

# COMP 2406 B - Fall 2022

## Tutorial #0 (not for submission)

---

### Objectives

This tutorial aims to help you set up your coding environment and introduce you to the basics of HTML, CSS, and JavaScript. The agenda is:

- Practicing basic HTML/CSS
- Exploring Chrome Developer Tools
- Installing Node.js
- Practicing basic JavaScript programming

### Expectations

This (self-study) tutorial is not for marks – just for glory 😊. You do **not** need to submit it. If you need help completing it, please ask us on Discord or attend our office hours.

Remember to use the available resources (w3schools, Node.js documentation, Eloquent JavaScript book, lecture materials, etc.) for more information if you are struggling to complete the problems. For example, if you're uncertain how to do something in HTML or CSS, you may wish to consult w3School's references for each:

<https://www.w3schools.com/html/default.asp>

<https://www.w3schools.com/css/default.asp>

---

### Problem 1 (Create a web page)

Learning HTML is best performed by entering new HTML into a web page and viewing the results in the web browser. A web server is not necessary to create a web page that can only be viewed on your desktop or laptop computer.

Try the following:

1. Open your favorite text editor. Popular text editors include [Visual Studio Code](#), [Atom](#), [Sublime Text](#), and [Notepad++](#).
2. Create a new file.

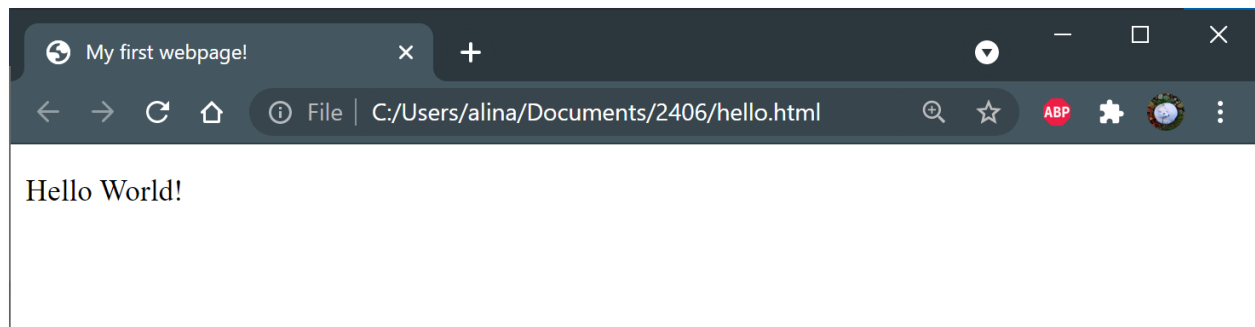
**Tutorial 0 (self-study)**

3. Enter the HTML provided below into the file. Did you know that some Text Editors will preserve the formatting of the text you copy and paste from an external word processor, such as Brightspace Editor or MS Word? This includes invisible symbols for styles and effects. If you copy/paste a code, make sure you paste it as **plain text**. Otherwise, it might not run as expected.
4. Save the file to your computer's hard drive as **hello.html**. The file extension .html is necessary for your browser to know the text file contains HTML.
5. Using Chrome or your favourite web browser, open hello.html by pressing Ctrl+O on Windows or Cmd+O on a Mac. Other ways to open the file include double-clicking the file or dragging the file into the browser.

HTML code :

```
<!DOCTYPE html>
<html lang="en">
  <meta charset="UTF-8">
  <title>My first webpage!</title>
  <body>
    <p>Hello World!</p>
  </body>
</html>
```

The Chrome browser below shows hello.html. The path to the file appears in the address bar.



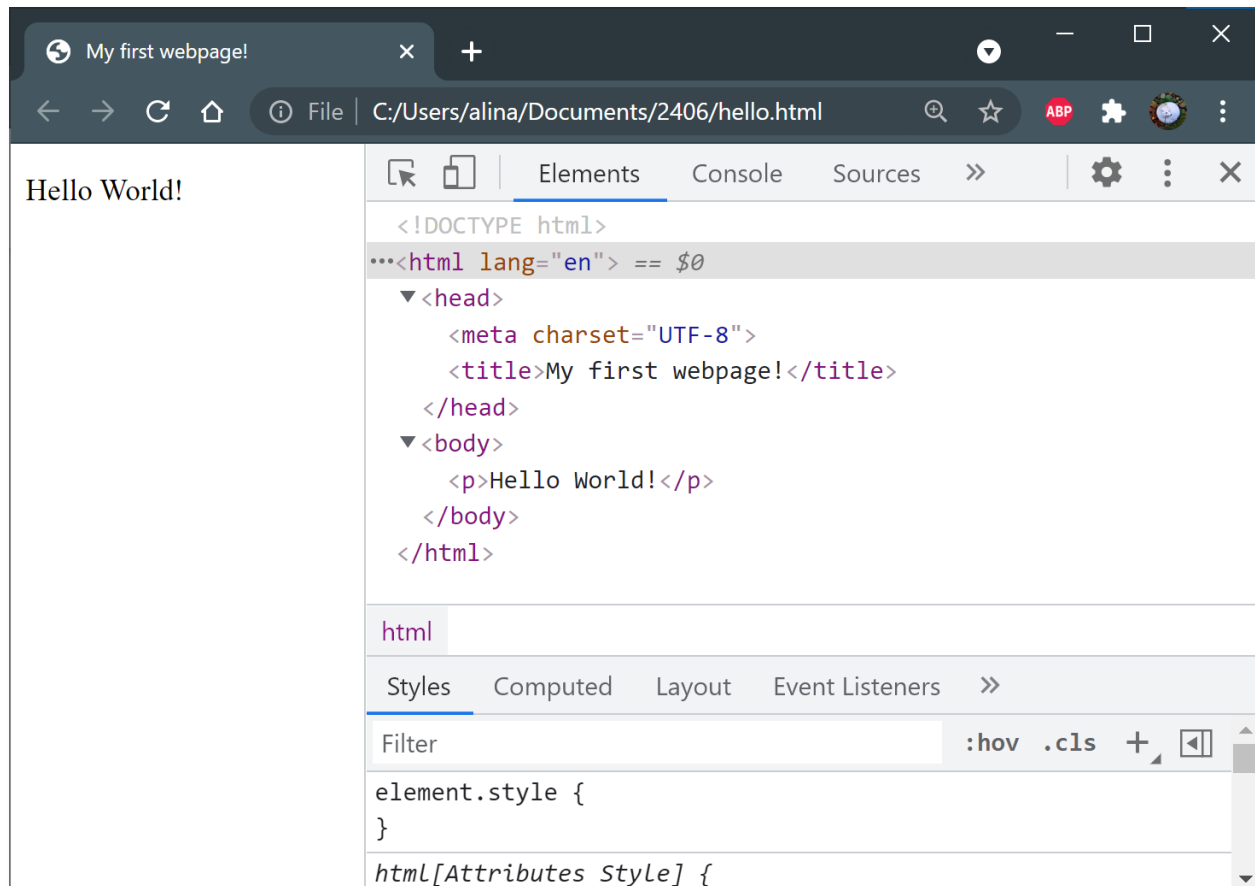
## Problem 2 (Explore Chrome DevTools)

**Chrome DevTools** is a collection of web development tools built into the desktop Chrome browser. Open the browser's development tools to view a web page's HTML elements. Pressing Ctrl+Shift+I (Windows) or Command-Option-I (Mac) opens Chrome DevTools. Other desktop browsers have similar development tools.

**Tutorial 0 (self-study)**

A browser's development tools show the browser's internal representation of a web page's HTML. The browser's internal representation and the web page's actual HTML may differ slightly.

In the screenshot below, the DevTools's Elements tab displays the browser's internal representation of the web page's HTML. The `hello.html` file does not have a `<head>` tag, but Chrome creates the head element automatically and places the `<meta>` and `<title>` tags inside `<head>`.



## Problem 3 (Practice basic HTML/CSS)

Use your text editor to open the file **index.html** provided with this tutorial.

- Resize the browser window – notice that the image will resize accordingly. Find the element that is responsible for this.
- Position the title and the paragraph to the right of the image. One way of doing this is by adding `float:left` to the styles attribute of the image.
- Modify the HTML to include `h2` and `h3` headers, bold text (use `<strong>` tag), another paragraph, links, etc.

---

Tutorial 0 (self-study)

- If you feel adventurous, add a button:  
`<button type="button">Count me in!</button>`  
(it will not do anything for now)
- Add styles to **index.html** before the `<body>` tag:

```
<style>
  h1 {
    font-size: 21pt;
    background-color:rgb(229, 217, 252);
    text-align: center;
  }
  em {
    font-size: 16pt;
    color: indigo;
  }
  strong {
    background-color: yellow;
  }
  #friday{
    font-size: 16pt;
    color: white;
    background-color: black;
  }
</style>
```

- Add some space between the image and the text. This can be done by adding the following code to the above styles:

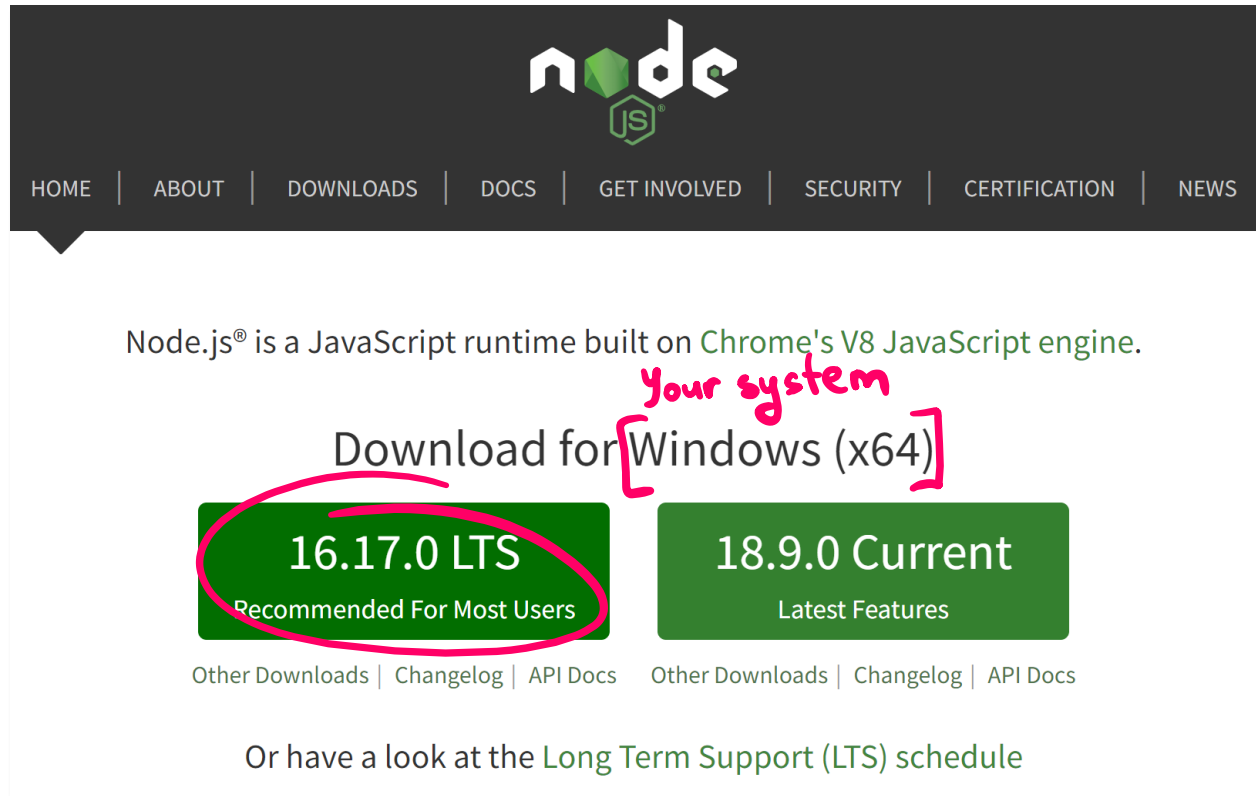
```
img {
  padding-right: 20px;
}
```

Alternatively, you can add `padding-right: 20px;` to the inline styles of the picture.

## Problem 4 (Install Node.js)

If you are using lab computers, you can skip the installation step, as Node.js is already installed on the lab computers. If you are using your own computer, go to <https://nodejs.org/en/> and download/install Node.js of the **latest stable version**:

## Tutorial 0 (self-study)



You can change the installation location if you want but be sure not to deselect any of the features on the 'custom setup' page of the installation wizard; the default options will install everything we need.

Once installed, you should be able to run the following command: **node -v** from the command line and see the version number printed out. If you did this successfully, Node.js is now installed on your computer. Node comes packaged with a second application called NPM. We will not need NPM for this tutorial, but you should check that it is installed correctly by running **npm -v** in the command line. If it prints out a version number, you're good to go.

The following screenshot is for Windows users. For other systems the idea is similar.

```
Windows PowerShell
PS C:\Users\Alina Shaikhet\Documents\2406> node -v
v16.17.0
PS C:\Users\Alina Shaikhet\Documents\2406> npm -v
8.15.0
PS C:\Users\Alina Shaikhet\Documents\2406>
```

You can run any JavaScript files you create for this tutorial by navigating to the file's directory from the command line and issuing the command:

**node YourFileName.js**

---

**Tutorial 0 (self-study)**

## Problem 5 (Write a Basic JavaScript Program)

Write a program that prints out the following pattern to the console (7 rows):

```
#  
##  
###  
####  
#####  
#####  
#####
```

Your program should contain no more than one **console.log()** statement (looped over multiple times) and you should be able to adjust the number of rows printed by changing one number within your code. Save your code in a file with the .js file extension and execute it using Node.js.

## Problem 6 (Save Your Work)

Keep your files organized. For example, place them into the “Tutorial-00” folder, where you can easily find them later for your reference. This tutorial is not for submission, but your work for the following 10 tutorials will be graded and should be submitted on Brightspace before deadline.