## **HTML AND CSS BASICS**

### **Introduction**

HTML is the markup that contains all the actual stuff that a web page has. All the text on this page you’re reading right now lives inside HTML tags that tell your browser how to order the content on the page. Go on, right click any element on the page and choose “Inspect Element” to open up your browser’s Developer Tools and it will show you the structure of the page.

[CSS](https://skillcrush.com/2012/04/03/css/) tells the browser if you want to display any of those tags a particular way, for instance, turning its background blue and pushing it a little to the left. In your Developer Tools, you can see the CSS styles in another panel, usually showing which specific properties were inherited from which lines of CSS.

### **Learning Outcomes**

Look through these now and then use them to test yourself after doing the assignment:

* Why do we separate HTML and CSS?
* What are classes and IDs (and how are they different)?
* What are elements?
* What are tags?
* What are attributes?
* What are forms?
* What is a div?
* What are selectors?
* What are properties?
* What are values?
* What is the “query string” in a URL and what does it do?
* What is the difference between “pixels” and “ems”?
* How do CSS styles for a particular element get inherited? i.e. how does an element get its “default” styles?
* What are two CSS attributes you can change to push an element around on the page?
* What are the three different ways to include CSS in your project or use CSS to style a particular element?
* What is the “default stylesheet” or “user agent stylesheet”?
* Why use a CSS reset file?

### **Assignment**

1. Dive right into learning HTML and CSS with [freeCodeCamp’s interactive tutorials](https://www.freecodecamp.org/learn) to introduce you to these new concepts. Do the entire section labeled “Responsive Web Design Certification” only. It’s long and, if you’re new, might take you at least a few days to get through. This resource is meant to help you develop familiarity, so do not expect to memorize everything. Instead, expect to google these concepts in the future when you need to use them. They also have some projects at the end of this section. They’re good practice, but you can consider them optional. (A large part of their proposed ‘300 hours’ for completion comes from these projects, so it probably won’t take you that long if you only do the exercises).  
   Note: If you are using a browser add-on such as Dark Reader (or a similar add-on that creates a “dark mode” for a website), it may interfere with certain exercises. If your solution isn’t accepted, try disabling the browser add-on and submitting it again.
2. Optional: Learn about your browser’s default stylesheet and CSS resets [in this video](https://www.youtube.com/watch?v=14Vb6tZCjEY) (resets are mentioned starting at the 2:00 mark). This is why there are some spaces that show up in your layout even if you haven’t specified CSS. Real developers almost always use a CSS reset to blow away the default stylesheet and let them work from scratch. If you’re still curious, here’s a [longer video](https://www.youtube.com/watch?v=HqRFPLP7Ffs) on resets.

### **Additional Resources**

This section contains helpful links to other content. It isn’t required, so consider it supplemental if you need to dive deeper into something.

* A good page to bookmark for later reference is [Centering in CSS: A Complete Guide](https://css-tricks.com/centering-css-complete-guide/) from CSS Tricks.
* [W3Schools' CSS tutorial](https://www.w3schools.com/css/)
* [OverAPI’s CSS Cheat Sheet](https://overapi.com/css) and [OverAPI’s HTML Cheat Sheet](https://overapi.com/html)
* [CSSreference.io](https://cssreference.io/) and [HTMLreference.io](https://htmlreference.io/)
* [Learn CSS Layout](https://learnlayout.com/)
* [CSS Flexbox](https://flexbox.io/) and [CSS Grid](https://cssgrid.io/)
  + The [Complete Guide to Flexbox](https://css-tricks.com/snippets/css/a-guide-to-flexbox/) and the [Complete Guide to Grid](https://css-tricks.com/snippets/css/complete-guide-grid/) from CSS Tricks are both excellent references.
  + These [CSS FlexBox](https://youtu.be/FTlczfR82mQ) and [CSS Grid](https://youtu.be/EFafSYg-PkI) videos by [Dev Ed](https://www.youtube.com/channel/UClb90NQQcskPUGDIXsQEz5Q) might be helpful for you as well!
  + Hop on over to [Flexbox Froggy](https://flexboxfroggy.com/) or dig into [Grid Garden](https://cssgridgarden.com/) to learn some concepts with a fun game!

### **Knowledge Check**

This section contains questions for you to check your understanding of this lesson. If you’re having trouble answering the questions below on your own, review the material above to find the answer.

* What is the difference between HTML and CSS?
* For accessibility in HTML, what is the attribute used to describe an image (on screen readers or if it fails to load)?
* What is the difference between CSS Grid and Flexbox?
* For a responsive website, should it be designed mobile-first or desktop-first?
* Describe the components of the CSS Box Model.
* In CSS, what is a breakpoint?
* What is a div and how are they used?
* What are the two main groups of CSS properties that control typography style?
* What is the “query string” in a URL and what does it do?
* What is the difference between “pixels” and “em”?
* How does inheritance work for CSS styles, i.e. how does an element get its “default” styles?
* What are two CSS attributes you can change to push an element around on the page?
* What are the three different ways to include a CSS stylesheet in your project or use CSS to style a particular element?
* What is the “default stylesheet” or “user agent stylesheet”?
* What is the purpose of a CSS reset file?