

Lesson

Monday

Introduction to Programming

(/introduction-to-programming)

/ Git, HTML and CSS (/introduction-to-programming/git-html-and-css)

/ VS Code Workflow: Pair Programming In-Person or with Live Share

Text

Now that you've had an opportunity to practice with Git, it's time to start coding our first HTML. While you'll still be using Git commands to commit and push your code, you'll spend most of your time coding in VS Code. Let's review VS Code workflow expectations for in-person and online students.

VS Code Workflow for In-Person Students

VS Code workflow for in-person students is straightforward:

- Create your project folders and files in the command line.
- Write and edit your code in VS Code.
- Make sure to switch who is driving and who is observing regularly.
- Use Git to commit your changes regularly.
- Push your code to a remote GitHub repo regularly. Computers do accidentally crash or get unplugged, and you can lose all of your work if it is not pushed to a remote repo. A good time to push is before you take a break.

- When you are finished with your project, make sure that all students in your pair or group of three have the project pushed to their own GitHub account.
- At the end of the class, make sure that any projects or other files that you want to hold on to are saved to your GitHub account or emailed to yourself. When Epicodus computers are shut down or restarted, they are reset and your projects will be deleted.

VS Code Live Share Workflow for Online Students

Coding in VS Code means it's time to use VS Code Live Share so you and your pair can both work in the same editor.

After practicing VS Code Live Share this morning, you should be ready to go — and hopefully you've figured out any technical issues that might've come up in the process.

Before we continue, let's go over the protocol for alternating hosting and driving in VS Code Live Share. You will be following this protocol every day you pair program with your peers remotely.

- **Before coding, decide who is going to host the Live Share session.** That person will need to create a directory for the new project, `cd` into that project, and start a new Live Share session. Don't worry — if you aren't the host, you'll have an opportunity later in the day to be the host so you can get more practice.
- **Before moving on, determine a general time when you will alternate hosting.** The least disruptive way to do this is to switch hosting once per day (lunchtime is a good midpoint). While we don't recommend alternating hosting too often, it's ultimately up to you and your pair to decide. If one of you is having technical difficulties hosting, the other person may be the host for most or all of the day.

- **When it's time to make your first commit, the *host* will create a new repository in GitHub.** Since any Git commands related to committing and pushing code will be happening in the host's terminal, they will be the only one that can actually update the repository. We will cover how to make commit trailers to attribute commits to multiple people in a future lesson. Don't worry about that yet, though.
- **Switch drivers (who is typing) every 20-30 minutes.** You do *not* need to switch hosts to switch drivers — remember, you can both type in VS Code together when you are using Live Share. In fact, it's not recommended to switch hosting too frequently. Keep your eye on the time and practice good pairing etiquette to make sure you both get a chance to drive. If you're the driver, you should proactively offer to let the other person drive. If you are not driving, you may need to remind the driver of the time if they are not paying attention. It could be helpful to use a timer to track the driving time for each pair and remind you when to switch. It's also okay to alternate more often, especially if one person is stumped and the other person has a potential solution. *Always* make sure to communicate first before switching driving — and *never* start typing if you aren't driving and haven't gotten permission first. That would be rude! You may want to review Remote Pairing Etiquette (<https://www.learnhowtoprogram.com/introduction-to-programming/getting-started-with-working-remotely/remote-pairing-etiquette>) if you have any questions about how to be an effective pair programmer.
- **At the time you've already determined, switch hosting!** A good switching point is often when you are starting a new project. Or, after a lunch break.
- **If you are still working on the same project when you switch, follow the *Copying a Project* steps in this lesson** (<https://www.learnhowtoprogram.com/introduction-to-programming/git-html-and-css/practice-github-remote-repositories>) to recreate the existing repository in the previous host's GitHub account in a *new* repository in the new host's account. Then proceed as normal, except the new host

will now be committing and pushing code in their *own* repository. You probably won't have to do this in the first course section of the program because you'll be creating many smaller projects throughout the day and there are many convenient points where you can switch hosting and start a new project.

- **At the end of the day, fork any projects that aren't already in your GitHub account by following the steps on *Forking* in this lesson**

(<https://www.learnhowtoprogram.com/introduction-to-programming/git-html-and-css/practice-github-remote-repositories>). This way, you'll have all projects you worked on in your *own* account where you'll easily be able to access them. Your contributions will also be added in GitHub once you start using commit trailers in the next class session.

Previous (/introduction-to-programming/git-html-and-css/practice-command-line-git)

Next (/introduction-to-programming/git-html-and-css/hypertext-markup-language-html)

Lesson 12 of 64

Last updated February 28, 2023

disable dark mode



© 2023 Epicodus (<http://www.epicodus.com/>), Inc.