Git 301

Warmup

Take a moment to create a GitHub account, or verify your password. Feel free to post your username on your Slack profile. Follow me @hcientist.

Roadmap

In this session, you'll learn how to fork a repository, make changes locally, open a pull request, and synchronize the project back to your computer.

Step 1 - Fork the repo

Go to https://github.com/hcientist/Stylin-Syrup-Sow/ and click `Fork` in the upper right-hand corner. If prompted, choose your personal account.

Step 2 - Clone your new repo

Clone your new personal fork to your computer. If you like, use the green `Code` button to get the repo URL with SSH syntax.

```
$ git clone git@github.com:YOURNAME/Stylin-Syrup-Sow.git
Cloning into 'Stylin-Syrup-Sow'...
remote: Enumerating objects: 265, done.
remote: Counting objects: 100% (25/25), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 265 (delta 6), reused 14 (delta 2), pack-reused 240
Receiving objects: 100% (265/265), 50.55 KiB | 3.61 MiB/s, done.
Resolving deltas: 100% (77/77), done.
```

Step 3 - Confirm your origin

Confirm the origin URL is set correctly. (From here out, you should see your e-ID instead of `hcientist`.)

```
~/Stylin-Syrup-Sow$ git remote get-url origin
git@github.com:hcientist/Stylin-Syrup-Sow.git
```

Step 4 - Add an upstream remote

Add my repo as an additional remote. You can use the HTTPS URL since you're not writing to this repository.

```
$ git remote add upstream https://github.com/hcientist/Stylin-Syrup-Sow.git
$ git remote
origin
upstream
```

Step 5 - Create a new branch

Create a new branch by "checking it out" with a name that roughly describes its purpose, like `add-YOURNAME`.

```
$ git checkout -b add-mike
Switched to a new branch 'add-mike'
$ git branch
* add-mike
  main
```

Step 6 - Add some style and commit

1. Open the `index.html` file in your favorite editor, and add a style rule that you think improves the page.
2. ensure the page still passes HTML validation
3. ensure the page still has no accessibility issues
4. Once you've made your change and ensured compliance with validation and a11y standards, add and commit the file like we've done before.

```
$ git add index.html
$ git commit
[add-hcientist 0823ef7] Add Mike
 1 file changed, 1 insertion(+)
```

Step 7 - Push your changes to GitHub

Use `git push` to copy your changes to GitHub, but make sure you specify the correct branch name.

```
$ git push origin add-mike
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 347 bytes | 115.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'add-mike' on GitHub by visiting:
remote:      https://github.com/hcientist/Stylin-Syrup-Sow/pull/new/add-mike
remote:
To github.com:hcientist/Stylin-Syrup-Sow.git
 * [new branch]      add-mike -> add-mike
```

Step 8 - Open a pull request

Use either the link shown in your terminal, a notification that probably popped up on GitHub, or just the `New pull request` button on the `Pull requests` tab to start a new pull request. It should show that you are trying to send changes from your new branch to the `main` branch on the UUG repo.

Step 9 - Wait

Wait here for your pull request to be merged.

Step 10 - Switch back to main

Switch back to the main branch by checking it out.

```
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

Step 11 - Pull your merged changes

```
$ git pull upstream main
hint: Pulling without specifying how to reconcile divergent branches is
hint: discouraged. You can squelch this message by running one of the following
hint: commands sometime before your next pull:
hint:
hint:   git config pull.rebase false  # merge (the default strategy)
hint:   git config pull.rebase true   # rebase
hint:   git config pull.ff only       # fast-forward only
hint:
hint: You can replace "git config" with "git config --global" to set a default
hint: preference for all repositories. You can also pass --rebase, --no-rebase,
hint: or --ff-only on the command line to override the configured default per
hint: invocation.
From https://github.com/hcientist/Stylin-Syrup-Sow
 * branch            main       -> FETCH_HEAD
 * [new branch]      main       -> upstream/main
Already up to date.
```

Step 12 - Check the logs and update your GitHub repo

Run `git log` and look for your change.

Then `git push origin main`.

Step 13a - Remove your development branch locally

Once your change has been merged and pulled back into `main`, you can "safely" delete your development branch with `git branch -d add-yourname`. If git doesn't recognize that your change has been merged because it was modified along the way, you can force delete with `git branch -D add-yourname`.

Step 13b - Remove your development branch remotely

To remove a branch you pushed to GitHub, you can prefix it with a colon. For example:

```
git push origin :add-yourname
```