

Two-Step Branch Creation and Push

Here are the commands to create your branch, **feature_fogninis_family**, and push it to your remote repository (GitHub).

Step 1: Create and Switch to the New Local Branch

This command checks out the **main** branch, pulls the latest changes to ensure you branch from the most current code, and then immediately creates and switches to your new feature branch.

Bash

```
# 1. Ensure your local main is up-to-date with the remote
git checkout main
git pull origin main

# 2. Create the new branch named feature_fogninis_family
and switch to it
git checkout -b feature_fogninis_family
  • git checkout -b <new-branch-name> is a shorthand for creating a new
    branch and checking it out in one go.
```

Step 2: Push the Local Branch and Set Up Remote Tracking

After you've done some work and committed your changes on the new branch, you use this command for the **very first push**.

Bash

```
# After making and committing your changes (git add . and
git commit -m "Initial feature work")
git push -u origin feature_fogninis_family
  • git push origin <branch-name>: This publishes your local branch to the
    remote repository (origin).
```

- The **-u** flag (or **--set-upstream**) is the key part:
 - It tells Git to set up a **tracking relationship** between your local `feature_fogninis_family` and the remote branch of the same name on `origin`.
 - After this initial push, all future pushes and pulls on this branch can simply be `git push` or `git pull` without specifying the remote name or branch name.



Subsequent Activity

Once the tracking is set up, your day-to-day workflow on this branch will be much simpler:

Activity	Command
Check status	<code>git status</code>
Commit changes	<code>git add .</code> then <code>git commit -m "A meaningful update"</code>
Push updates	<code>git push</code>
Pull latest	<code>git pull</code>

Export to Sheets