Git Submodules
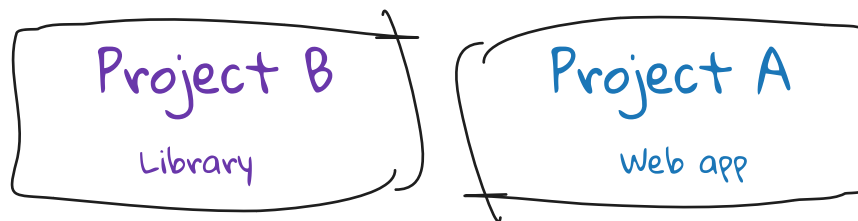
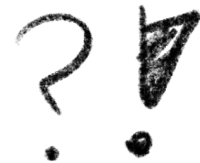## What are submodules and mainly WHY?!

Say that you have 2 different repositories,
a library and a web app project

Project B
Library

Project A
Web app
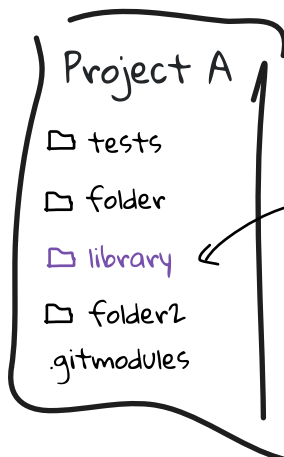
to use Project B as a subdirectory within inside project A

✓ without copying code
✓ without build & deployment
✓ without having the same commit history

?!

and keep them as a separate repositories

✦ we use git submodules ✦

Project A
📁 tests
📁 folder
📁 library
📁 folder2
.gitmodules

a folder but actually referencing a
branch from project B

☺ and yes ! with all the code from project B.

you commit , push , pull separately

Now how do we do that?

☐ Adding a submodule

```
git submodule add (repository url)
```

This will generate **.gitmodules**

the configuration file that tells us what submodules we have. *should be pushed!*

↻ Now that you have a submodule, to push and pull from the submodule you should be inside it's folder.

And That's it!

☐ Cloning & Updating

Whenever you clone the main project, the submodules come empty and so we need to initialize them and pull them

```
git submodule update --init --recursive
```

initialize submodule

recursively (all submodules)

to merge your current submodule branch with remote

```
git submodule update --remote --merge
```

You could use all regular git operations inside the submodule folder.