

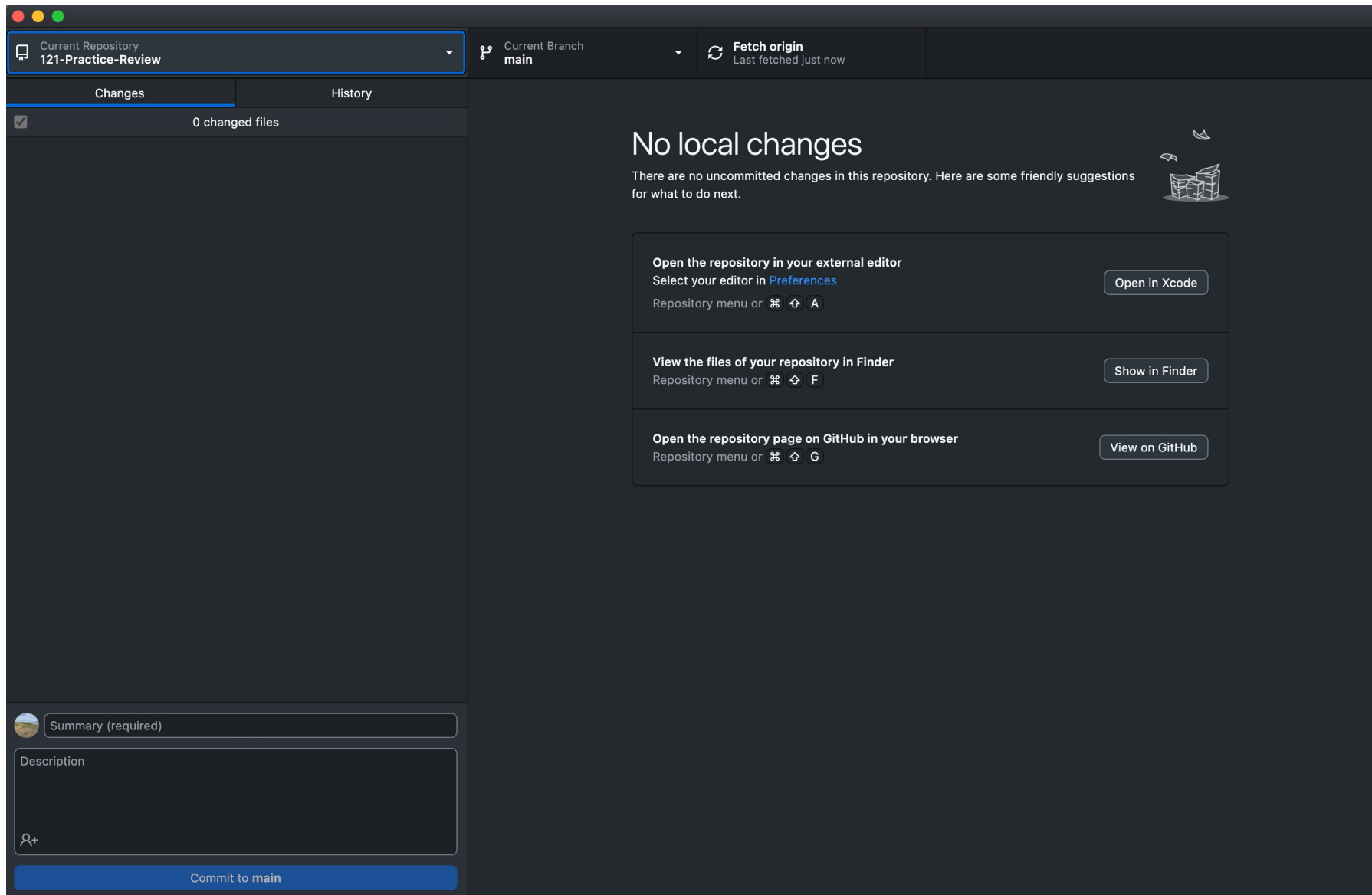
# Using Git

Focus on GitHub Desktop

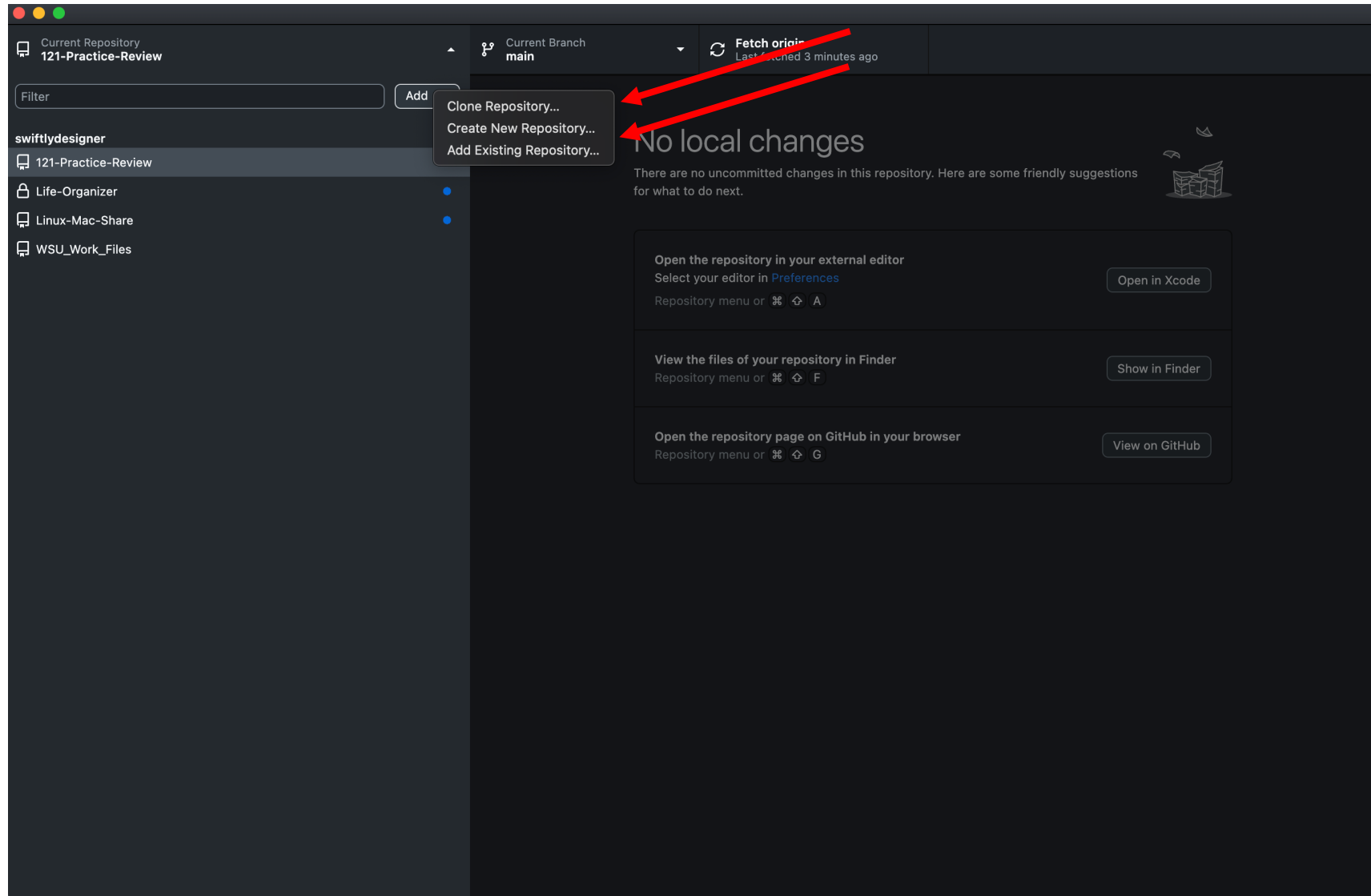
# GitHub Desktop Setup (Others are similar)

- You will be prompted.
- Login with GitHub; follow on-screen prompts

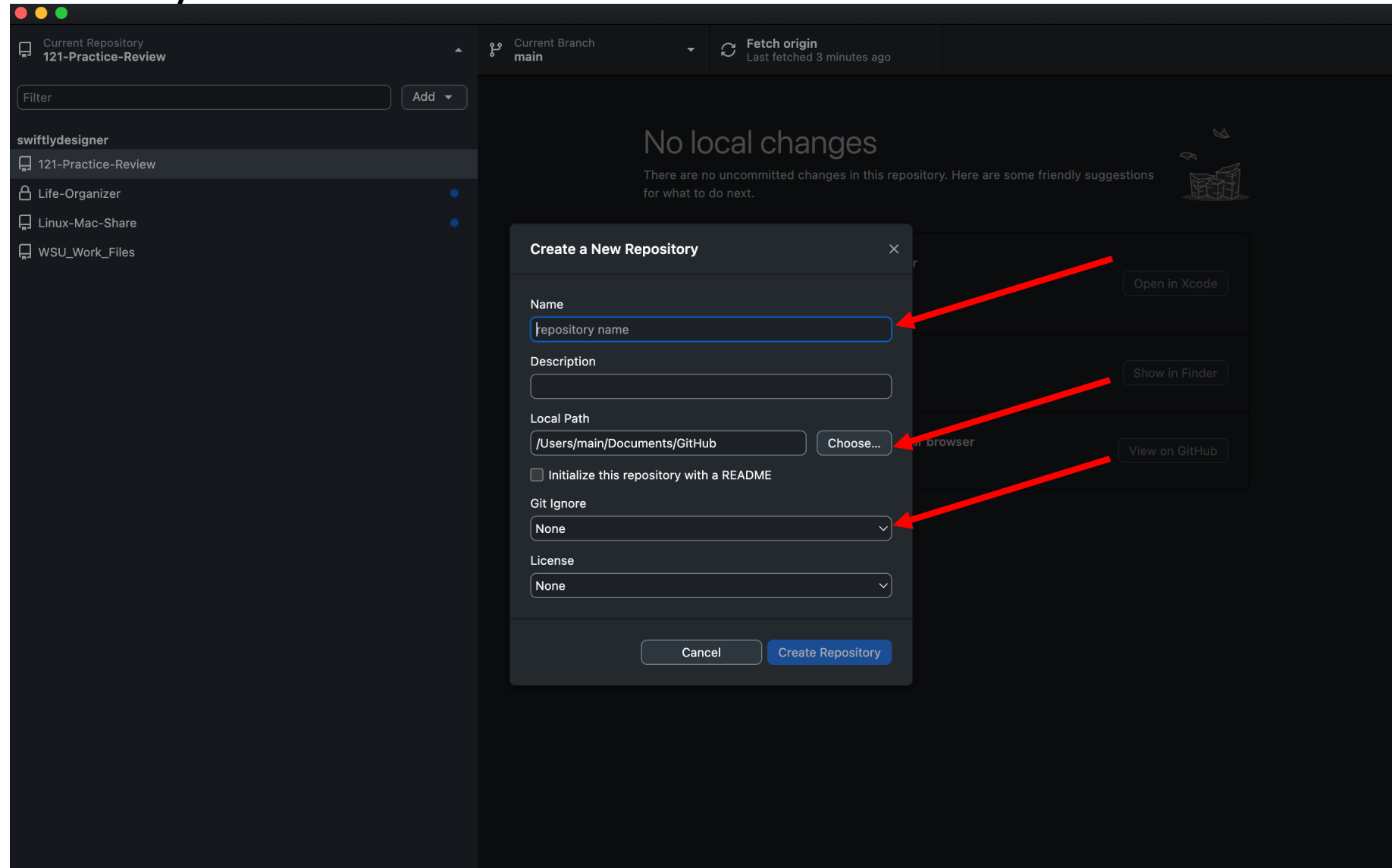
If you have Repos, it will automatically select one.



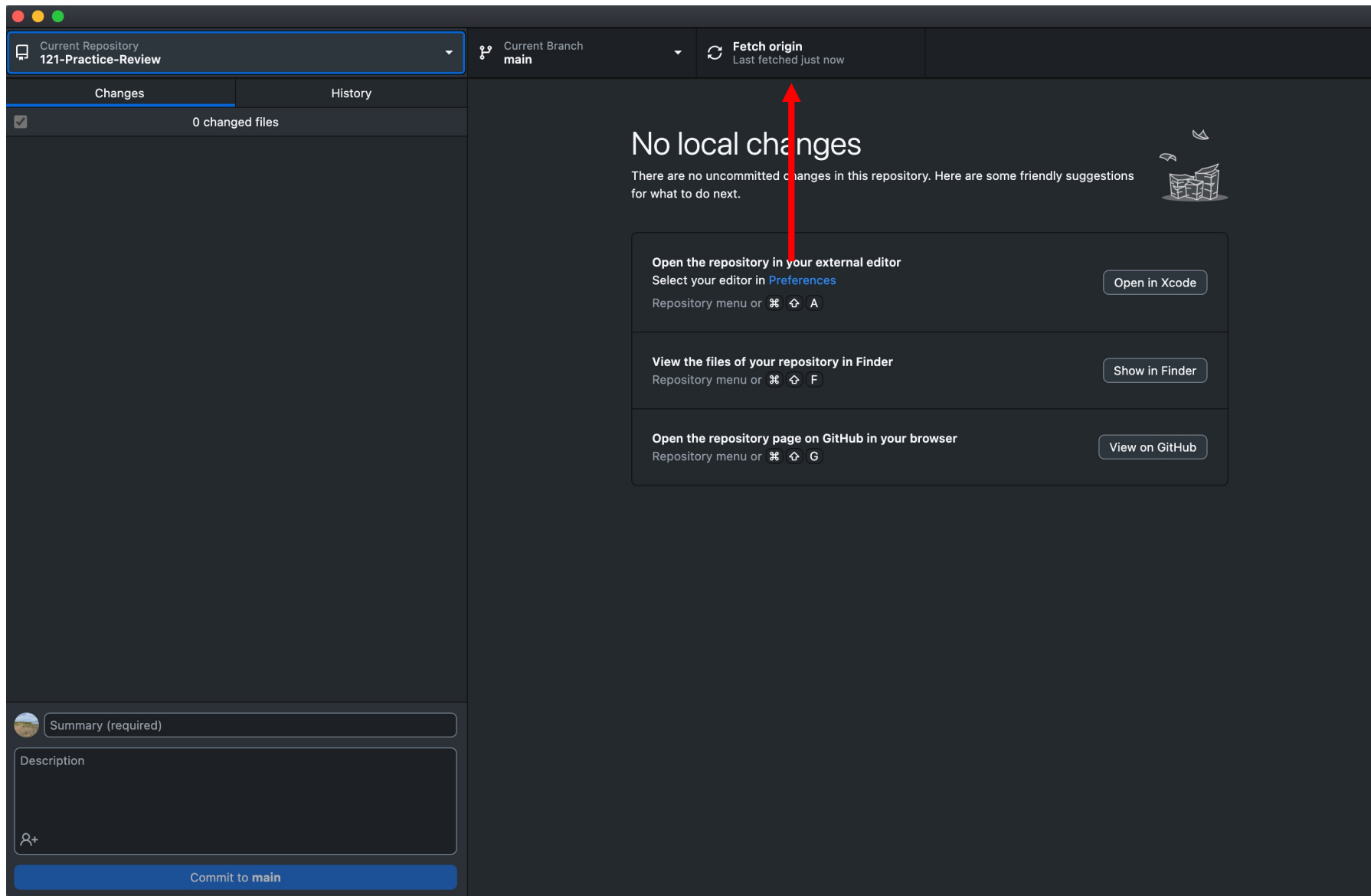
Click Add dropdown menu. Select Clone if you created one already; create if you have not.



Set Git Ignore to VisualStudio. Choose a name. If you cannot think of something, use <last\_name>121 (Parker121).

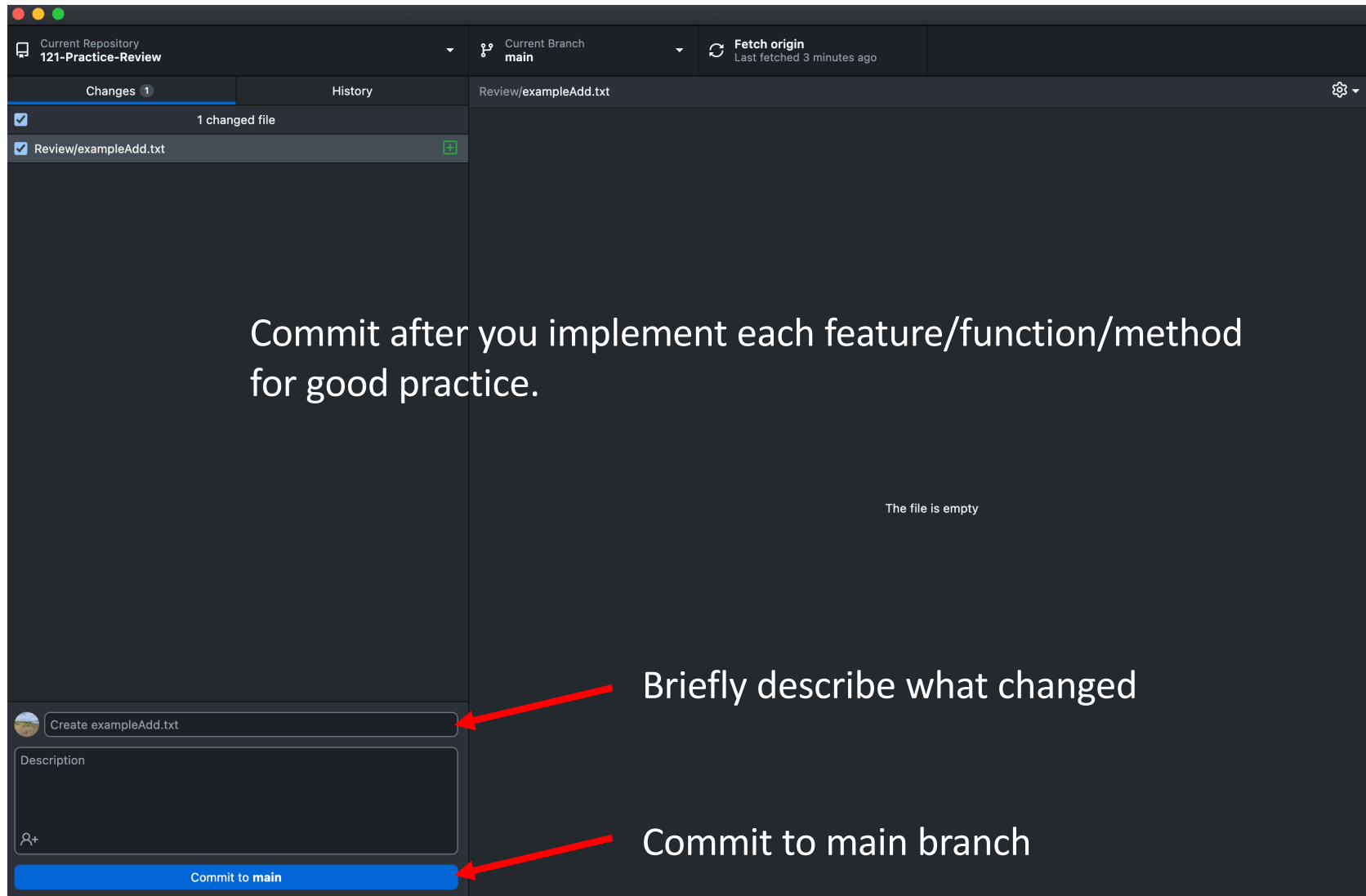


# Fetch: Get an update from remote server

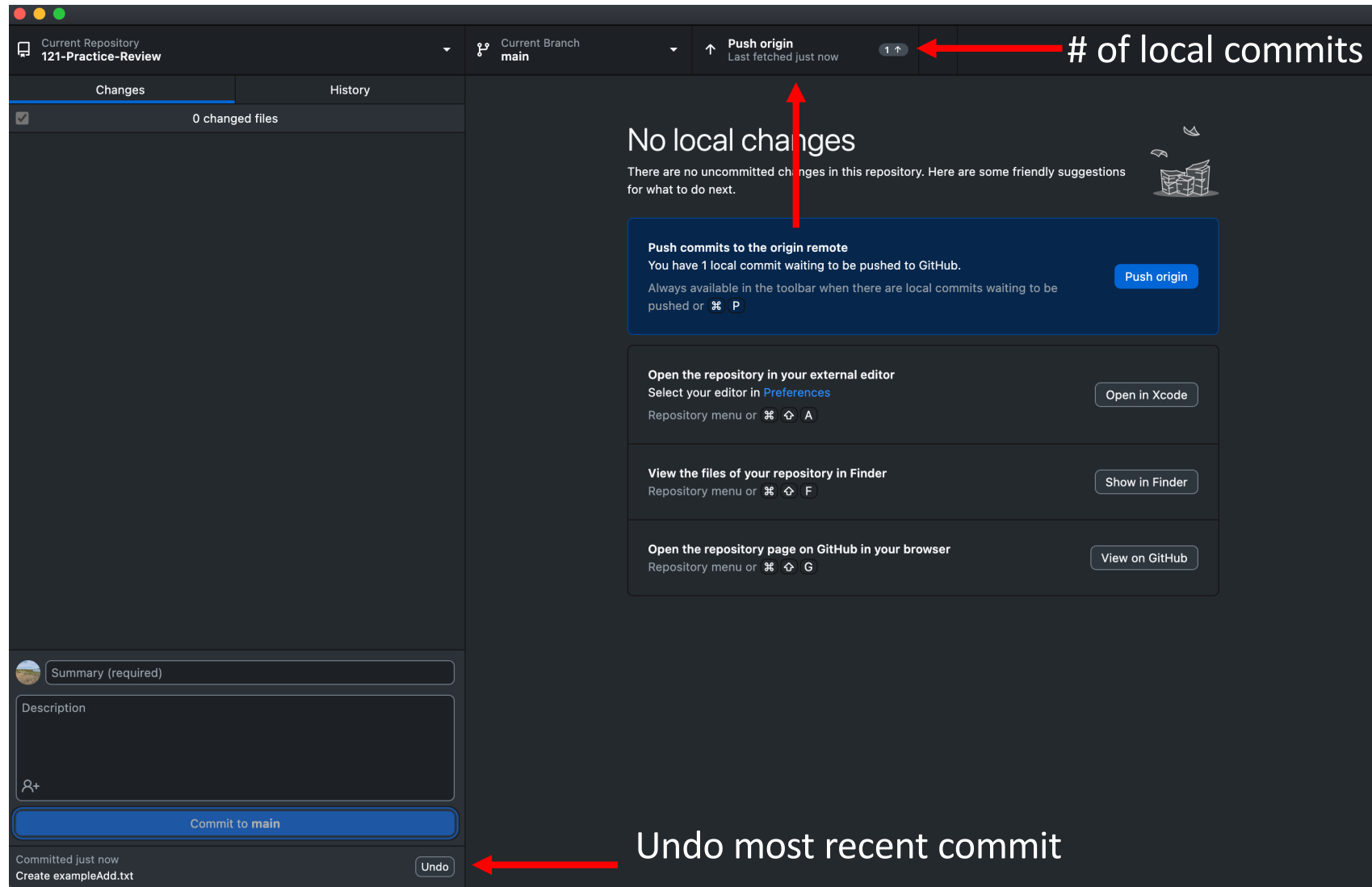


# Commit: Save a local change to the database

Note: This does not push it to the server yet.

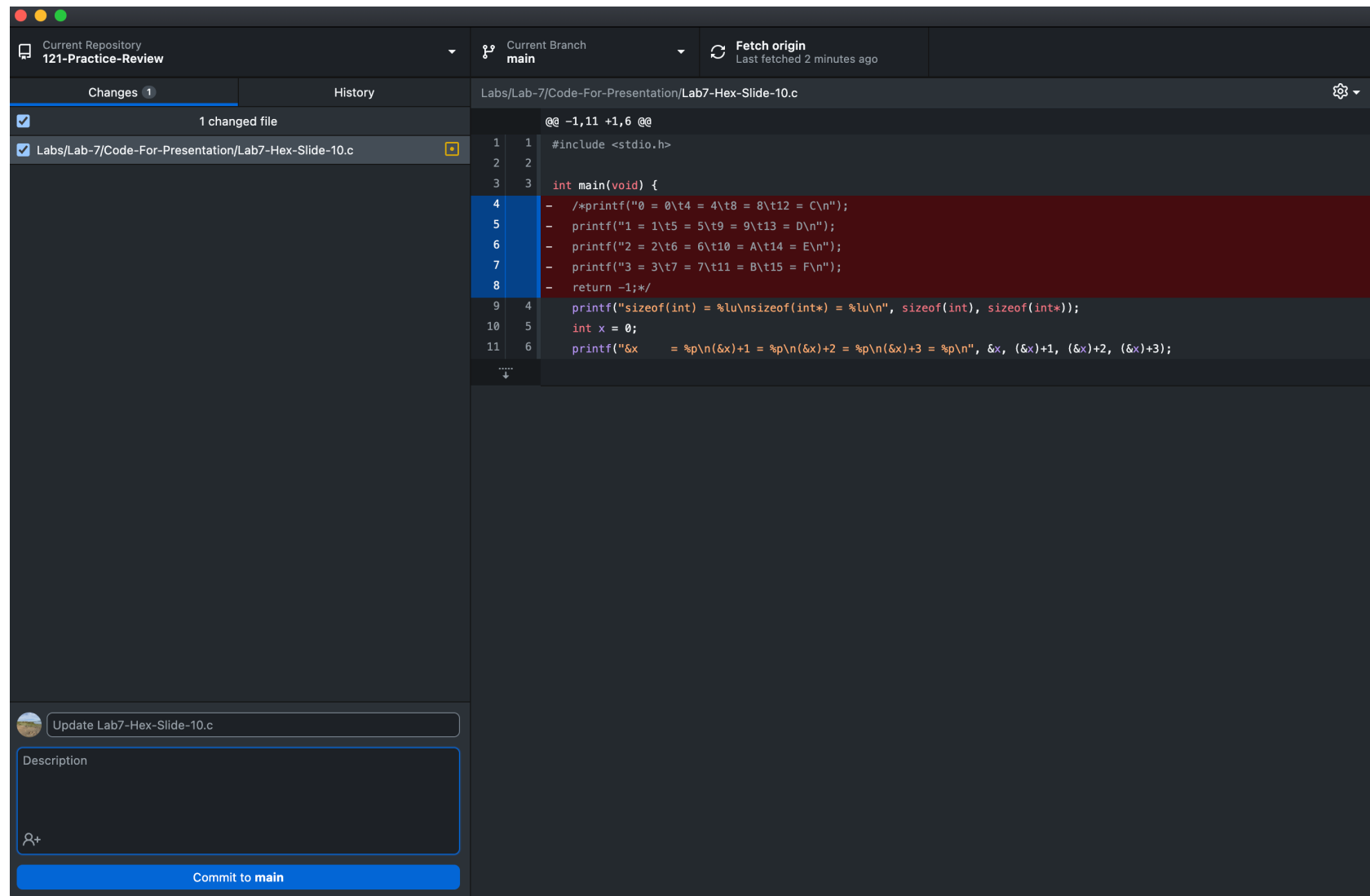


# Push: Push all local commits to the server.





# Example of modified file (non-binary)



The screenshot shows a code editor interface with a dark theme. At the top, there's a header bar with 'Current Repository: 121-Practice-Review', 'Current Branch: main', and a 'Fetch origin' button. Below this, a tab bar shows 'Changes 1' and 'History'. The 'Changes' tab is active, showing a list of changes with a checkbox and the file path 'Labs/Lab-7/Code-For-Presentation/Lab7-Hex-Slide-10.c'. The main area displays a diff view for this file. The diff shows a comparison between the current state and the previous state, with line numbers on the left. The code is in C and includes standard headers and a main function. The diff highlights changes in the main function, specifically in the printf statements and the return statement. The bottom of the editor has a commit section with a description field, a commit message field, and a 'Commit to main' button.

```
@@ -1,11 +1,6 @@
1 1 #include <stdio.h>
2 2
3 3 int main(void) {
4 4 - /*printf("0 = 0\t4 = 4\t8 = 8\t12 = C\n");
5 5 - printf("1 = 1\t5 = 5\t9 = 9\t13 = D\n");
6 6 - printf("2 = 2\t6 = 6\t10 = A\t14 = E\n");
7 7 - printf("3 = 3\t7 = 7\t11 = B\t15 = F\n");
8 8 - return -1;*/
9 4 printf("sizeof(int) = %lu\nsizeof(int*) = %lu\n", sizeof(int), sizeof(int*));
10 5 int x = 0;
11 6 printf("&x      = %p\n(&x)+1 = %p\n(&x)+2 = %p\n(&x)+3 = %p\n", &x, (&x)+1, (&x)+2, (&x)+3);
```

Simply setup a repo using the above instructions, then:

- Create a VS Project in the folder. Name it PAX.
- Be sure to setup a structure that allows you to have multiple folders in here for all PAs.
- Share the repo with me so I can edit (I will be publishing comments through this).
- Share with swiftlydesigner for RW privileges. (See GitHub help for sharing).
- If you have problems with anything in the included documents, email me or look at GitHub's help pages (they are helpful, unlike many "help pages")