

## Workbook Instructions

Here's how to proceed with each lesson and lab in this course:

### HTML File

- Open the current lesson **.html** file in your Code Editor.
- Scroll down to confirm that the **script** tag is importing **FINAL.js**.
- Preview the **html** file in the Chrome browser.
- Right-click **Inspect** to check the Console for the final output.
- Back in the html file, switch to importing the **PROG.js** file, which is a blank copy of **START.js**.

### JS Files

- Each lesson has 2 JS files: **START.js** and **FINAL.js**.
- Write your code for each lesson in **START.js**, where a little starter code together with detailed instructional comments are provided.
- Each numbered step in a lesson has code for you to type.
- Each step explains what you are to type next, and why.
- Check the Console with each step to make sure your code is working.

### Labs

- Most lessons are accompanied by a **Lab** for you to do after you have completed that lesson.
- Labs are challenges, in that the code steps are not provided. Instead, you are given prompts and must figure out what code to type.
- Unlike lesson html files, lab html files do NOT have separate JS file to import. In order to streamline the files system, all lab JS code is typed in the Lab html page, in the script tag.
- All lab .html files have a companion Solution html file. This is where you check the answers—but do so only after having given the lab your best effort.

### Submitting Lab work as Homework

- Labs are homework. When you are done, check the Solution file, but also submit your work through the class Slack Channel, or by whatever means your instructor provides.
- The Solution JS code is not necessarily the only way to do something. By submitting your code—which may differ from the “official solution”, your instructor can give you valuable coding tips and feedback on best practices.

### Markdown Files (.md files)

- Each lesson has a markdown file with extension **.md**.
- The markdown file contains the step-by-step instructions for the lesson or lab.
- This file can be rendered in your VSCode Editor by typing **Command-Shift-V** (Mac) or **Ctrl-Shift-V** (Windows).
- You may want to use a vertical split screen view, with your coding page (PROG.js) in one pane, and the readme file for that lesson in the other pane.
- To enable vertical split screen in VSCode, choose **View > Editor Layout > Split Up**.

- Both lessons and labs have companion markdown files.

## PDF files

- **Noble-Desktop-JavaScript-Bootcamp-Workbook.pdf** is a collection of all the many markdown files for each lesson and lab. It is assembled all in one place for your convenience.
- Workflow option: you may prefer to keep your whole screen for coding and to not to split your screen between the JS file and the markdown file. In that case, you may want to set up a second computer / monitor and keep the the **PDF** file open on that other monitor.
- PDF files can be viewed in Adobe Acrobat Reader, in Preview on Mac, and in the Chrome browser.

## Making a PDF from Markdown file

- In **Extensions** install **Markdown PDF**
- Open a **.md** (markdown) file in VSCode and click inside the file.
- From Menu, choose **View > Command Palette**
- Click in **Command Palette** and type a **?** to get some choices dropdown
- Choose **Show and Run Commands**
- Start typing the word **Markdown** to jump to that area of the menu
- Choose **Markdown PDF: Export (PDF)**
- After a few seconds--then your PDF will appear in the same folder as the Markdown file, and with the same name.
- You can also right-click in Markdown file and choose **Markdown PDF: Export (PDF)**

## Viewing a PDF from with VS Code

- In **Extensions** install **vscode-pdf**
- Click on a PDF in your VSCode Explorer: the PDF will load in a VSCode Window.