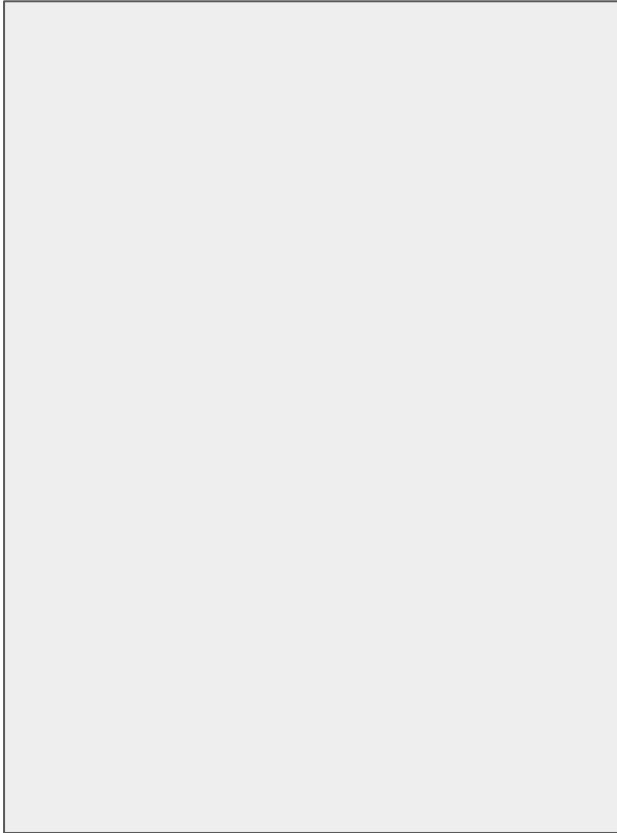


git projects

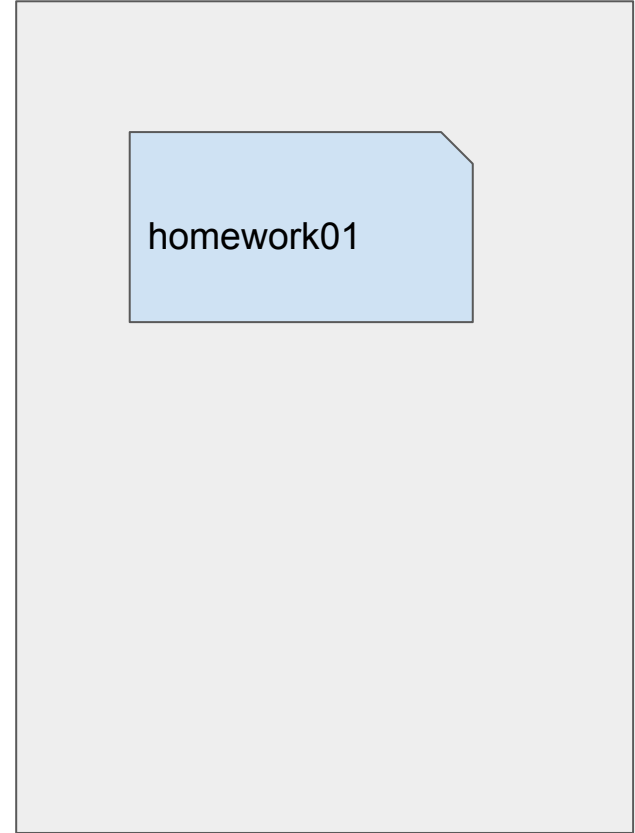
Starting from a remote

Starting from Remote: Step 1 - create repo on Github

My computer (aka “local”)



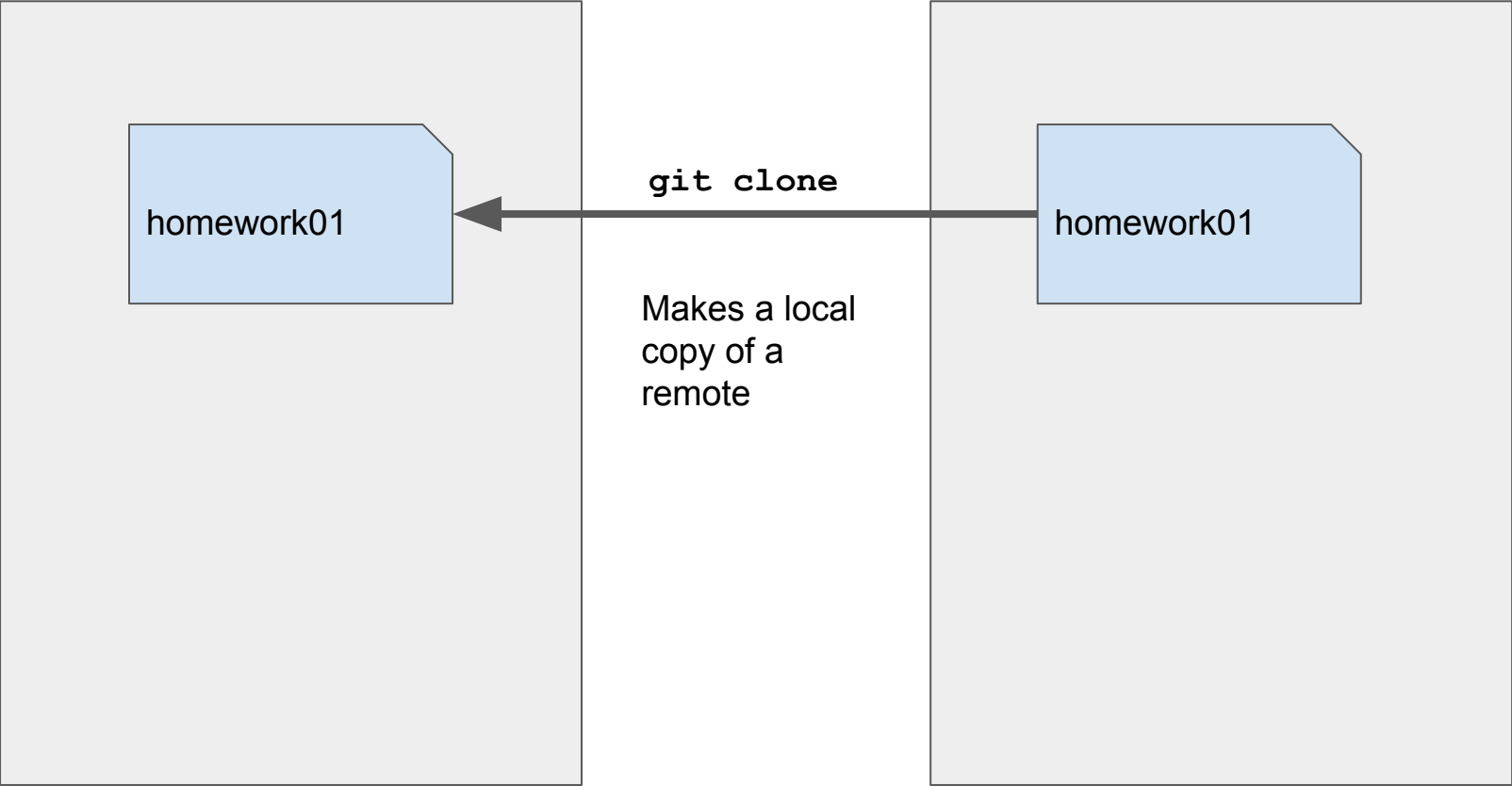
Github (aka remote)



Starting from Remote: Step 2 - clone repo from Github

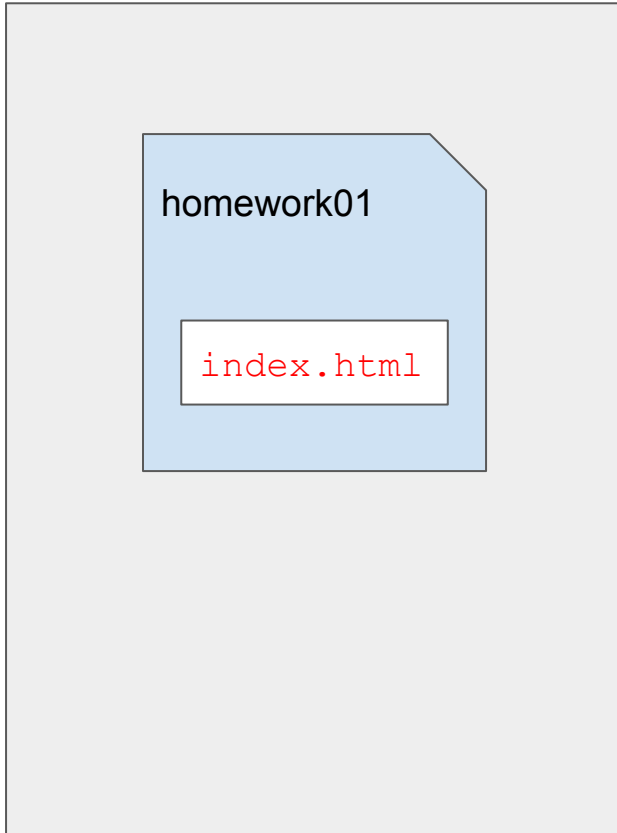
My computer (aka "local")

Github (aka remote)



Starting from Remote: Step 3 - make local changes

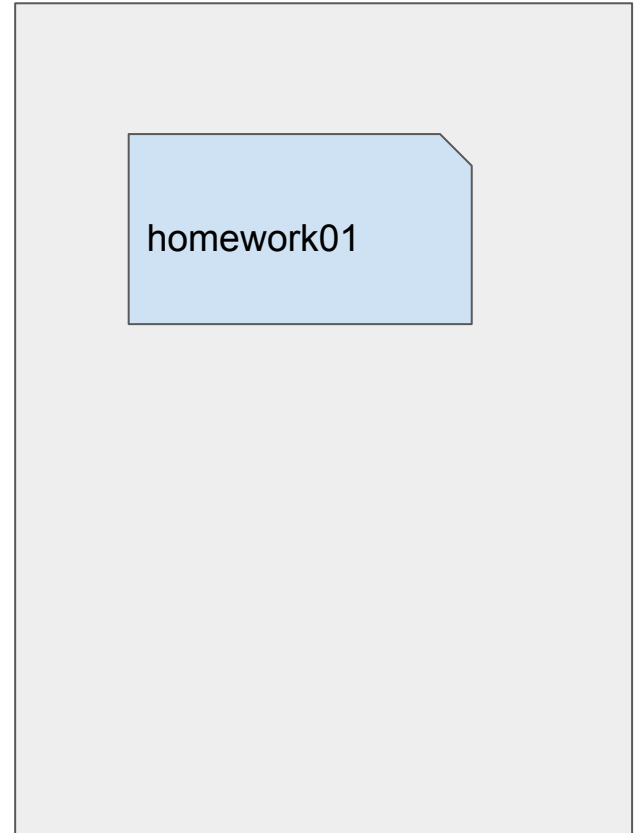
My computer (aka "local")



Git is interested in
changes to the project

- New files
- File edits
- Deleted files

Github (aka remote)



Starting from Remote: Step 4 - stage local changes with `git add`

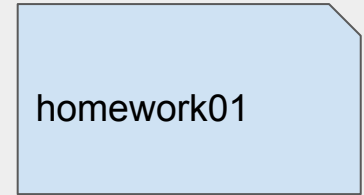
My computer (aka "local")



```
git add index.html
```

Staging changes is like putting things in a shopping cart: I am choosing the changes I want to keep

Github (aka remote)



Starting from Remote: Step 5 - keep local changes with `git commit`

My computer (aka "local")

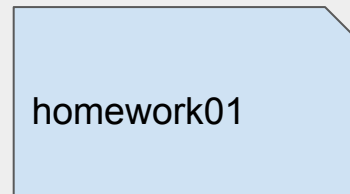


```
git commit -m  
"Adds index.html"
```

Committing changes
is like making your
purchase. You have
decided to keep them
changes.

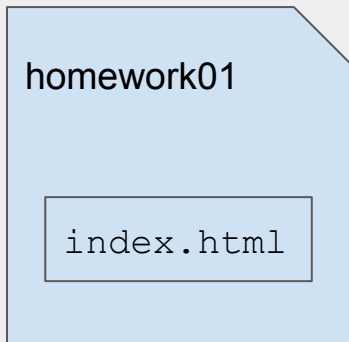
At this point it is still
relatively easy to roll
back or modify your
commit. It hasn't
affected your remote
(i.e., shared) project.

Github (aka remote)



Starting from Remote: Step 6 - update from remote with `git pull`

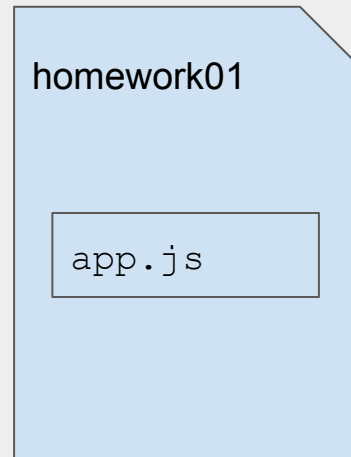
My computer (aka “local”)



In a team, the **remote** is changing. If you have changed files that someone has already changed, git will prevent you from overwriting their published changes.

Best practice is to pull changes before publishing with a push.

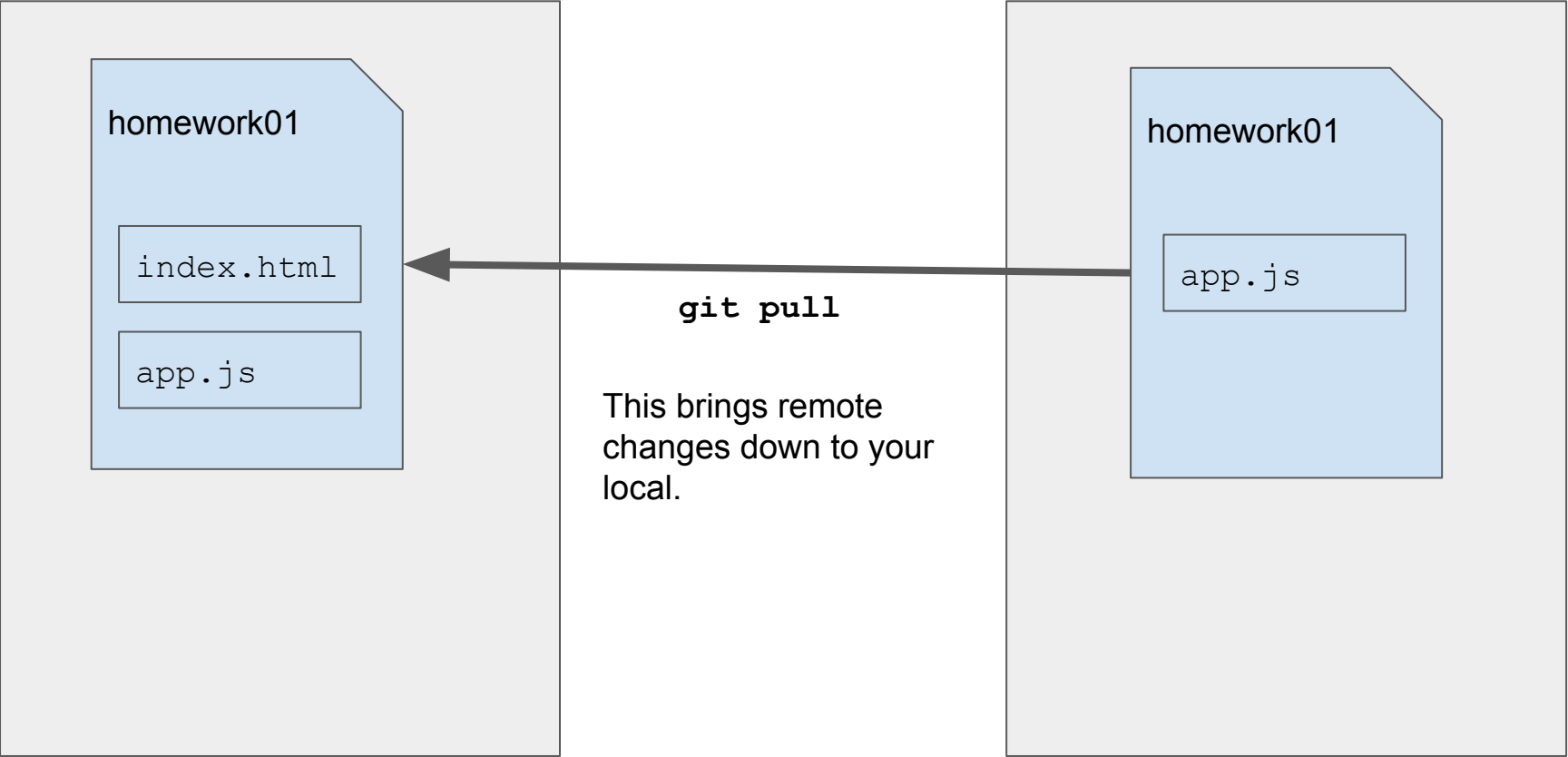
Github (aka remote)



Starting from Remote: Step 6 - update from remote with `git pull`

My computer (aka “local”)

Github (aka remote)



Starting from Remote: Step 7 - publish changes with `git push`

My computer (aka “local”)

homework01

index.html

app.js

Pushing changes
publishes them for the
world to see, review,
and use.

`git push`

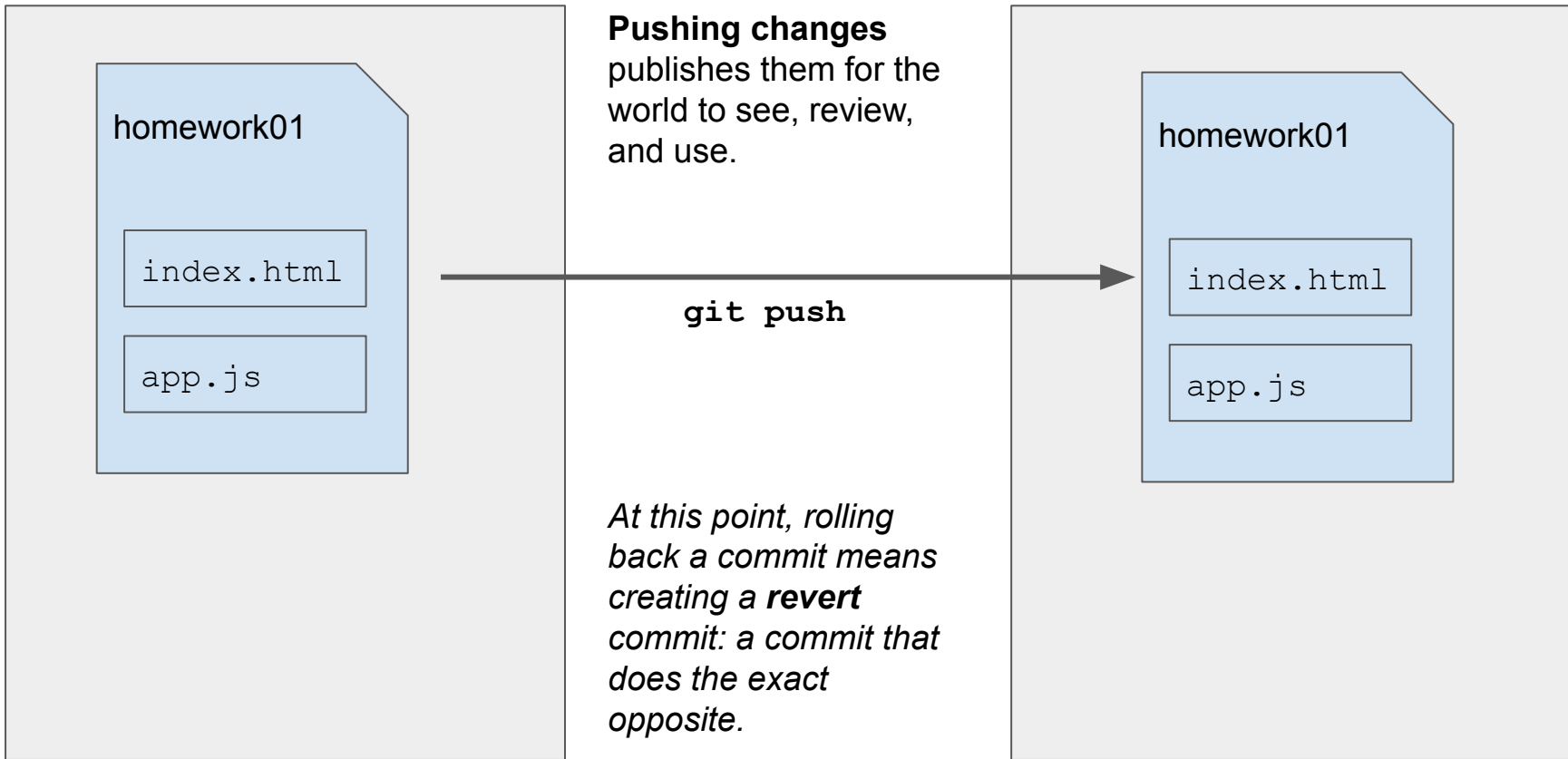
*At this point, rolling
back a commit means
creating a **revert**
commit: a commit that
does the exact
opposite.*

Github (aka remote)

homework01

index.html

app.js

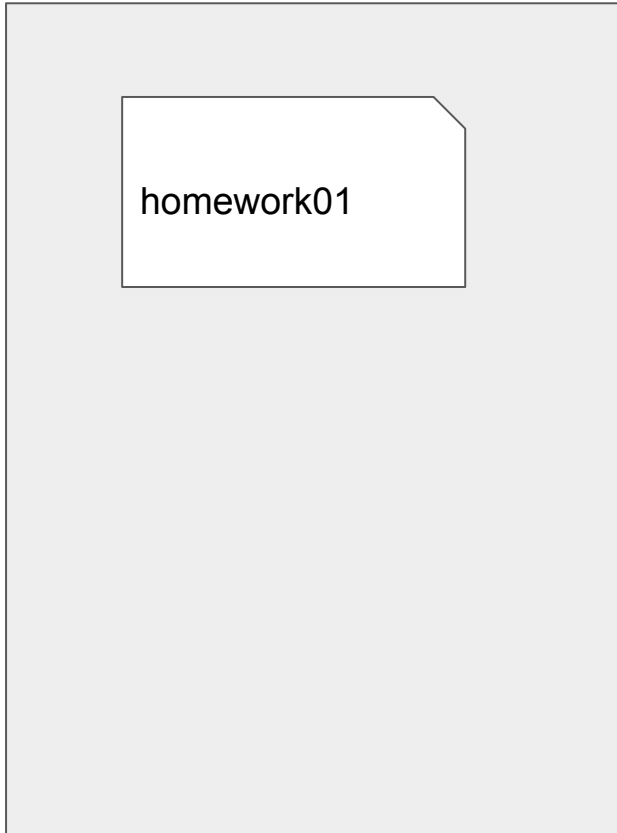


git projects

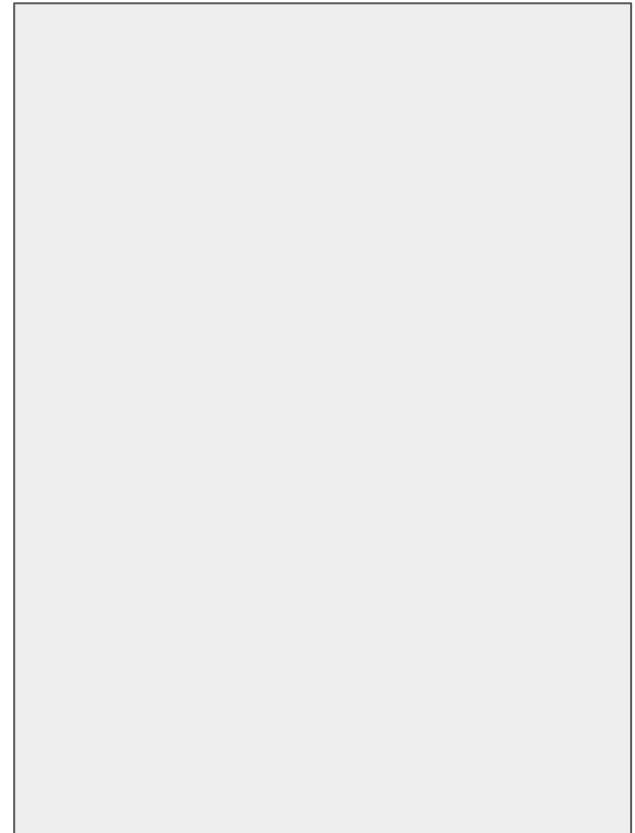
Starting from local

Starting from Remote: Step 1 - create a folder on your computer

My computer (aka “local”)



Github (aka remote)



Starting from Remote: Step 2 - `git init` a repo on your computer

My computer (aka “local”)



homework01

```
git init
```

This starts tracking changes that happen inside this folder.

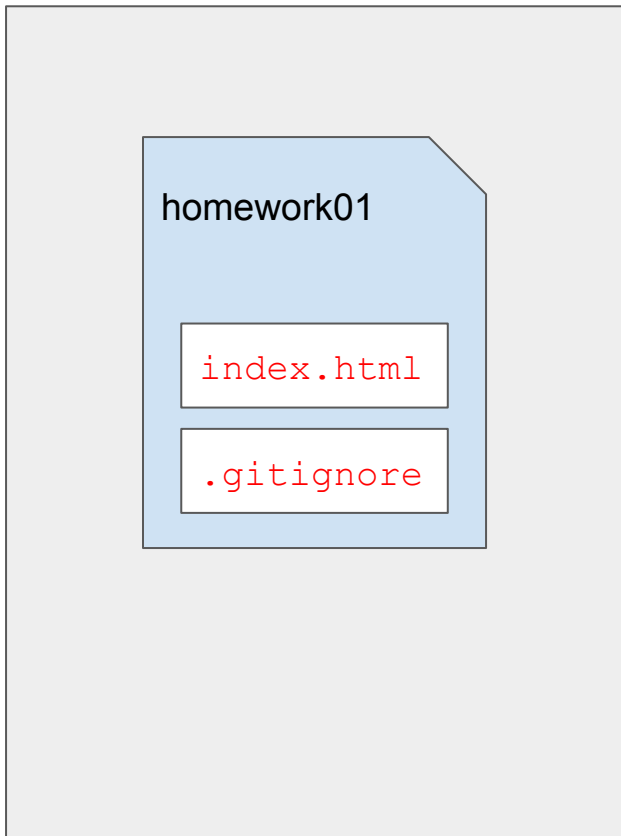
You can init a folder with or without files in it. If you init, with files in the folder, they still need to be staged and committed.

There is no project on github

Github (aka remote)

Starting from Remote: Step 3 - make local changes

My computer (aka "local")



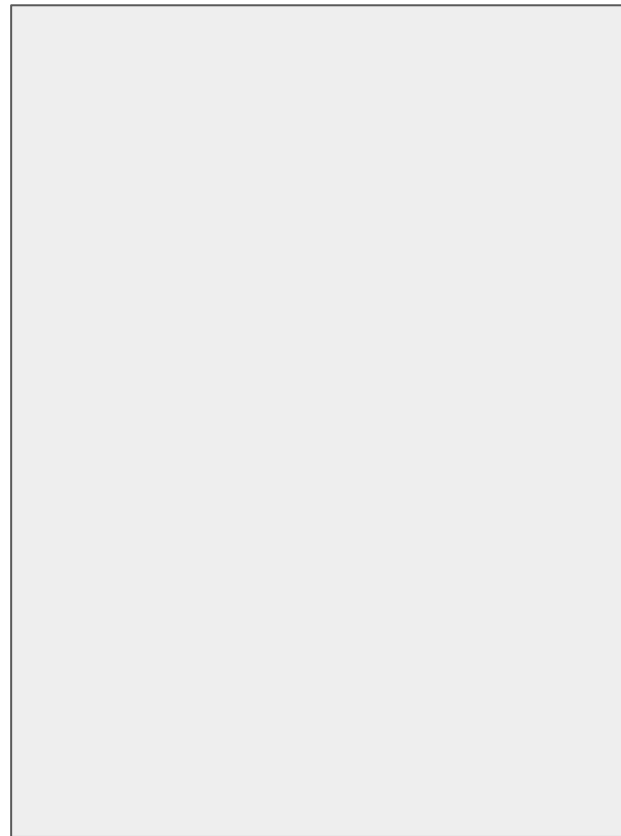
Remember: git is interested in **changes** to the project

- New files
- File edits
- Deleted files

Make sure you create a `.gitignore`. Put this in it:

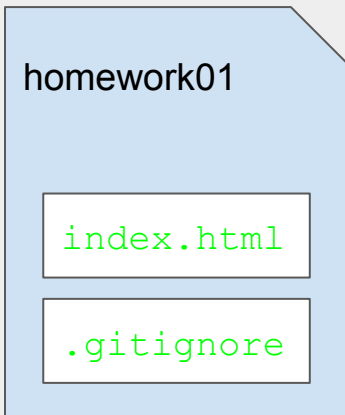
```
.DS_Store  
node_modules  
.vscode
```

Github (aka remote)



Starting from Remote: Step 4 - stage local changes with `git add`

My computer (aka “local”)

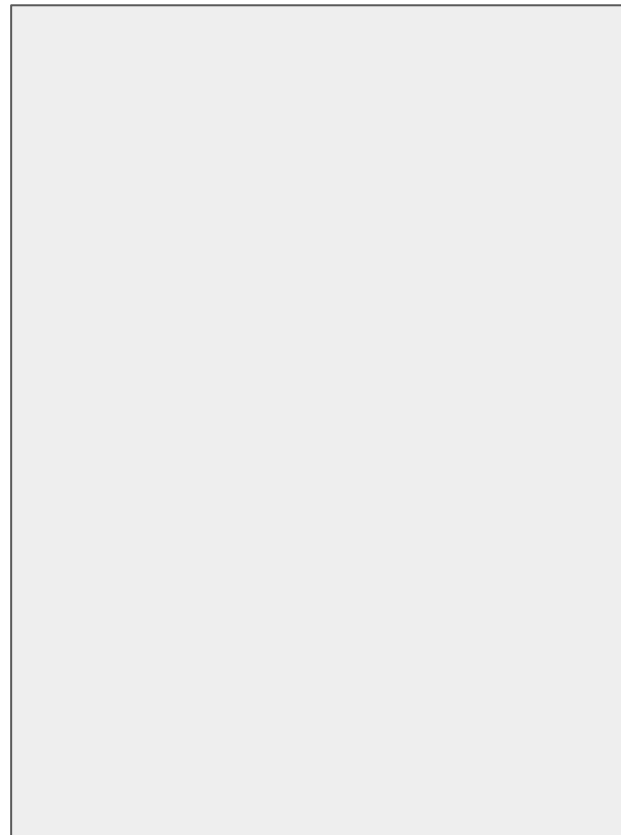


```
git add -A
```

Staging changes is like putting things in a shopping cart: I am choosing the changes I want to keep

`git add -A` adds **all** the available changes, including new files, file edits and deleted files.

Github (aka remote)



Starting from Remote: Step 5 - keep local changes with `git commit`

My computer (aka "local")



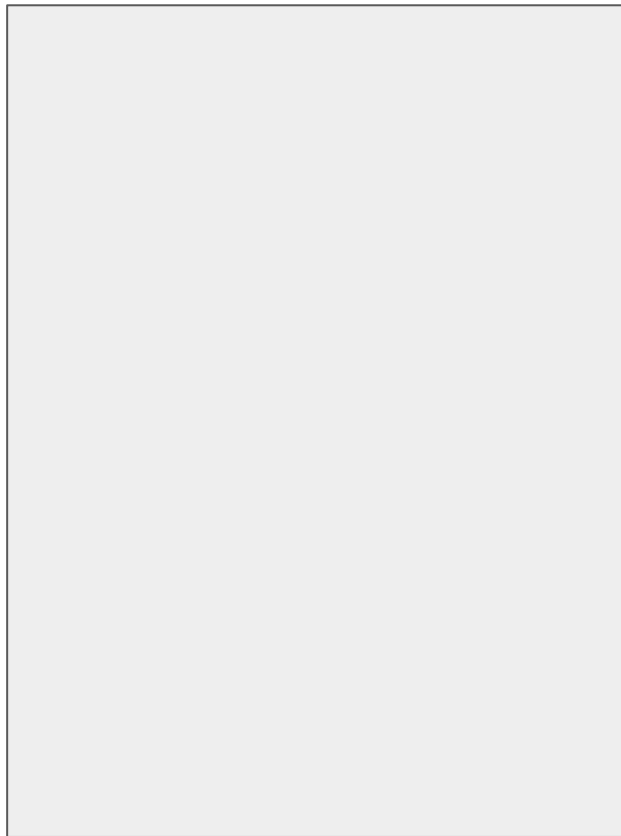
```
git commit -m  
"Initial commit"
```

Committing changes

is like making your purchase. You have decided to keep them changes.

At this point it is still relatively easy to roll back or modify your commit. It hasn't affected your remote (i.e., shared) project.

Github (aka remote)



Starting from Remote: Step 6 - create an empty project on github

My computer (aka "local")

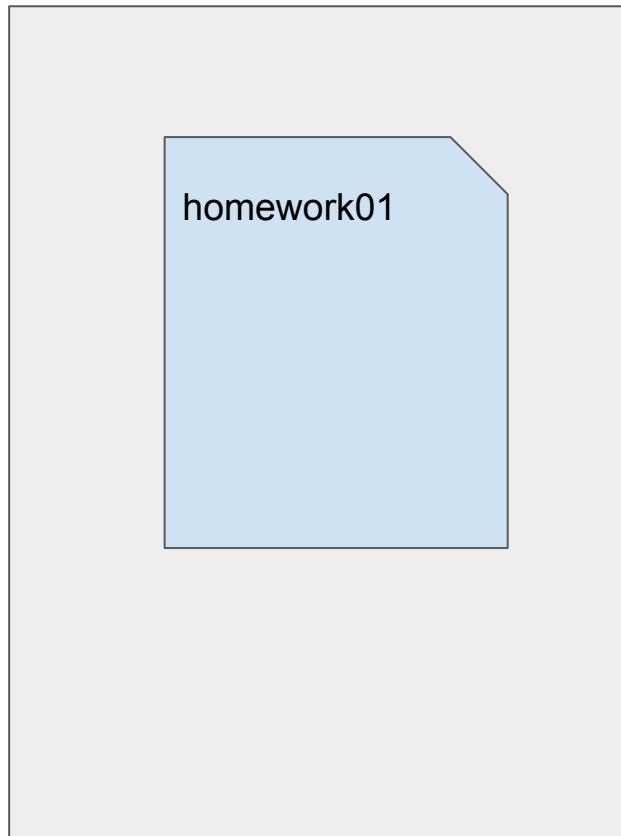


Make sure your remote project has nothing in it. Do not add a README, a .gitignore or a license.

Git projects have to have the same timeline. Since your local project already has some additions, it has the

Our local project is still not connected to github

Github (aka remote)



Starting from Remote: Step 7 - add the github project as a remote

My computer (aka "local")

homework01

index.html

.gitignore

```
git remote add origin  
git@github.com:johnfyoung/  
homework01
```

We need to connect our local to a bare github project. We do this by adding the github project as a **remote**.

Using the clone address from the github repo, add it as a remote called origin

Now the local repo is aware of the github remote

Github (aka remote)

git@github.com:johnfyoung/homework01

homework01

Starting from Remote: Step 8 - publish changes with `git push`

My computer (aka “local”)

Github (aka remote)

