How to build the Hello World Application!

This is a documentation to guide a newbie on developing Hello World application.

The Hello World program generally is a computer program that outputs or displays the message "Hello, World!" to the user. In this application we used HTML(Hyper Text Markup Language) as the structure (body) of the application, CSS (Cascading Style Sheet) as the styling solution, and JavaScript to make the application more interactive. If you have previous experience on coding and have downloaded a text editor app, you might skip forward to step 3.

What do you need?

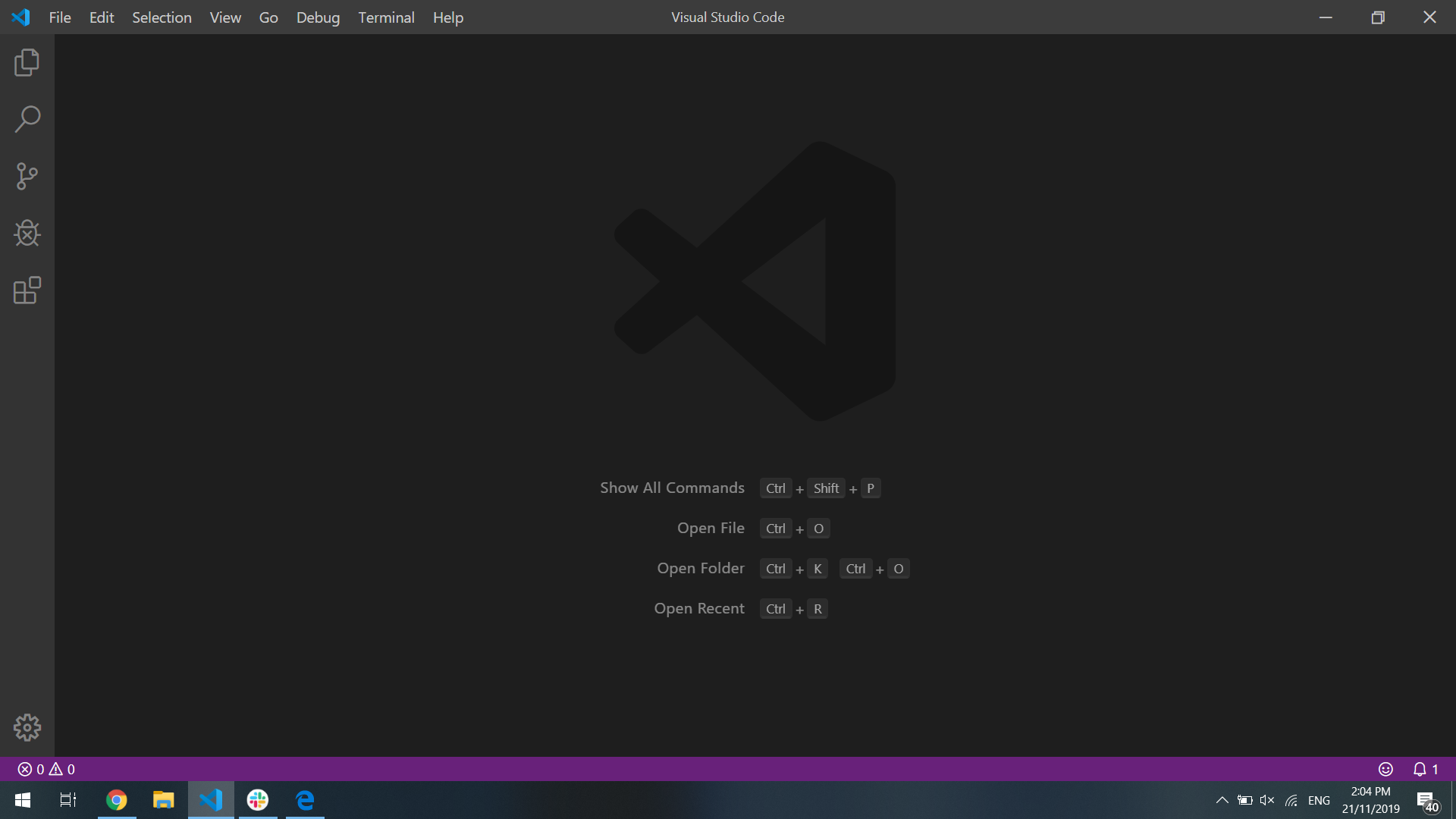
1. A computer with any Mac or Windows operating system working on it.
2. A text editor is the application used to write the script. You can use Brackets or Visual Studio Code or Notepad++. They are all free to download.
3. A proper internet connection is not really required but very helpful.

Let’s start writing your first ever coding!

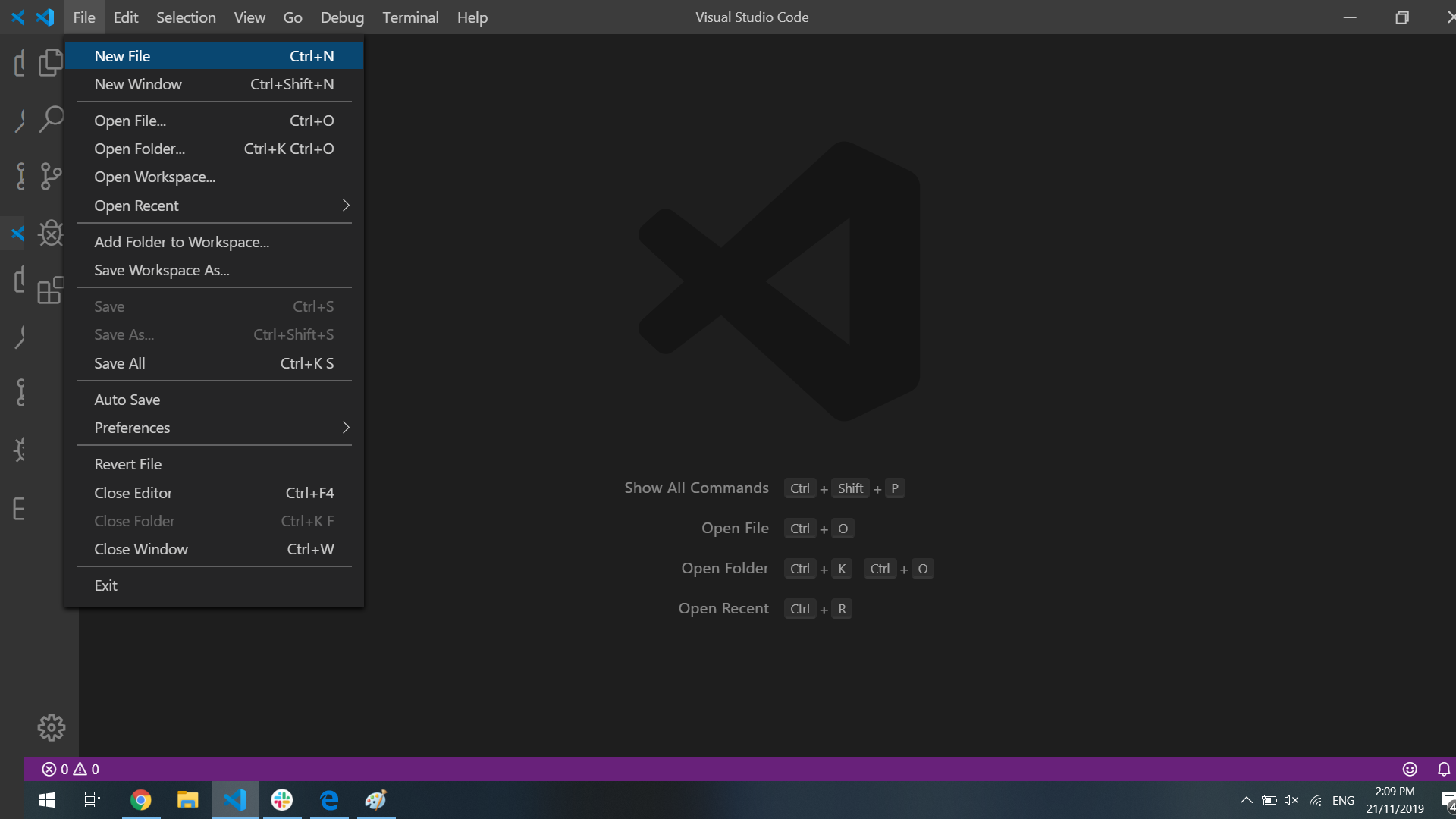
1. Once your PC or Mac has booted up, make sure you have a running text editor on it. This will be our ‘best friend’ to create, edit, debug(finding the problems), and testing of our project.
2. If you don’t have any of the text editor, you can download it at their websites. We recommend using Microsoft Visual Studio as it directly shows any errors for every file extension. Follow these steps:

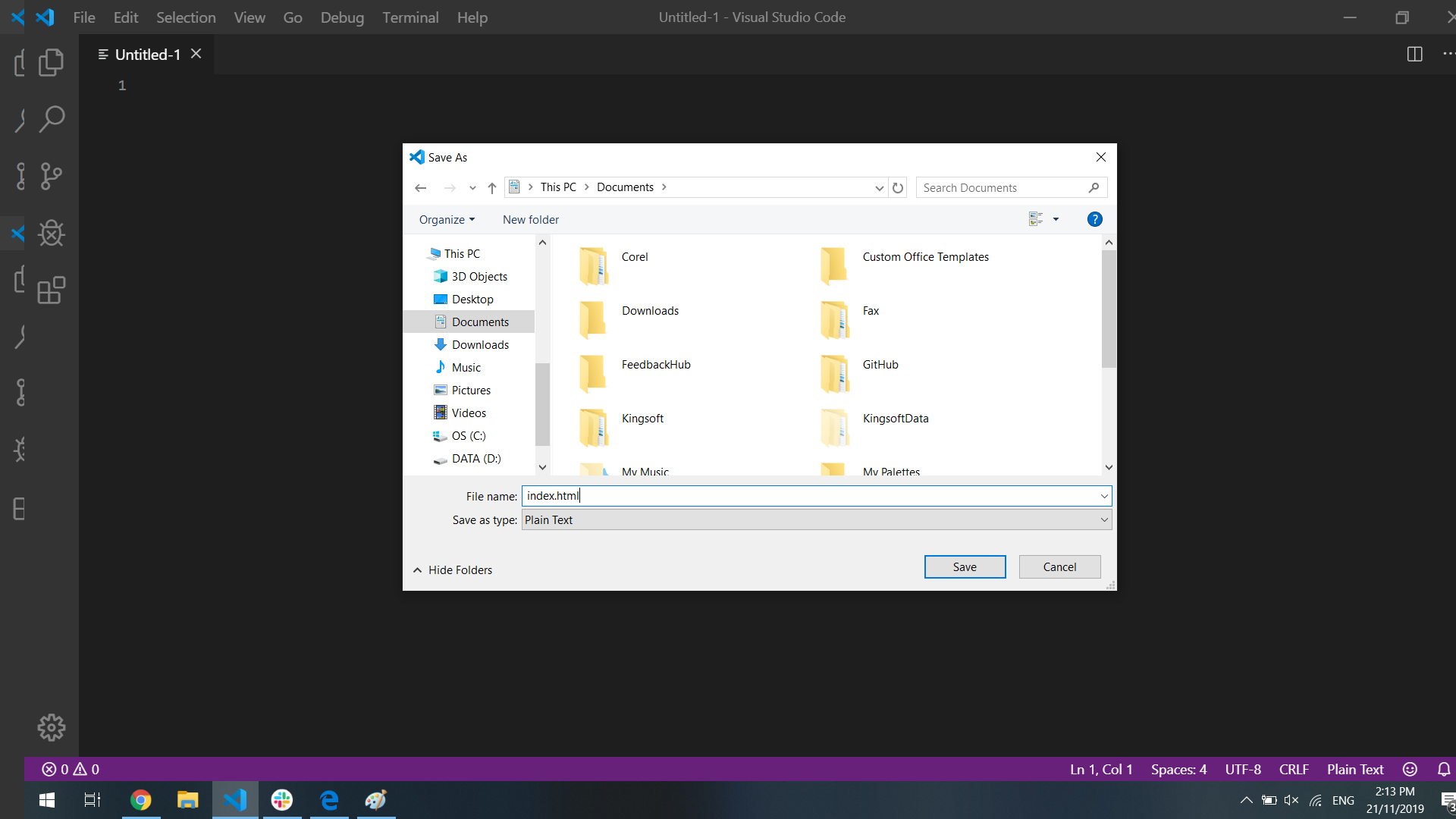
* Open your browser and type <https://code.visualstudio.com/Download>
* Choose your operating system and download the app.
* Move Visual Studio Code to Application and install it.
* Click the Visual Studio Code to open.
  + - If problem persists on Mac, we will force open Visual Studio Code by performing these:
      * On keyboard, use Command+space and type Terminal which should open terminal
      * Type this: command xattr -d com.apple.quarantine /Applications/Visual\ Studio\ Code.app

1. Once your Visual Studio Code has been installed you will see this on your screen.



Congratulations for officially entering world computer programming!

1. Next create a new working file selecting File and New File. The keyboard shortcut is Ctrl + N for Windows or Command + N for Mac users.
2. Save your working file on your directory (personal file) and name it index.html. Remember to give the file extension as it will determine the type of file. This file will be a HTML file as we are building the structural of our webpage.



1. On index.html, we can start typing in this command, once done save the file.

<!DOCTYPE html>

<html>

<head>

<title> </title>

</head>

<body>

</body>

</html>

This is the basic structural when building a web application. Each tag has their opening and closing tag. Closing tag is indicated by backslash /. Head will nest the type of document (initiated with <!DOCTYPE html>), and the title of this page(placed between <title> </title> tag for example <title>Hello World</title>. Body is where the content of the page goes.

1. Next we would like to create content of our page inside the body tag, it is located below the <head> tag .

<div>

<h1 id="display"> </h1>

<h1>Click any of these two buttons</h1>

<button onclick="myFunction1()"> <h2>Hello World</h2></button>

<button onclick="myFunction2()"> <h2> Goodbye World </h2></button>

</div>

The command is started with <div> tag, this is a container that holds contents in it. Inside it we have two <h1> tags which will display the text inside in as heading 1. To differentiate each tag we give it a class or id as their personal identification. <h1 id="display"> presents the ‘Hello World’ message and <h1>Click any of these two buttons</h1> exhibits the command for the users to read. We have 2 button tags with JavaScript function on them (we will elaborate more on the next step) where the first one sends the “Hello” message and the other one delivers the “Goodbye” message. This part is concluded with closing div tag </div>.

1. Then, we attach the script tag inside this project. This is placed between closing div tag and closing body tag.

<script>

function myFunction1(){

document.getElementById("display").innerHTML="Hello World";

}

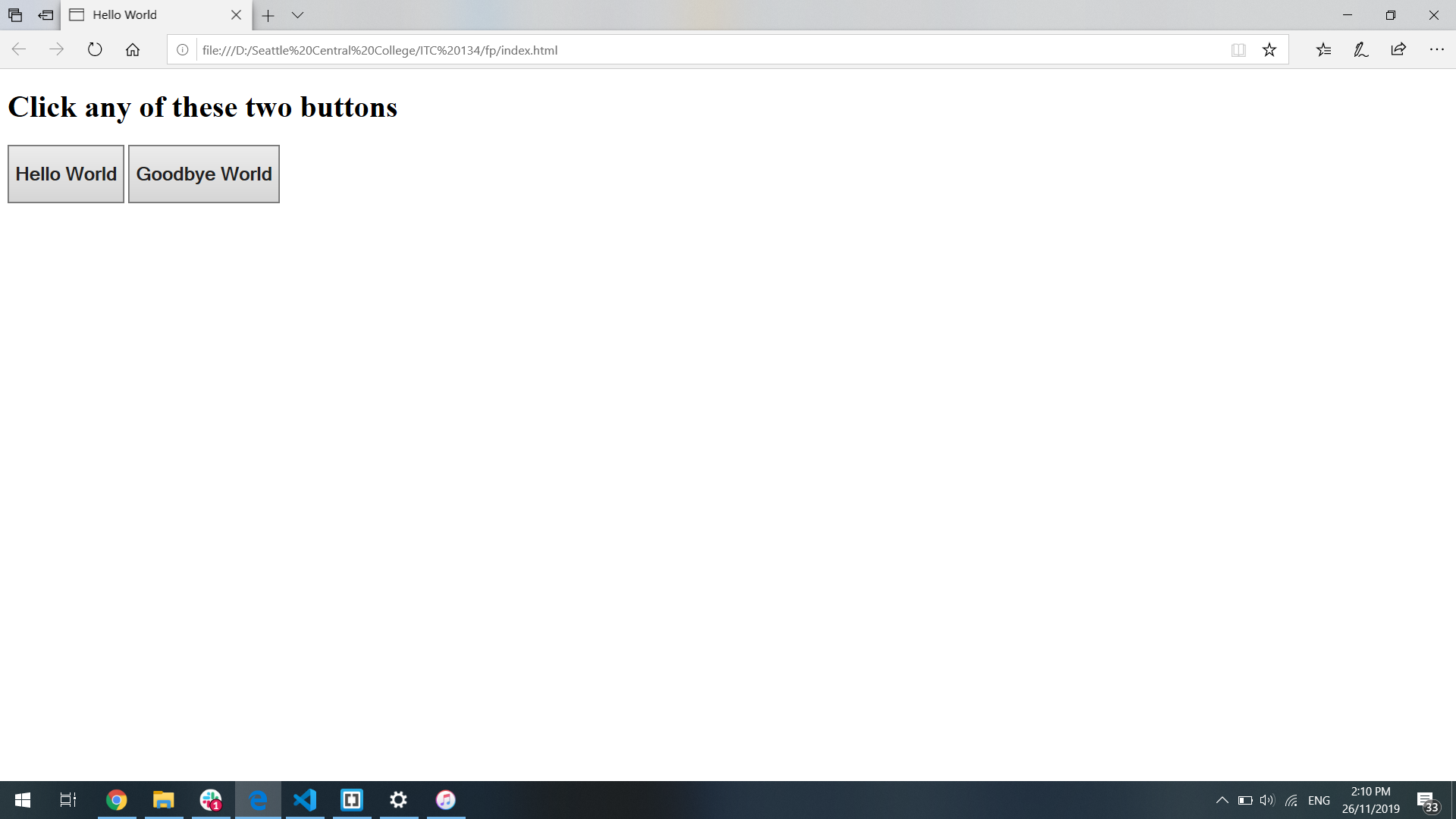
function myFunction2(){

document.getElementById("display").innerHTML="Goodbye World";

}

</script>

We have two functions which will output hello message and goodbye message, respectively. These functions are then linked with the body through onclick="myFunction1()" and onclick="myFunction2()" inside button tags. The purpose is when the user clicks on the button, JavaScript will capture the click events and display the message.

1. What we have done so far are building the basic foundation of web application by using HTML and using JavaScript to bring more interaction with the users. Below is a preview of the page.
2. It might not seem attractive as the aesthetic part has not been added yet. We need to add CSS in our page. We recommend to use external page of CSS as it can be applied to multiple pages and ease of manageability. To do so, we need to create a new file (repeat step 3) and save it as styles.css (step 4). To connect our HTML page with CSS page we need to use this command:

<link rel="stylesheet" href="styles.css" />

Place this snippet inside the head tag (it can be placed after or before title tag) and make sure the link href="styles.css" which will direct to the directory of our CSS file location.

On styles.css copy this command for our page and save it.

body {

background-color: coral;

}

#display {

background-color: chartreuse;

border-radius: 50px;

width: 300px;

text-align: center;

}

button{

border-radius: 100px;

width: 10%;

}

To select an element from the HTML we just directly mention each element on our css page followed with {}. The CSS syntax has to be inside the curly braces. Since we have given h1 an ID of display we must call the id using #display, therefore on every page where id of display is showing up will apply the CSS rules.

The end result of our web application will look like this:

