This is my workflow.

There are many similar

but this one is mine.

This workflow is by a non-techie person with no knowledge of git or python or even cmd. It has grown organically as I have found my way through the tangle of technology. It's probably not the most efficient, but it minimises the amount of git commands you have to learn.

You will need to create an account on gitlab. EXPAND

Fork the project EXPAND

This gives you your own copy of the project. However, now you have it, it will not be updated automatically. It's like a snapshot of the project at that moment in time.

- Project repo – this is the main repository of information. The changes and additions that you make will be sent to the project repo and, once daily, the website is updated from the project repo.
- Your remote repo. This is the snapshot of the project repo. It's remote because it is kept online on your account. The URL will be www.gitlab.com/YOURUSERNAME/opl-data
- Your local repo. This is a copy of your remote repo that you have downloaded and is stored on your hard drive. Again, it will not update unless you tell it to.

Download the github desktop client. To be honest, any git client would be fine but these instructions specifically refer to that piece of software. EXPAND

Clone your repo. Using the Github client, File > Clone Repository. There are three tabs on the dialogue box. Choose "URL". Add the URL for your remote repo. This will be https://gitlab.com/YOURUSERNAME/opl-data.git

This establishes a connection between your local repo and your remote repo.

Download Gitbash. Any similar software is fine but this guide refers specifically to Gitbash. EXPAND

Branches

The default branch is "main". However, you don't want to work on main. This is more like a reference branch that should match the project repo main branch. If it diverges (because you have accidentally committed to main), then things get sticky. Try not to.

An example workflow to make any change:

- Update your repos from the project. People add to the project all the time, around the clock so you will need to update before you start any piece of work. For this you will use Gitbash or similar command line app. Navigate to the repo (suggest cd c:/opl-data but this depends on where you downloaded your local repo to)

  EXPAND ON HOW TO NAVIGATE

- Now pull the updates from the project to your local.
- `git pull https://gitlab.com/openpowerlifting/opl-data.git main`
- If there were updates then we need to pass these along to your remote.

- Click on the Github client to wake it up and you should see that it is offering to push updates to origin SCREENSHOT REQD
- Now that your repos match the project repo, create a new branch (Branch > New branch). Call it something descriptive.
- You can now work locally, making whatever changes are required.

SECTION NEEDED ON RUNNING CHECKERS

- Once you are happy, commit your changes to your branch SCREENSHOT REQD. Describe the changes you are making in the commit name.
- You can now “Publish branch” SCREENSHOT REQD
- This updates your remote and offers you the option to create a Merge Request SCREENSHOT
- EXPAND ON WHICH OPTIONS TO CHECK

A merge request contains all the changes you want to make and presents it to both the checker scripts and to a human reviewer. If the checker passes and the human is happy with what you are trying to do, your changes will be merged into the project repo. However, they will not show on the website until they are deployed. Typically, Sean does this once a day.

Once the branch has been merged, Branch > delete branch and you will be back on main. However, the main branch on your repo still doesn’t contain the changes you just made. So you need to start the process again with `git pull https://gitlab.com/openpowerlifting/opl-data.git main`