Contents

Getting Started

This guide is written with the assumption that the workshop participant (henceforth to be referred to as "you") has had no coding experience whatsoever and no familiarity with developer tools. We at learn-a-bit understand that is going to rarely be the case, but since we don't know exactly what you DO know, we felt it safest to include everything we can think of. On the following pages you will find guides to help you get up and running. Some third-party instructions explained the necessary topics well enough. In those cases, we merely referenced the external sources.

To get started, you will want to access the starter files. They are stored in a repository on GitHub. Don't know what GitHub is? No problem. There's a guide for that in a later section.

You will also need a text editor. Most developers use an IDE (see the Developer Tools section). You can even use a basic text editor, but you will be missing out on all the extra features you get with an IDE. One caveat, you should NEVER use a word processor application (MS Word, etc.). These programs add invisible characters to the document. When you try to execute the code, your web browser will see these characters and will error out. As an alternative, there are online text editors available as well. There is nothing to download or install since it runs in your browser and it makes accessing your code from any computer with an internet connection easy. We recommend CodePen if you choose to go this route as we will be using this as one of the available tools for code submission and review.

If you choose to use and install an IDE, you may want to browse some extensions. These add extra features to your IDE that make your coding easier. There is a list of some of our favorite extensions in the developer tools section below. Extensions aren't necessary, but rare is the developer that doesn't use them.

Once you have completed your work, you will need to make the files available for review and to be viewed as a web page. We have two options available: GitHub and CodePen. Guides on submission are on the next few pages.

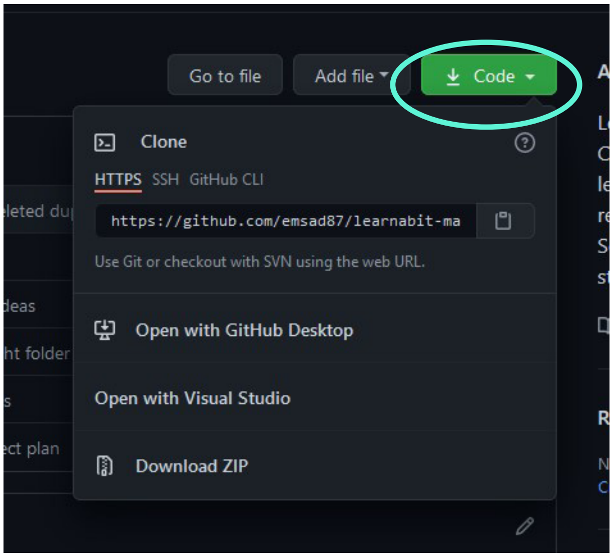
GitHub

To access the starter files and if you choose to post your files on GitHub for code review, you will need to start by making an account. Go to github.com and from there follow the instructions to create an account.

To start working, you have two options available. You can either fork the learn-a-bit repo on GitHub to your own account, or you can download the starter files as a zip and decompress to your local machine.

Download

To download the source files as a zip, you will need to go to the repository on GitHub that has the code. From the main page of the repository click on the button with the arrow that says “code”. Select “Download ZIP” and save the archive to your desktop. If you’re not using a local IDE on your computer but intend to work completely online, you can skip the rest of this section on GitHub and jump ahead to the CodePen section.

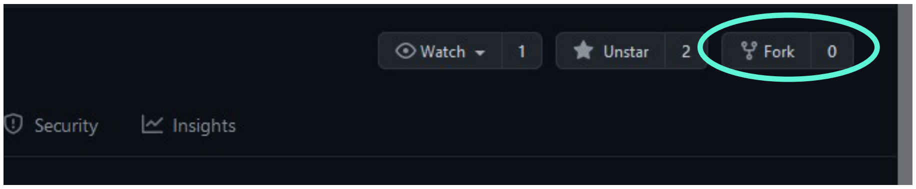


https://github.com/emsad87/learnabit-ma

Use Git or checkout with SVN using the web URL.

Forking

Forking is where you create a new code repository from an existing one but that is still linked to the original. It allows you to make changes without effecting the original code base while still maintaining a reference to it. Go to the repository on GitHub that has the code. Click on the fork icon and the repository will be forked to your own account. You will “own” the fork and will be able to make whatever changes you want without them being reflected in the original files on the learn-a-bit repository. You can now clone the repository to your local environment and push the change.



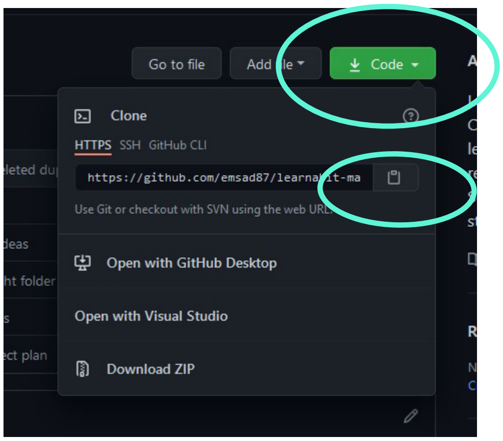
Version Control

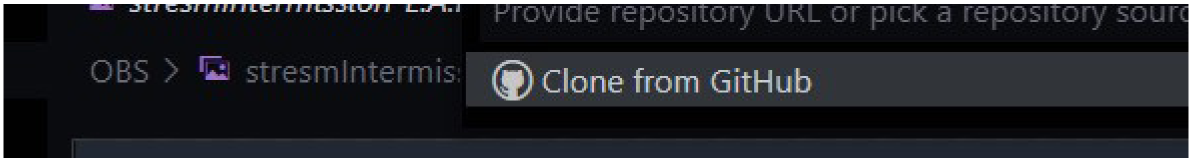
Whether you forked the repository or downloaded the files you will need to set up git on your computer if you want to push your changes to GitHub you will need to install git and configure your IDE. Here is a good guide on how to do just that:

https://www.jcchouinard.com/install-git-in-vscode/

It has instructions on how to set up VS Code to use git as well. You do not need to understand git to use GitHUb, but if you're interested in learning more, check out the links section at the end of the guide.

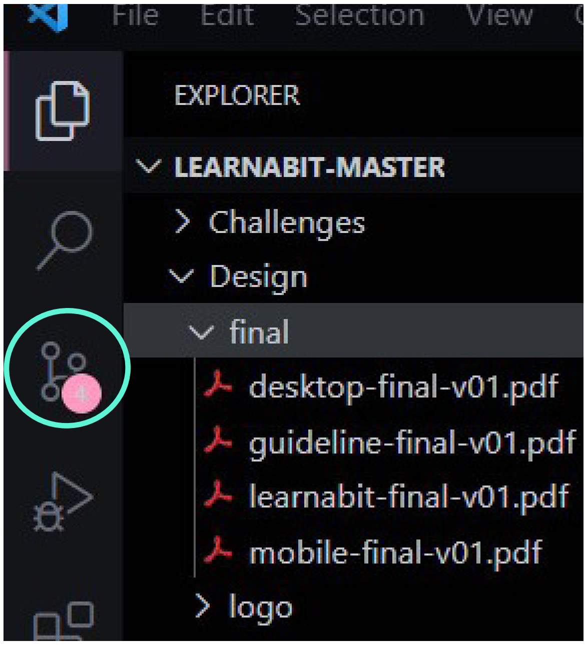
Once you have git set up and your IDE configured to work with GitHub, you can now clone the fork you made to your computer. From your browser, navigate to your GitHub account and open the forked repository. From the main page of the repository, click on the button with the arrow that says "code". Select "HTTPS" and then click on the clipboard icon next to the web address just below. With the address copied to your clipboard, open VS Code on your computer. Hit F1 and type in "clone" into the search bar. Click on "Git: Clone" and then paste the URL you just copied and hit enter. VS Code will now ask you where you want to save the files. Choose a location and it will create a new folder named after the repository on GitHub and all the files and folder structure will be downloaded to your computer. VS Code will then ask you if you want to open the folder. Say yes and you will see the parent folder and all the files located in the explorer pane to the left. Just click on the file you want to work on and you're good to go.

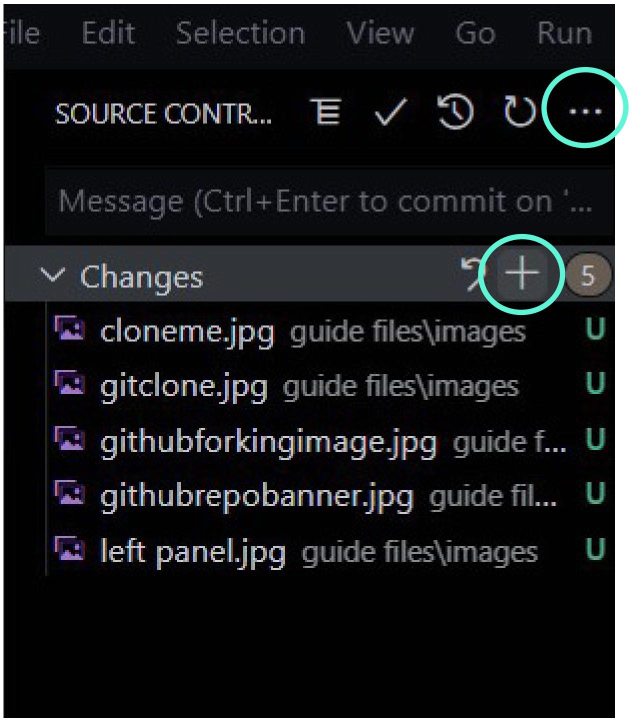




Commit and Push

At the very left side of the VS Code window is a list of icons. The one that looks like a flow chart is the version control pane. When you make a change to a file and save it, little numbers will show up over the icon, like this. Clicking on this icon will take you to the version control pane. You will see all the files you modified or added to the project listed under a text box to the left. Clicking on the "+" sign next to the files will "stage" the changes or you can click the "+" next to "changes" and stage all the changes at once. You will now need to commit. In the text box just above "changes" you will need to enter a short message. If you want, you can just put in a space, but the commit message is there to help identify what was changed at this commit. You will then click on the three dots just above the message, hover over commit, and then click commit. You have just now taken a snapshot of your code in its current state. This is great if you want to go back to a previous version of one or more of your files later. Commits are like saving copies of previous versions of your files without having a bunch of loose files sitting around. The next step will be to push the changes up to your online repository. Without leaving the source control pane, click on the three dots at the top again and then click "push". This should then push all changes and new files you staged and committed up to your online repository on GitHub.




File
Edit
Selection
View
Go
Run
SOURCE CONTR...
Message (Ctrl+Enter to commit on '...
Changes
5
cloneme.jpg guide files\images U
gitclone.jpg guide files\images U
githubforkingimage.jpg guide f... U
githubrepobanner.jpg guide fil... U
left panel.jpg guide files\images U

CodePen

If you're not going to use an IDE to code, you will want to use an online editor. This will allow you to write code and then provide a link to your work, allowing others to view it. A popular choice is CodePen. It is easy to use and has good support. To save your files, you will need to create an account if you don't already have one.

go to: codepen.io

Click the "Sign Up" button in the top right corner of the screen and follow the instructions to create a new account. Once logged in you can create a new pen. Click on "Pen" at the top left of the screen under the section where it says "create". This will take you to a new screen with three boxes at the top and one at the bottom. You will enter your code at the top and the web page will render at the bottom. Click "Untitled" at the top left to change the name of the pen you're working on. We suggest something that aligns with the workshop. Once you're ready to share, click the button "share" at the bottom right. This will copy the URL to your clipboard and you will be able to paste that to social media when you're ready to share.

Development Tools

You’re going to want to choose some sort of a text editor. There are plenty of IDEs (Integrated Development Environment) available for free online. As to which one to use, the choice is up to you. All of the guides here are written with Visual Studio Code in mind but any decent IDE should offer just about the same functionality and have plenty of online documentation available. As mentioned earlier in the guide, you should never use a word processor app (MS Word etc.). If you’re not sure what editor to use, just pick one. Most are pretty good and you can always switch to another one later if you like. See the end of this section for articles about IDEs if you want more information.

Modern web browsers have a feature called developer tools. You can open developer tools in Chrome by clicking on the three vertical dots at the top right of the browser, clicking “More Tools”, and clicking “Developer Tools”. There is also a shortcut (Shift+Ctrl+I on PC, Cmd+Opt+I on Mac. Firefox’s developer tools can be found by clicking on the 3 horizontal bars at the top right, clicking “Web Developer”, and clicking “Web Developer Tools”. The shortcut is the same as Chrome.

Lists of Things

Articles on IDEs

lambdatest.com/blog/best-ide-for-web-development

codeburst.io/what-makes-vscode-so-popular-11e1b3c59ffd

Visual Studio Code Extensions

A great way to enhance your coding experience is to use extensions. These add-ons to the core IDE can help a lot. Below is a list of some of the Learn-a-Bit teams favorites. You can download and install extensions by clicking on the icon on the left that looks like a small 2 x 2 grid with one of the cells floating away. Click on this icon and you will be able to search or browse for extensions withing the left pane. You will need an internet connection to find and install these as they are not built in.

Auto Rename Tag - when you change the open or closing tag of an element, auto-changes the other

Live Server - opens your html in your default browser and auto updates after you save

Rainbow Brackets (Emsad uses) - this, and the following extension paint matched open and closing tags the same color (you only need one of these extensions. If you install both, the universe explodes).

- or -

Bracket Pair Colorizer 2 (Chris uses)

Indenticator - highlights the indent depth. VERY handy for complex html and JS files

htmltagwrap - does one thing, gives you a quick keystroke to wrap selected text in html tags.

The following extensions are good but may require a bit more configuration than a new programmer is ready for

Live Share - lets you collaborate online THROUGH VS Code. Pretty cool.

Prettier - auto-formats your code to make it… prettier. It may do something you don't like or that may be weird for a beginner and some people seem to have trouble with configuring it to behave, though, so this may be a bit advanced for a beginner.

Learning Websites - Free (mostly)

freecodecamp.org

Totally free! This site walks you through the very basics, one step at a time. No boring lectures here, each lesson is a code-along. If you want to learn, you've got to write (code).

kevinpowell.co

Lots of free stuff to help you get started with the fundamentals. Some of the more advanced stuff is behind a paywall.

wesbos.com

Wes Bos is pretty well known for his JavaScript 30 course. In fact, we have a "javascript30" Developer Collective club here at Mintbean! Tons of free stuff to help you along.

w3schools.com

This one can be a bit of a polarizing choice. I heard that w3schools used to have a few typos and broken code back in the day, but that looks to have neem fixed and this site seems to be making a comeback. I find myself looking a thing or two up here from time to time. What's really cool is they have a "try it yourself" integrated code editor in the site so you can practice while learning.

developer.mozilla.org

When people say, "read the manual", this is what they mean. Although not "officially" the documentation for the code, MDN Web Docs is based on it. The docs themselves can be a little overwhelming for the beginner, but there are great, free tutorials on the site too.

youtube.com

Launch YouTube and search for any topic you want. There are more free tutorials than we could even begin to catalogue, and more every day. You may find a video showing you how to do a specific thing, or a whole series that will teach you a language from the ground up. Look around and see if you find a teacher you like.

Learning Websites - Not Free (mostly… but check your library!)

Some public libraries have contracts with online teaching sites that allow them to offer their courses for free to the local community. I have access to both Treehouse and LinkedIn Learning through an online digital public library located in my county. Check your library. You never know.

teamtreehouse.com

linkedin.com/learning