Class VII – VIII: GIT & GITHUB

GIT STATUS

GIT IGNORE

GIT ADD

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Cuando cree un nuevo proyecto:

1. Git init  
   En github crear un nuevo repositorio.
2. Git add .
3. Git status (para chequear status)
4. Git commit -m “first commit”
5. Git push -u origin master

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Después de *Milestones* en mi proyecto, ponerle nombres adecuados “Responsive”, “withFlex”, “withGrid”, “withBS”, HACEMOS:

1. Git add .
2. Git commit -m “first commit”
3. Git push -u origin master

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Para crear una rama:

1. Git branch juan-rama
2. Git checkout juan-rama

Rama hace sus modificaciones

1. Git add .
2. Git commit -m “CSS”

Hace falta merge:

1. Git merge juan-rama

Ejemplo: desp de crear la barra de modificacion. Como hice una modificacion, y el archivo estaba ya creado 🡪 git commit -m “Navegation superior”

Si creo un nuevo archivo, sí primero hago:

1. “git add .” y luego
2. git commit -m “modificacion hecha” , y si era una nueva rama
3. Pasarse al master
4. GIT MERGE *nombre\_de\_la\_rama*

Git commit -am “” 🡪 se usa para add y commit al mismo tiempo

--

No se recominda trabajar sobre la rama master. Trabajar sobre un rama. Cada vez que se hace una modificacion sobre el archivo, hacerlo en una rama. Cuando todas las modificaciones estén correctas podemos volver al master.

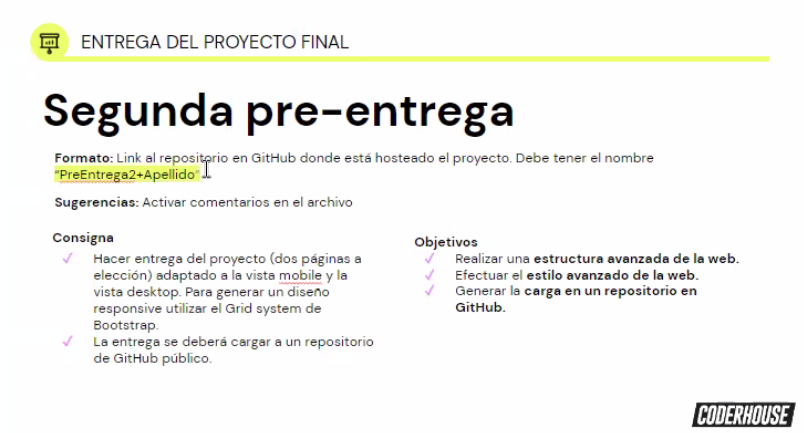
GIT CHECKOUT *nombre\_de\_la\_rama*: Movernos entre ramas

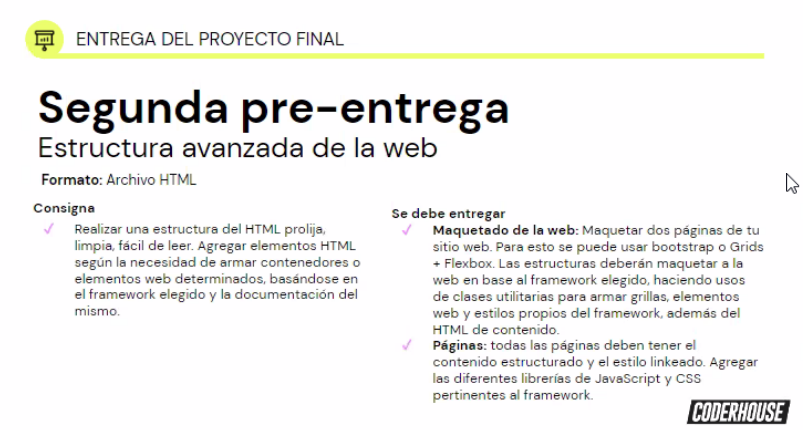
GIT BRANCH: Ver cuáles son las ramas abiertas

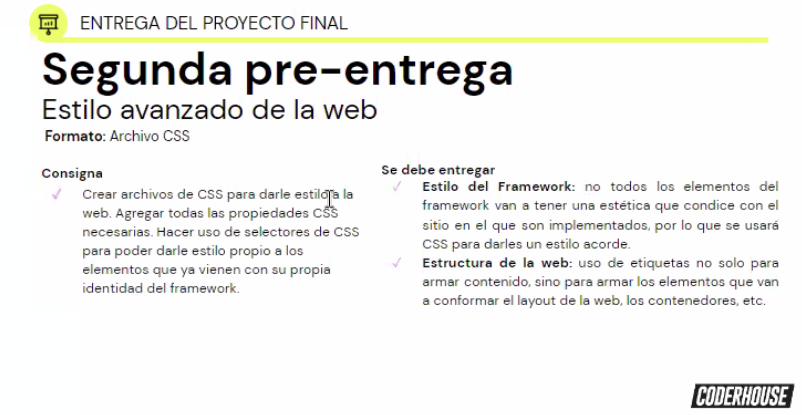
GIT BRANCH -D *nombre\_de\_la\_rama:* Para borrar una rama

Como unir una rama a la main:

GIT MERGE









Únicamente grid de BoosTrap.

Usar media-queries también! No 100% BS para hacer todos los HTML responsives. Es decir se puede hacer una de los 5 html responsive con **media query**

Flex-box con CSS puro

<https://docs.google.com/presentation/d/1f6oyGRfjMpYg6VJ-PEoLOOdO_PmGFvLlrg78ZjwAY1o/edit#slide=id.g1b74bf6a3af_0_301>