Over the course of this class, I have learned a lot about the ins and outs of HTML, CSS, and JavaScript. I always thought that HTML did all of the work in a website, when in reality, it pretty much does none of it. I would say that the internet as we know it today is because of CSS, visually anyways. Without CSS, one would essentially be looking at a large word document with some spacing and text sizes. On top of that, just in the last few years, so much capability has been added to the styling element of the internet. Most of that capability for the convenience of the developer. Then, once you add JavaScript to the equation, you are able to make live changes to a website and interact with the users’ machine for various tasks to make the experience even better.

With all that I have learned this semester, I believe that the topic of developer convenience would be a solid web development trend to cover. What I mean by developer convenience is all of the updates that have been added to HTML and CSS to ease the creation of a webpage. First, lets talk about code editing software. There are so many options out there for developers to choose from all with their own sets of features to assist a developer. In this class we were suggested to use VS Code and that is exactly what I did. Having been using Notepad++ before hand I would say for certain that this was a tremendous improvement after week 3. Having not started to use any of the bells and whistles that VS Code offers, just the fact that you are able to start typing a tag then hit tab and it fills in the rest. Or by using a period (.) or pound sign (#) when creating the tag to signify a class or ID name (ex: “div.class” turns into: <div class="class"></div> and “div#class” turns into: <div id="class"></div> by hitting tab). This alone was such a time saver and made writing HTML code so much more convenient. It was a week or two later I discovered the command, lorem. By using lorem and then adding a number after it (lorem20) you are able to add that many random words to any part of the HTML script within VS Code. This is extremely convenient when one is testing out new tags and how they work with the corresponding CSS code. There are so many more features that are part of VS Code that are simply there to make a developer’s life easier, if one is not using one of these code editors, they are simply wasting valuable time.

Even though code editors have been around for a very long time, I would not necessarily call it an emerging development trend. Yes, the features are very convenient, but they are not new; just new to me. A web development trend that is relatively new in the eyes of web development, however, would be the display: Grid element in CSS. Even though Grid was first made available in 2007, it was not really used until the late 2010’s. What this means is that anyone that started their web development journey in the last five years has been fortunate to be able avoid using the “old” way. In order to understand how convenient, the Grid layout is, you would have to know what the “old” way was. If one wanted to produce a website anywhere near the capabilities of what Grid allows, there would have to be endless nested div tags, countless classes, IDs, and style elements. This could very easily turn into tens of thousands of lines of code between HTML and CSS Grid could very easily cut in half. Well, what is “display: Grid;” and what does it allow you to do to an HTML page? CSS Grid layout, in short, turns any HTML page into, what I have been comparing it to, an Excel spreadsheet. It allows you to have as many rows and columns as you want that can all be laid out and styled uniquely. You are also able to merge any of the Grid’s cells together like one could in Excel. How does turning a webpage into an Excel-esque layout? Instead of having countless nested div tags, you have a few lines of CSS code, and then different class or ID names for the “cells” that are being worked with, and that is it. It would look something like: #grid {display: grid; grid-template-areas: ‘cell1 cell2’ ‘cell3 cell4’;}. This would turn the ID “grid” into Grid with 4 boxes. With other CSS elements you would be able to style this, easily, however you would like.

Even though Grid layouts still have not been widely adapted into production websites, this is the direction webpages are heading. The sheer capability and ease it provides to developers to layout a webpage will make it the gold standard of webpage development. Considering I did attempt to develop my own webpage once upon a time, long before I knew what Grid was, I am so excited to see that this is an option. I can only imagine what the next big update will be that makes display: Grid; look like hundreds of nested div tags.