**CSS Class and Id Attributes:**

* Class: Placed inside beginning tag of the element, the CSS then targets it
  + Ex. .shaded { background-color; light blue}
  + Elements can have more than one class
* Id: uniquely applies the name to the element; there is only one id name
  + Ex. #myheader
  + Id attributes allows scripts to only work with one specific element

**HTML File Basics:**

* Naming rules: save names in all lower case, first character should be a letter, no spaces in the filename, don’t start file name with a number
* <meta> tags describe properties about the page.
  + <meta charset= “UTF-8”>: tells that it uses all languages
  + <meta name= “viewport” content= “width=device-width, initial-scale=1.0”>
    - This sets the width of the site to the width of the device
* <div> tag is non-semantic. It’s just a way to format text that doesn’t have an exact meaning
  + Screen readers use the tag, so it’s best to describe the tag as accurately as possible
* Validating makes sure that the code is correctly written

**Units of Measurement:**

* No space between number and the unit
* Absolute units: the same size no matter the page size
  + Px
* Relative units: changes based on page or other factors
  + Em, %
  + Ex: width: 100%;
  + Scale better; make pages more responsive
  + 1em = 100% of default font size

**Block vs. Inline Elements:**

* Block elements, take up 100% of the space that they have, don’t allow other elements to share the space that they are on
* Inline elements: allows other elements to share the space they have, as long as there is room
  + Display: inline; display: block
* Inline-block
  + Causes a block element to flow like an inline element
  + Display: inline-block
* Text-align: center; Content is centered in the invisible box
* Margin: 0 auto; centers the box itself

**CSS Selectors:**

* Element Name Selectors
* ID & Class Selectors
* Specificity
  + Rank that determines which element will be applied
  + More specific takes priority
  + Id tags, then class tags, then element selectors
* Img.heat
  + Using elements with a class of heat
  + Targets img elements that have the class of heat
* H1, h4 {} (all h1 and h4 elements with have the following properties)
* Pseudo-selectors
  + Li:nth-of-type(2) selector (selects just the second bullet in a list)
* Pseudo-classes
  + Allows the change of an element when a user hovers over it

**The Box Model:**

* Content, Padding, Border, Margin
* Box-sizing: border-box;
  + Same length as the borders in the parameter

**Backend Developers:**

* Developers who make the browsers correspond to the correct data
* They need to understand backend coding languages
  + JavaScript, Python
* Work with the frontend developers
  + User data: storing data, and ensuring it’s only presented to users who are supposed to access it
  + Payment processing: accepting data, securely storing that info, and making charges to that payment
  + APIs (Application Programming Interfaces): resources meant to be accessed not by web browsers, but by mobile phone apps and other programs
* Skills needed:
  + Apache, NGINX, Microsoft IIS (Internet Information Services)
  + Web Frameworks: libraries for programming languages that help you, the developer, handle web requests
  + JavaScript
    - Node.js
    - Express
  + Java (not the same as JavaScript)
    - Spring
    - JSF
  + Python
    - Django
    - Flask
  + C#
    - ASP.NET MVC/ CORE
* SQL (Structured Query Language)
* MySQL
* SQL Server
* PostgreSQL
* Version Control Software
  + Git
  + Subversion
* Cloud Hosting
  + Amazon Web Services
  + Heroku
  + Google Cloud Platform