Name _____ Period _____ Role (Circle one) Programmer/Driver

Name _____ Period _____ Role (Circle one) Programmer/Driver

Getting Started with Javascript

Your Tasks (Mark these off as you go)

- ☐ Install Visual Studio Code
- ☐ Explore Visual Studio Code
- ☐ Have Ms. Pluska check off the above tasks
- ☐ Create your first javascript project
- ☐ Add comments
- ☐ Receive credit for the individual portion of this lab

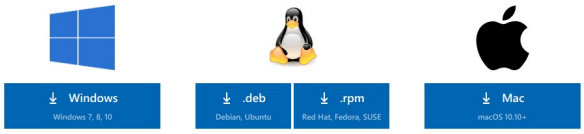
☐ Install Visual Studio Code

Visual Studio Code is the Integrated Developmet Environment (IDE) we will be using for this course. As your skills develop and projects grow more complex, you grow to appreciate and love the features an IDE can offer.

Navigate to the following link to download Visual Studio Code

https://code.visualstudio.com/Download

Select the appropriate version for your operating system



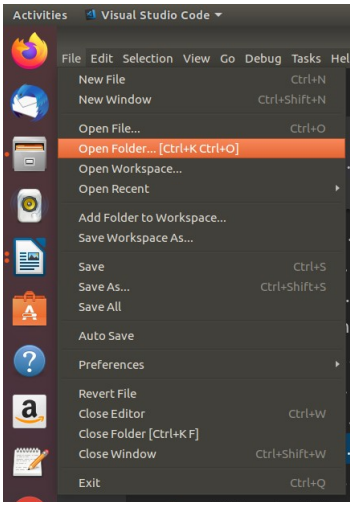
Linux users only

Once you have downloaded the correct file consult Ms. Pluska for further instructions.

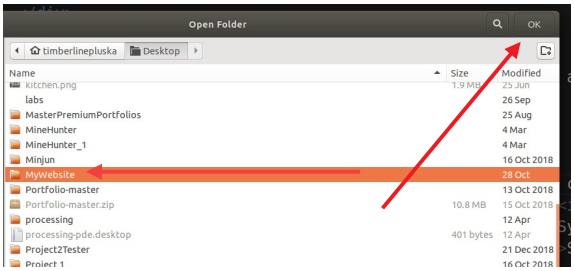
☐ Explore Visual Studio Code

Open a project in Visual Studio Code

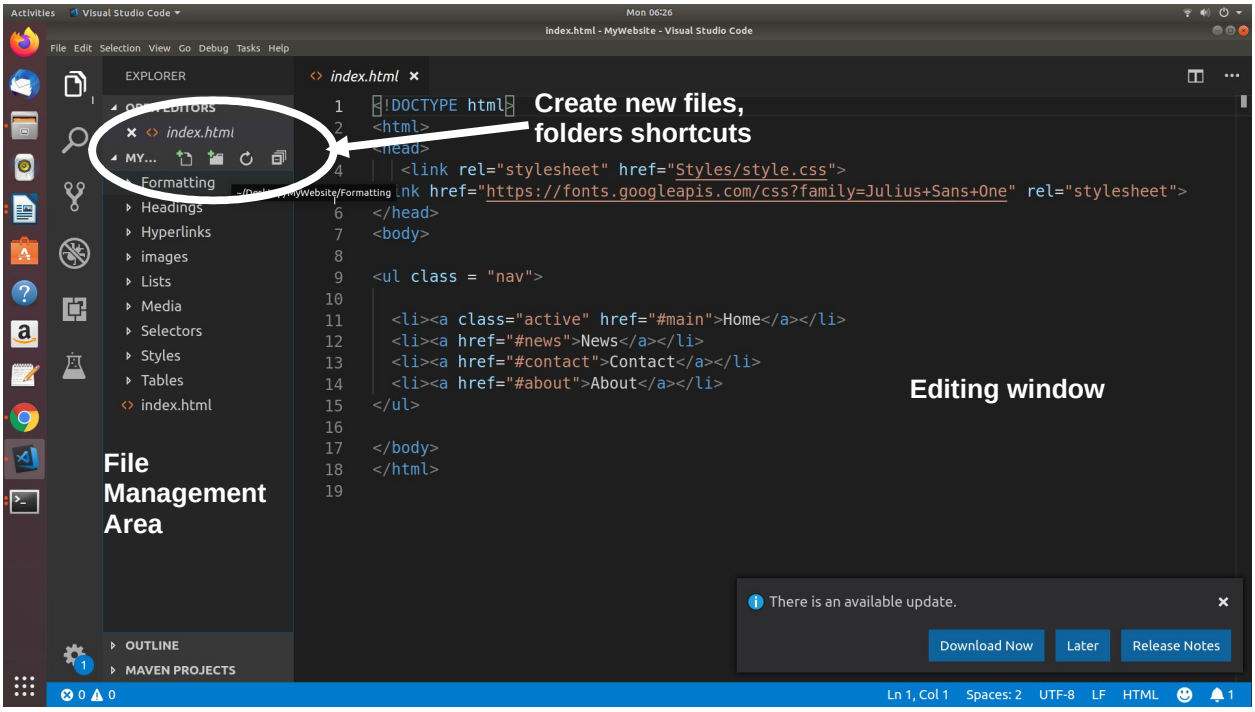
Open Visual Studio Code (VSCode). Select *File* from the menu and then the *Open Folder* option.



Navigate to your Website directory and click OK.



The screen shot below indicates the areas where you will be spending most of your time. For now start clicking around and explore the workspace



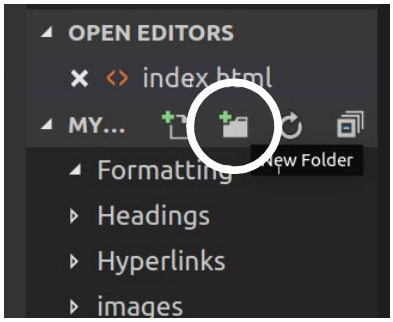
☐ Have Ms. Pluska check off the above tasks

Before you continue have Ms. Pluska check off the above tasks

Do not continue until you have Ms. Pluska's (or her designated TA's) signature _____

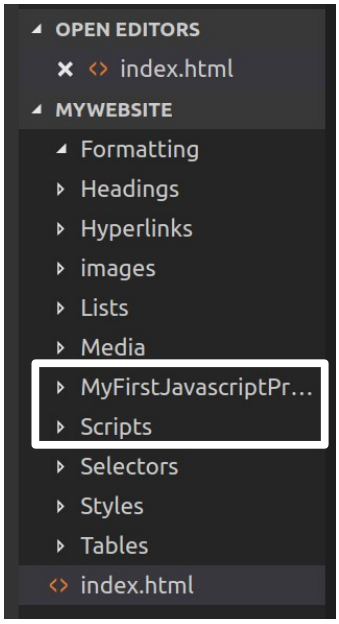
☐ Create your first javascript project

Create the required files



All modern browsers are capable of rendering javascript, which makes developing in javascript quick and easy. Visual Studio code will help you stay organized and will speed up your development process. Below are the steps for creating your first project.

- Create a new directory in your main website directory. Do this by clicking the new directory icon in the file shortcut menu. Call the new directory *Scripts*.
- Create another new directory in your main website directory called *MyFirstJavascriptProgram*. Your directory listing should look similar to the screen shot below,

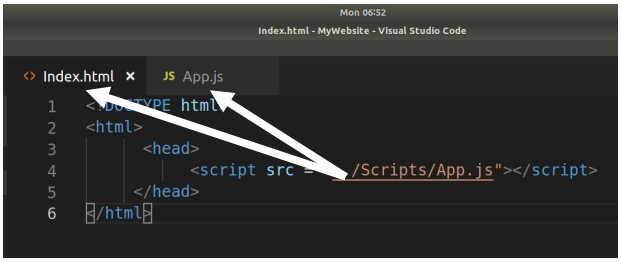


- Open your *MyFirstJavascriptProgram* directory. Create a new file inside this directory called *Index.html*
- Open your *Scripts* directory. Create a new file inside this directory called *App.js*
- Open your Index.html file in the editor, add the following code. The code below will load the javascript you will write in your App.js file.

Index.html

```
<!DOCTYPE html>
<html>
      <head>
            <script src = "../Scripts/App.js"></script>
      </head>
</html>
```

Things should look like the screen shot below. Notice you can easily toggle back and forth between your files using the tabs.



Using the console

The console is a panel that displays important messages, like errors, for developers. Much of the work the computer does with our code is invisible to us by default. If we want to see things appear on our screen, we can print, or log, to our console directly.

In JavaScript, the console keyword refers to an object, a collection of data and actions, that we can use in our code. Keywords are words that are built into the JavaScript language, so the computer will recognize them and treats them specially.

One action, or method, that is built into the console object is the .log() method. When we write console.log() what we put inside the parentheses will get printed, or logged, to the console.

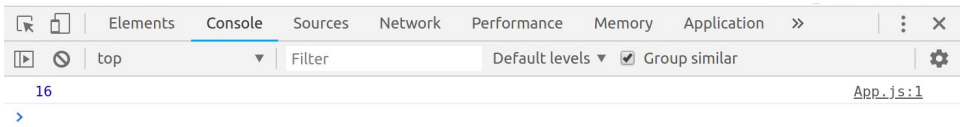
It's going to be very useful for us to print values to the console, so we can see the work that we're doing.

```
console.log(5);
```

The example above logs 5 to the console. The semicolon denotes the end of the line, or statement. Although in JavaScript your code will usually run as intended without a semicolon, we recommend learning the habit of ending each statement with a semi-colon so you never leave one out in the few instances when they are required.

To view the log simply right click on the webpage where your javascript is running, then select inspect. The console log will appear as a window in the developer tools menu.

- ➔ In your App.js page,
  - o write a line of code that logs your age.
  - o Write a line of code that logs your name. For example, console.log("Pluska");
- ➔ Navigate to your Index.html page and open it.
- ➔ Navigate to your developer tools and open the console. It should look something like the screen shot below. Notice the number 16 in the screen shot below. What did you type?



☐ Have Ms. Pluska check off the above tasks

Before you continue have Ms. Pluska check off the above tasks

Do not continue until you have Ms. Pluska's (or her designated TA's) signature _____

☐ Add comments

As we write JavaScript, we can write comments in our code that the computer will ignore as our program runs. These comments exist just for human readers.

Comments can explain what the code is doing, leave instructions for developers using the code, or add any other useful annotations.

There are two types of code comments in JavaScript:

- A single line comment will comment out a single line and is denoted with two forward slashes // preceding it.

```
// Prints 5 to the console
console.log(5);
```

You can also use a single line comment to comment after a line of code:

```
console.log(5);  // Prints 5
```

- A multi-line comment will comment out multiple lines and is denoted with /* to begin the comment, and */ to end the comment.

```
/*
This is all commented
console.log(10);
None of this is going to run!
console.log(99);
*/
```

You can also use this syntax to comment something out in the middle of a line of code:

```
console.log(/*IGNORED!*/ 5);  // Still just prints 5
```

➔ On the first line of your App.js use in line comments to type your name, date, and period (See the example below)
➔ Use block quotes to describe what your code does. (See the example below)

```
//firstname lastname
//date
//period

/* My program prints 5 and my name to the console*/
```

☐ Receive Credit for the group portion of this lab

- Make sure both you and your partner have completed the above tasks
- Indicate the names of all group members.
- Have Ms. Pluska check off group tasks
- Submit your lab to the needs to be graded folder to receive credit for the group portion of this lab.
- Do not submit your lab until you have Ms. Pluska's (or her designated TA's) signature

_____