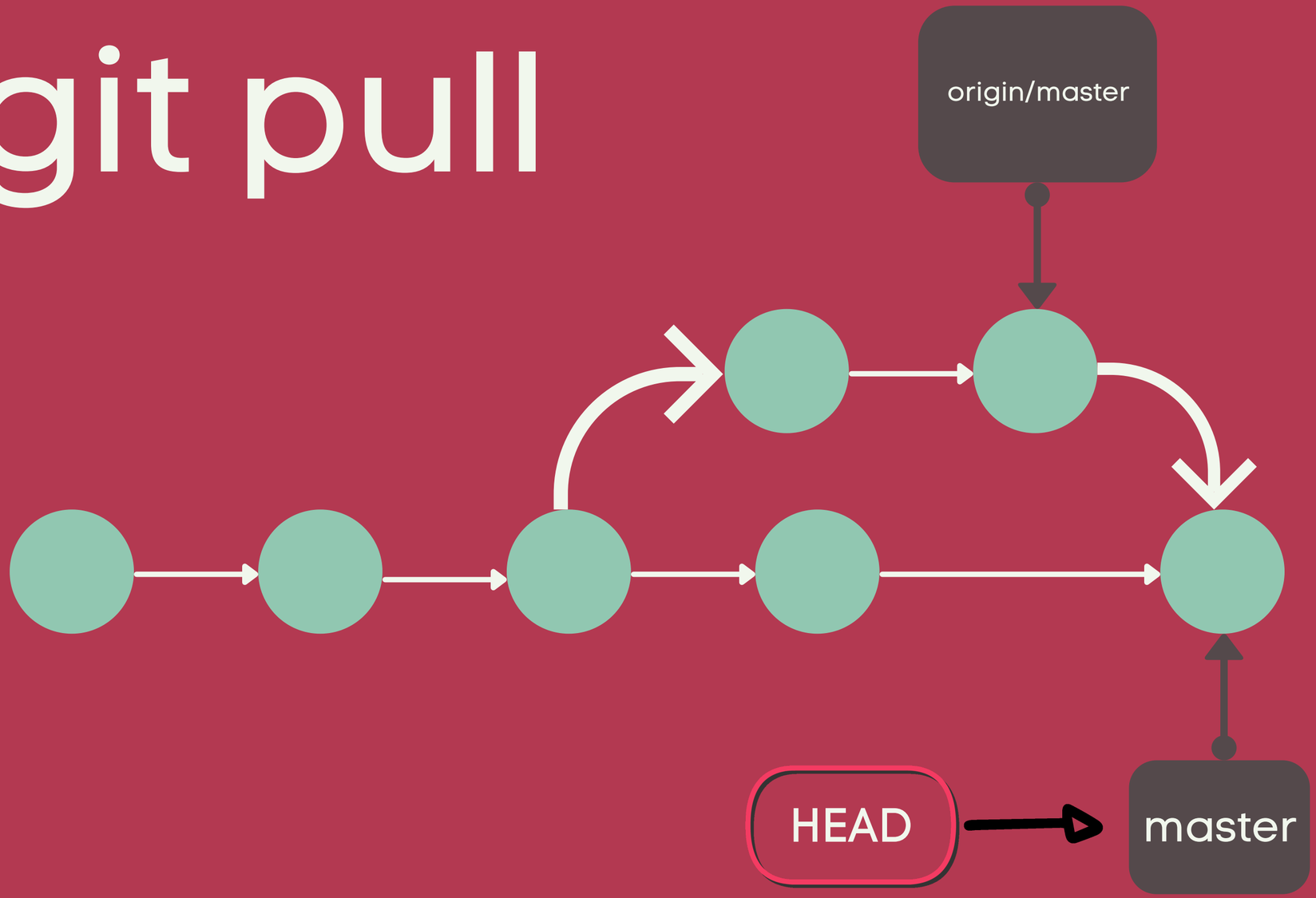


git pull =

git fetch + git merge


local
origin/master
HEAD
master

git pull



collaborate with pull requests

undoing things

github pages

github student pack

that's all folks