

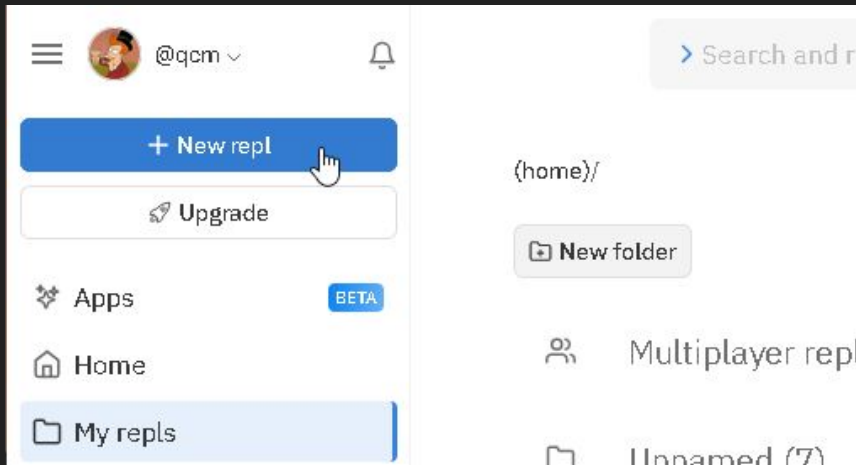
GIT + Replit - local code editing

Tools needed:

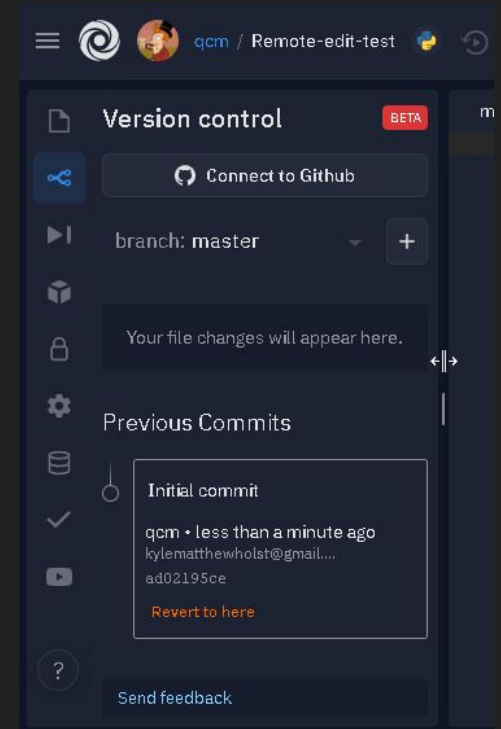
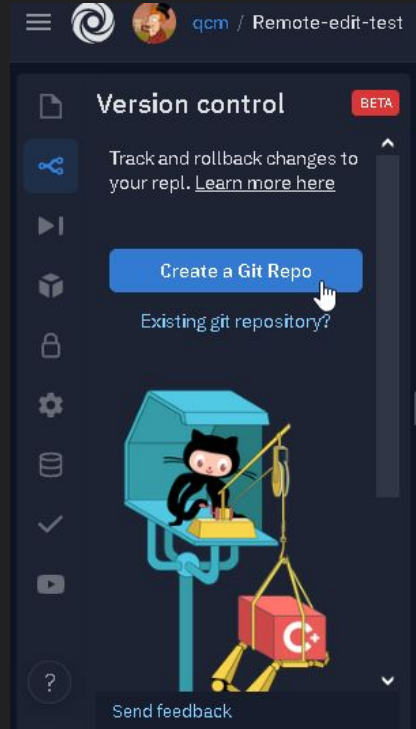
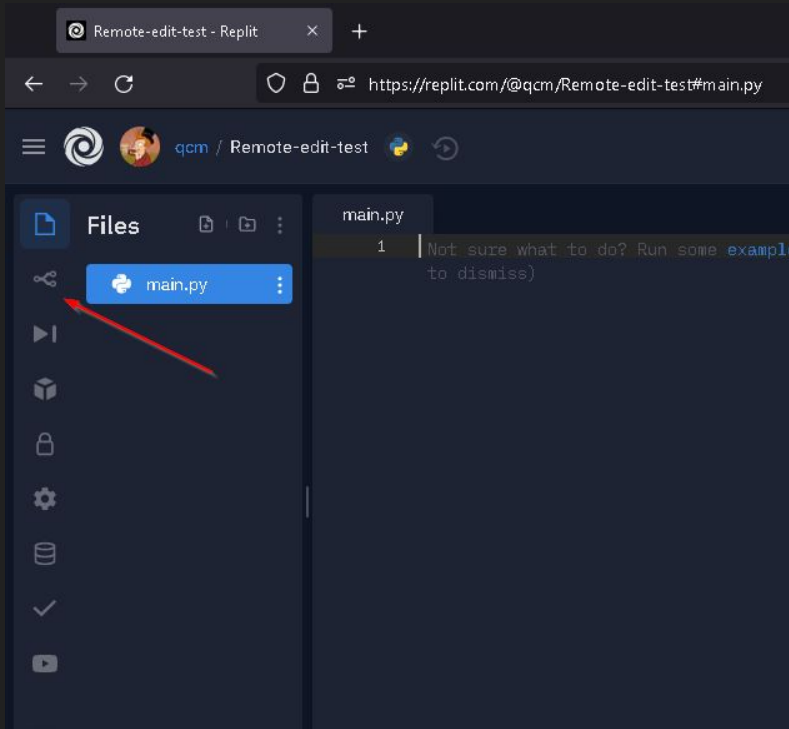
- Computer OS (windows, linux, mac)
- Text editor, IDE
- Github client

Step 1 - make a new repl

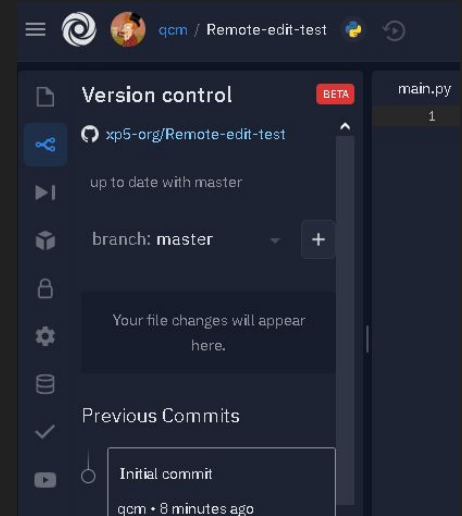
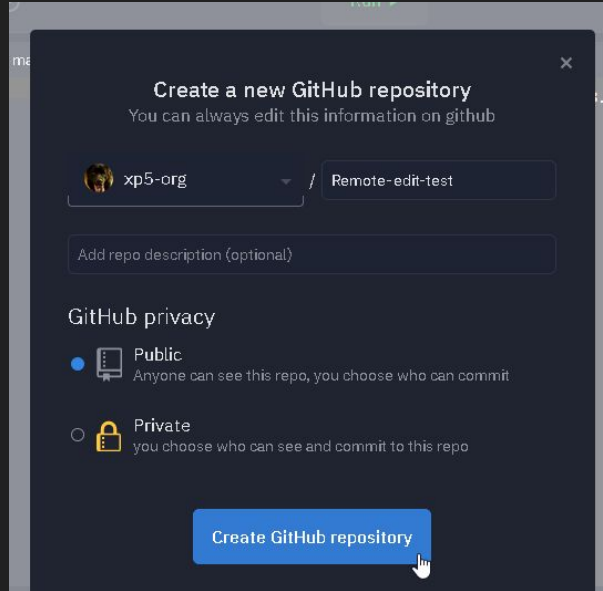
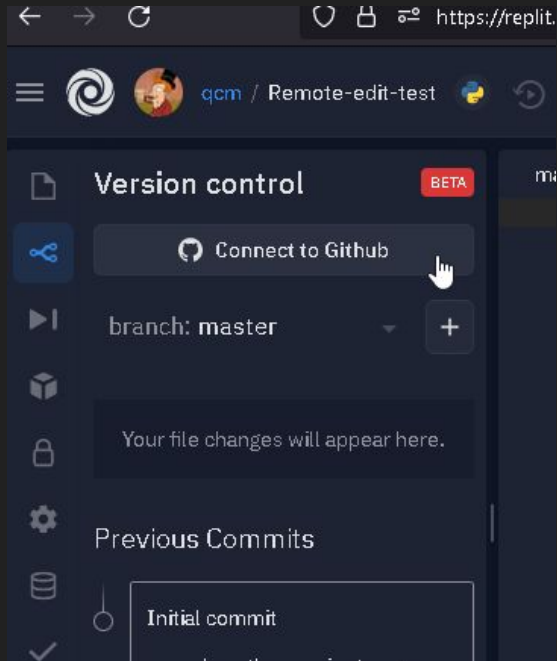
step



Create new repl project



Step 3) Click connect to github, create repo



Step 4) open new repo on github and get the url

The screenshot displays the GitHub web interface. On the left sidebar, under 'Repositories', there is a search bar with 'remote' entered and a 'New' button. Below it, the repository 'xp5-org/Remote-edit-test' is listed. The main content area shows the repository page for 'xp5-org/Remote-edit-test'. At the top, there are tabs for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The 'Code' tab is active. Below the repository name, there are buttons for 'Go to file', 'Add file', and 'Code'. A red arrow points to the 'Code' button. A dropdown menu is open from the 'Code' button, showing options: 'Clone' (with a red arrow pointing to it), 'HTTPS' (with the URL 'https://github.com/xp5-org/Remote-edit-test' highlighted and a red arrow pointing to the copy icon), 'SSH', 'GitHub CLI', 'Open with GitHub Desktop', and 'Download ZIP'.

Search or jump to... / Pull requests Issues Marketplace Explore

xp5-org / Remote-edit-test

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags

xp5-org Initial commit

main.py Initial commit

Help people interested in this repository understand your project by adding a README file.

Go to file Add file Code

Clone

HTTPS SSH GitHub CLI

<https://github.com/xp5-org/Remote-edit-test>

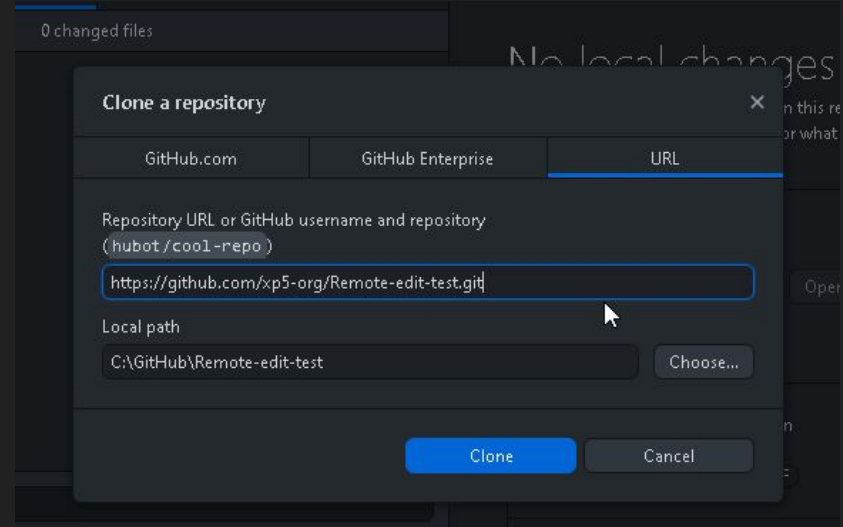
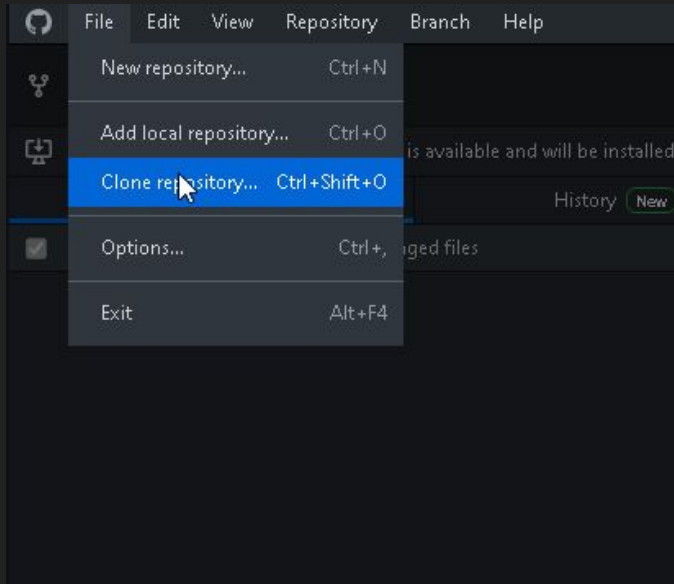
Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

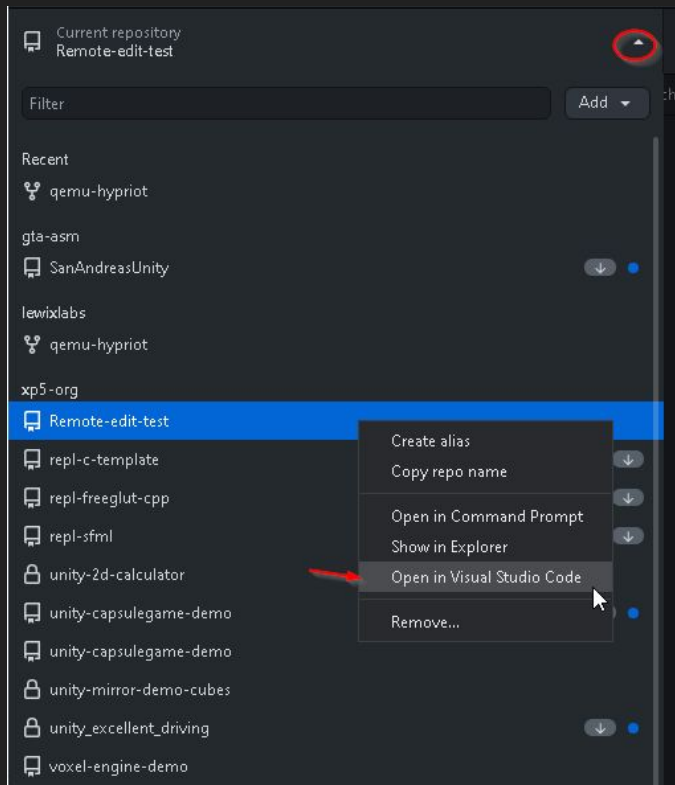
Download ZIP

Step 5) copy url to github desktop (or use cli)

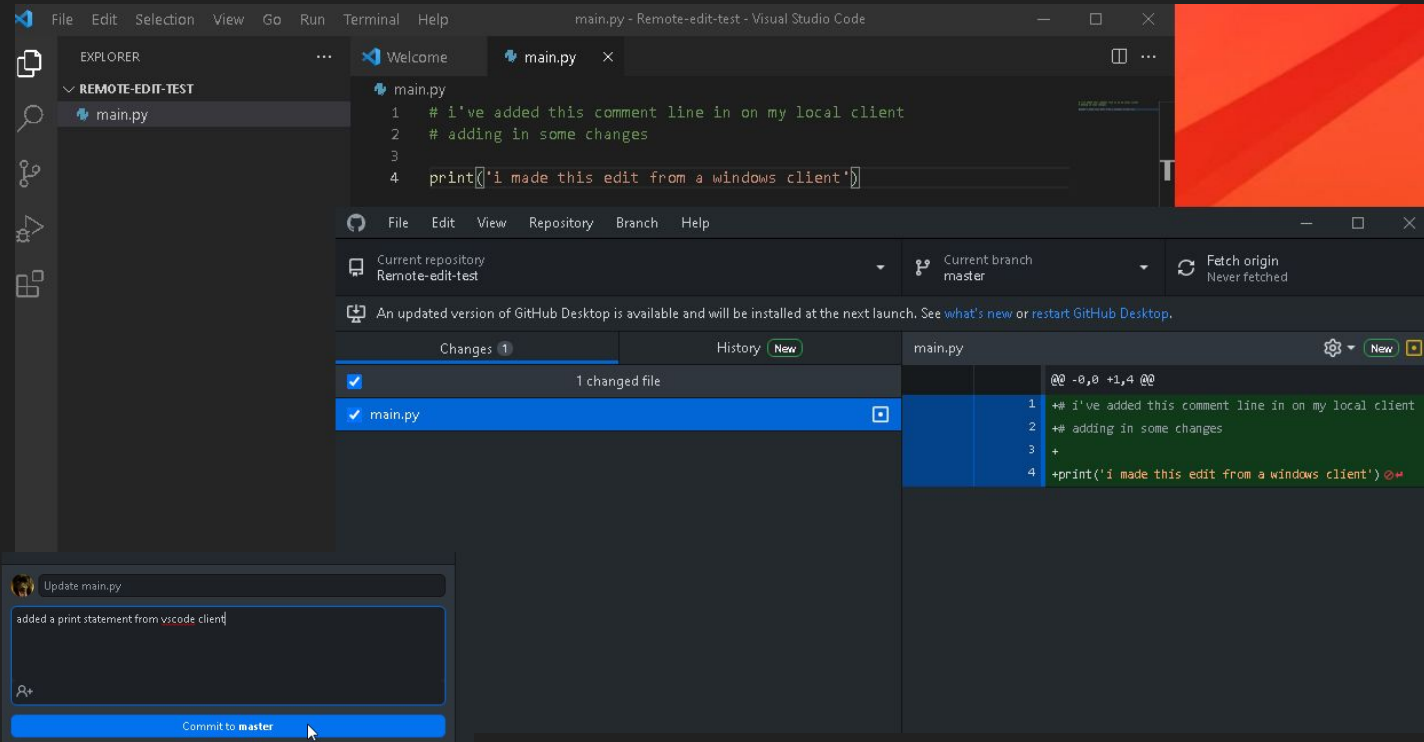
Copy URL to top field, the 2nd box “local path” fills in automatically



Step 6) clone finishes, open in visual studio



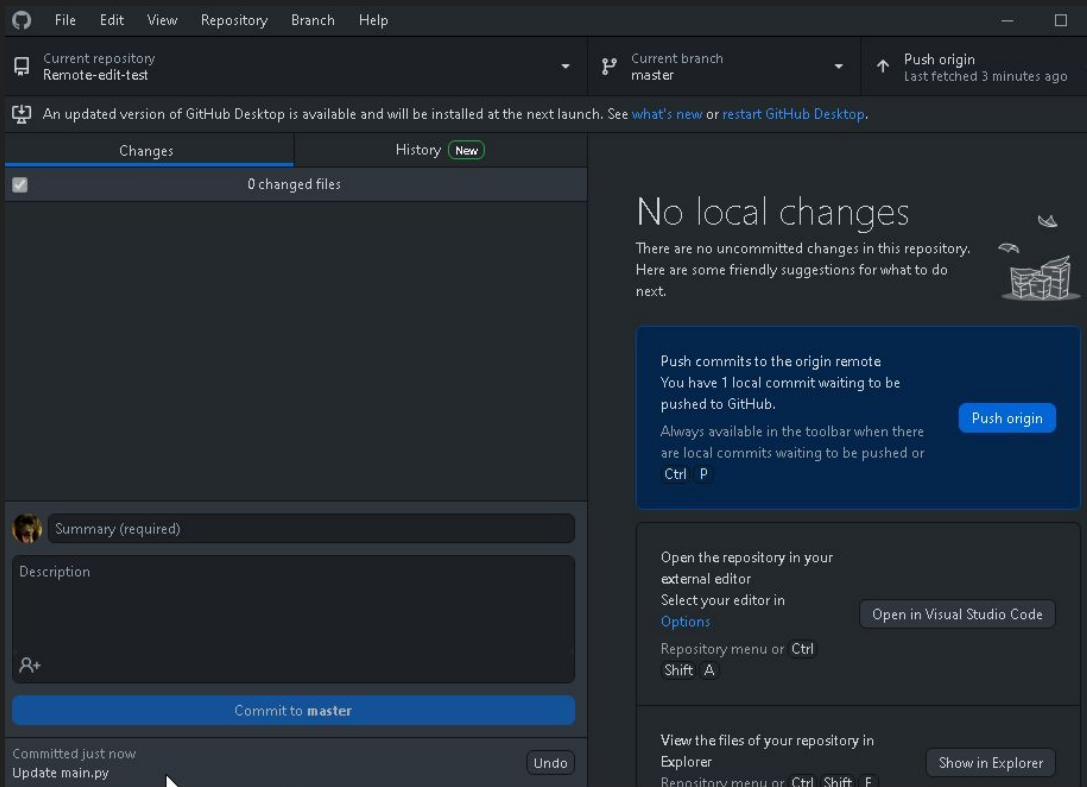
Step 7) make changes on local client, commit to github



Committing to master
“saves” these changes
in the versioning
system, but does not
send the changes to
github

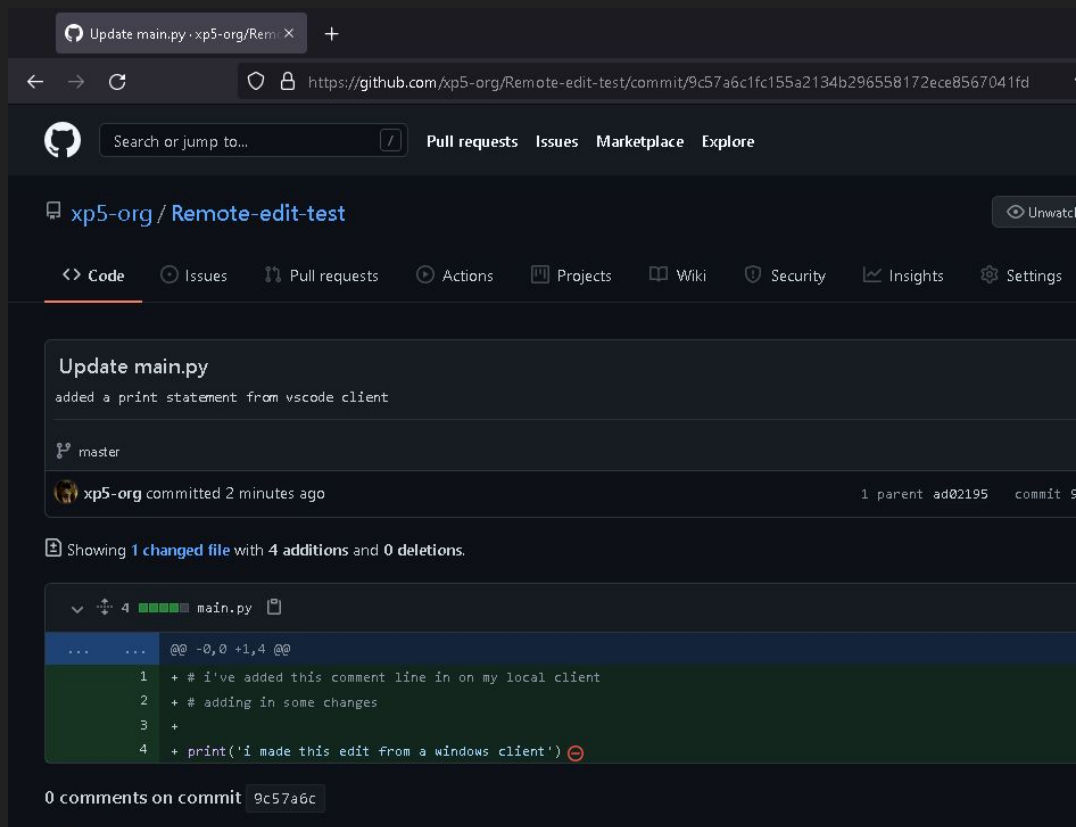
Commit to master =
save locally

Step 8) changes committed, view options here



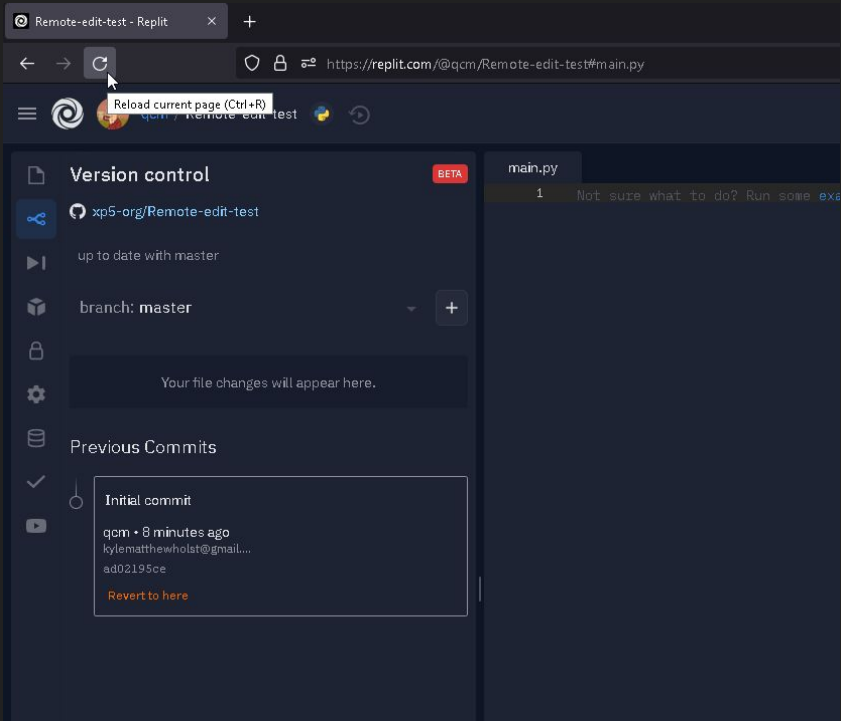
Can push to origin = send changes to Github master branch for everyone to see

Step 9) changes now in main code online at github

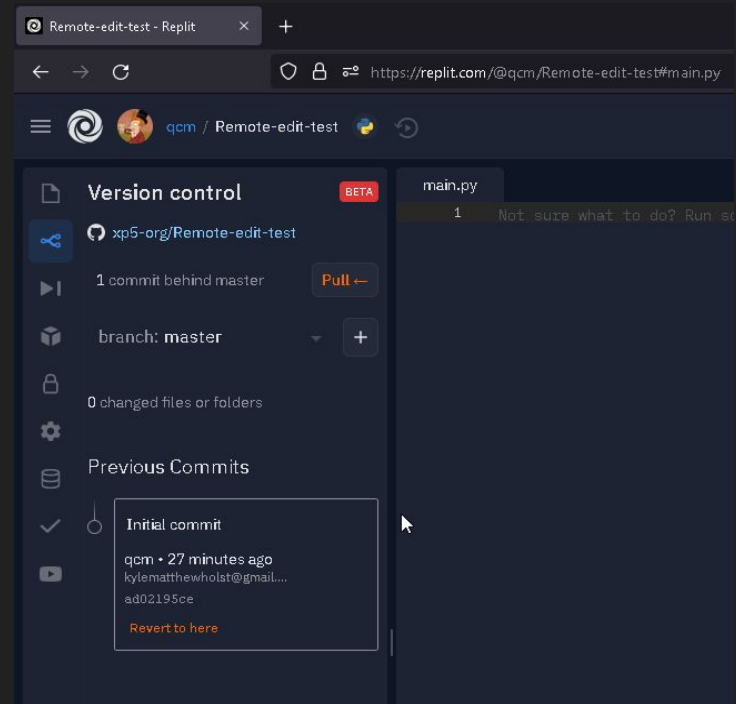


Step 10) check repl

Looks like it doesnt see it , refreshing

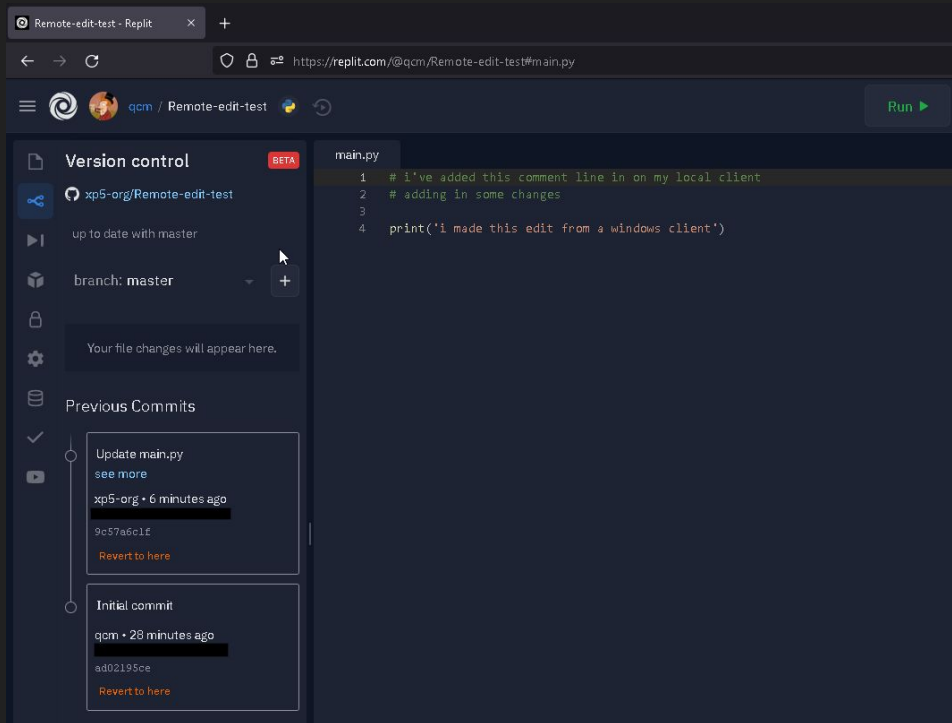


Much better, repl sees we are 1 behind.
Click pull to get the new code



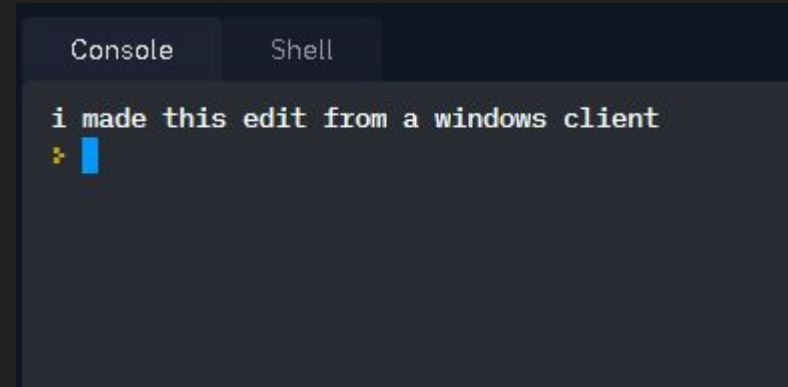
Step 11) check repl and run code

Changes pulled in from github



The screenshot shows the Replit web interface for a project named 'Remote-edit-test'. The browser address bar displays 'https://replit.com/@qcm/Remote-edit-test#main.py'. The interface includes a sidebar on the left with a 'Version control' section (marked BETA) showing the repository 'xp5-org/Remote-edit-test' on the 'master' branch. It indicates the user is 'up to date with master'. Below this, 'Previous Commits' are listed: an 'Update main.py' commit by 'xp5-org' 6 minutes ago, and an 'Initial commit' by 'qcm' 28 minutes ago. The main area is the code editor for 'main.py', containing the following Python code:

```
1 # i've added this comment line in on my local client
2 # adding in some changes
3
4 print('i made this edit from a windows client')
```



The screenshot shows the 'Console' tab of the Replit interface. It displays the output of the code executed in the editor: 'i made this edit from a windows client'. A yellow cursor is visible at the end of the output line.