

Git-Github Flow (with branches)

Step 1 - Creating a repository:


(the following actions only need to be carried out once at the beginning of the project)

- Create a new GitHub repository with a Readme file in it.
- Create a local copy of your repository → `git clone <address-of-the-remote>`
- NEVER work in MAIN, so create and switch to branch **dev** → `git checkout -b dev`
- NEVER push in MAIN or DEV, so create branch protection rules in github.

Step 2 - Branches and how to use them:

(repeat these actions every time you want to develop a new feature in your project)

- Every time you want to develop a new feature in your project:
 - `git branch`
 - Shows all the local branches in your project
 - Let's you know on which branch you are (asterisk)
- If the branch command shows you ARE in branch **dev**:
 - `git checkout -b <name-of-new-branch>` -> to create a new branch (check the naming conventions for branches)
 - `git checkout <name-of-branch>` -> to switch to a already created branch
(dev should never be worked in directly, so make sure to not commit in it)
- If the branch command shows you are NOT in branch **dev**:
 - Proceed with the commands in **step 3** and **step 4**
 - And switch to a branch already created:
 - `git checkout <name-of-branch>`
 - Or create a new branch and switch to it:
 - `git checkout dev`
 - `git checkout -b <name-of-new-branch>`
-> check the naming conventions for branches



```
~/dev/my-cool-site (main) >> git branch
another-branch
feature-branch-one
feature-branch-three
feature-branch-two
just-one-more
* main
really-old-stale-branch
~/dev/my-cool-site (main) >> _
```

Branches for new features should always be created from dev. For that, make sure to be in the dev branch when creating the new one.

Step 3 - Git status

(repeat this action every time you are stuck and don't know what git command to do)

- It tells you:
 - The branch you are currently on
 - The commits made
 - The changes made that need to be added and committed
 - If your remote is behind your local and you need to push

```
@:~/week-4-game <master>$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   assets/css/style.css
        modified:   index.html

Unmerged paths:
  (use "git reset HEAD <file>..." to unstage)
  (use "git add <file>..." to mark resolution)

        both modified: assets/javascript/game.js

@:~/week-4-game <master>$
```

Step 4 - Saving your progress

(repeat these actions every time you want to save a new piece of progress locally)

- If the status command shows red files run one of the following:
 - `git add <name-of-file> or <path-of-file>`
 - `git add .`
- If the status command shows green files:
 - `git commit -m "Message of the commit"`
- If the status command shows "nothing to commit":
 - Proceed with **step 5**.

Step 5 - PUSHing your progress

(repeat these actions every time you want to share a local branch and its progress)

- To push local branch and create an upstream connection with its remote counterpart:
 - `git push -u origin/<name-of-local-branch>`
 - Every time afterwards that you want to push into that branch:
 - `git push`
- If you don't want to create an upstream connection, you can do everytime:
 - `git push origin/<name-of-branch>`

Step 6 - PULLing dev

(repeat these actions whenever the dev branch in the remote is ahead of local dev, and if you don't know if it is, do it before creating a new branch from dev or before a PR*)

- If the branch command shows you are NOT in dev:
 - `git pull origin/dev`
- If you ARE in dev
 - `git pull`
- Solve the merge conflict if necessary and to finalise it:
 - `git commit -m "Message of the commit"`
- Repeat in every local branch

Step 6 - Pull Request (PR) and merging with branch dev

(repeat these actions every time you finish developing a feature and want to merge the branch its been developed into dev)

- Proceed with **step 6** and **step 5** in that order:
- Go to GitHub and create a Pull Request
- Ask your collaborators to approve the PR if required in the branch protection rule.
- Merge once it's possible.
- Delete the branch you has just merged both locally and in the remote.