**Module 1:**

* HTML is short for Hyper Text Markup Language
* The standards for HTML, CSS, JavaScript, and many other web technologies are created by the World Wide Web Consortium (W3C)
* Headings are one of the first elements we were introduced to
  + Heading elements are numbered <h1> – <h6>
  + Lower numbers are used for the most important heading information on a webpage, while the higher numbers are used for less important heading info
  + The most important headings are the largest, while the least important headings are the smallest
* We can do some basic formatting using the line break element <br>, which will insert a new line into a page
  + By default, HTML will collapse all extra spaces and extra lines in a document down to a single space, so <br> is one of the only ways to do formatting
* Most elements have an *opening tag* and a *closing tag*, with some kind of content in between them.
  + The *opening tag* is the tag you use to start writing an element, while the *closing* *tag* is the tag you use to end an element
  + The tags themselves do not appear on your webpage, but anything between the opening and closing tags will does appear
* Some elements, like a line break element, do not have any content or a closing tag. These are called *empty elements*
* The basic structure of an HTML element is like this:
  + Doctype declaration (info for the browser about the document)
  + <html> element
  + <head> element
  + <body> element
* Although HTML is *not* case-sensitive, you should always use lowercase inside of your tags, but regular English syntax/grammar between the opening and closing tags

**Module 2:**

* HTML is pretty ugly, so we’ll have to write some *CSS Styles* to make it look presentable
* There are 3 ways to write CSS for your HTML code
  1. Inline Styles: Styles created using the *style* *attribute* inside of the opening tag of an element
  2. Internal Styles: Styles created using the *<style> element* inside of the *<head> element* at the top of the page
  3. External Styles: Styles created in an external CSS file that you can attach to an HTML page using a *<link> element*
* CSS gives us many options to make our HTML code look better, such as changing the font color, the background color, the padding, margins, etc
* CSS has many different ways to express color, including *named colors, RGB values, RGBA values, HEX values, HSL values,* and *HSLA values*
  1. Named Colors: There are 140 predefined colors (like red or blue) that can be used. These colors are not very specific, but are easy to use
  2. RGB Colors: RGB values include a value for Red, a value for Green, and a value for Blue. The R, G, and B values are numbers between 0 and 255 and represent how much of that color is included in the RGB color. An RGB color looks like this: rgb(255, 255, 255).
  3. RGBA Colors: Exactly like RGB colors, but include an *alpha* value. The *alpha* value is the opacity of the color and can be used to make the color’s background show through it. The Alpha value is a number between 0 and 1. An RGBA color looks like this: rgba(255, 255, 255, .5)
  4. HEX Colors: Works like an rgb value, but instead of representing the R, G, and B values as regular base-10 numbers, it represents them as Hexadecimal (base-16) numbers. Hex numbers run from 0 – F. Hex colors can represent every color an RGB can. A Hex color looks like this: #FFFFFF.
  5. HSL Colors: Represent colors as a *Hue*, a *Saturation*, and a *Lightness*. The *Hue* is a number between 0 and 360 that represents a tint of red, green, or blue. Red values are close to 0/360, Green values are close to 120, and Blue values are close to 240. The saturation is a value between 0 and 100 that represents how intense the color is. The lightness is a value between 0 and 100 representing how dark/light a color is.
  6. HSLA Colors: Exactly like an HSLA, but adds an *alpha value*, which determines the colors opacity.

**Module 3:**

* There were many new elements this Module, including:
  + <hr> - The horizontal rule. This element can be used to create a thematic break between sections of a webpage by drawing a black horizontal black line between them. <hr> is an empty element
  + <strong> - Used for important text. Similar to <b>, but has semantic meaning indicating the text’s importance
  + <em> - Used to emphasize text. Similar to <i>, but will give it a special emphasis when spoken by a screen reader.
  + <mark> - Used to highlight a section of text.
  + <img> - Used to create an image. To create an image, you need the *src* attribute (to create a filepath), an *alt* attribute (to explain to the image to screen readers), a width, and a height.
  + <map> - Used to create an image map. Image maps are areas of an image that work as a link, but without turning the entire image into a link.
  + <area> - Used to along with a <map> element to define the clickable areas of an image map. There are different shapes used to create the clickable areas: rect, circle, poly (polygon).
  + <a> - An anchor element (a hyperlink). We can use anchors to link our users to external pages, internal pages, or even to other places within our current HTML page. Any visible element can be wrapped in an anchor to turn that element into a link. Anchors have two important attributes: *href* (hypertext reference) – the link to the place you would like to send your users, and *target –* one of four values used to determine where the new page will open. \_blank opens it in a new tab.
  + <table> - Used to create a table.
  + <tr> - Used to create a table row
  + <th> - Used to create a table heading
  + <td> - Used to create a table cell.
  + <ol> - Used to create an ordered list. Ordered lists are lists that use numbers (by default). These list have an order to their list items. You can change the character used to order them from a number to something else using the *type* attribute with a value of either I, i A, or a, for an uppercase roman numeral, lowercase roman numeral, uppercase A, or lowercase A, respectively.
  + <ul> - Used to create an unordered list. Unordered lists do not have any order to them. These lists are marked by bullet points, rather than numbers.
  + <li> - Used to create a list item. List items are the bullets/numbers in a list.
  + <dl> - Used to create a description list. Description lists are used for giving a list of terms along with their descriptions.
  + <dt> - Used to create a description term. Description terms are the terms you would like to define in a description list. These are similar to <li> elements.
  + <dd> - Used to create a description of the description term. These appear indented.
  + <bdo> - Used to create a *bi-directional override*. A <bdo> element can be used to reverse the direction of text. This requires using the *dir* attribute set to “rtl”.
  + <iframe> - Used to create an inline frame. Iframes can display entire documents inside of them and are often used to display other HTML docs or videos.
* New Attributes
  + *Title* – Used to create a tooltip that will show you its value when you hover your mouse over an element
* Comments are used to explain what is happening in a piece of code. They are especially useful for debugging or explaining our code to other programmers. Nothing inside of the comment will be read by HTML or rendered on your webpage. You can create comments using this: <!-- Your Comment Here -->
* Block Elements – Elements that take up the entire line and always begin on a new line. Ex - <p>, <div> <h1>
* Inline Elements – Elements that only take up as much width as they need and do not begin on a new line. Ex - <span>, <i>, <q>, <strong>

**Module 4:**

* All visible elements can have an ***id*** and ***class*** attribute.
* IDs can be used to refer to a single element for the purpose of CSS styling or JavaScript selection. Every ID must be unique
* Classes are shared by multiple elements. You can use classes to style a set of elements all at once.
* You can create a “bookmark link” by using an anchor with the href attribute set to the id of the element you want to link to.
* Bookmark links can link to other pages as well by including the id of the element you want to link to on the next page.

**Module 5:**

* New elements:
  + <title> - Used to create a title for the webpage. The title appear on the tab of the page
  + <link> - Used to link an external file to an HTML page. This is how you create an external style. Additionally, this is used to attach icons to the tab.
  + <article> - A semantic element used to represent independent, self-contained content, like a news article or a blog
  + <aside> - A semantic element used to contain side comments or often advertisements.
  + <details> - A semantic element used to define additional details that a user can view or hide
  + <figcaption> - A semantic element used along with <figure> to provide a caption for an image
  + <figure> - A semantic element used to add a caption to an image. This element works similarly to a <picture> or <video> element.
  + <footer> - A semantic element intended for the bottom of the page that contains copyright info, contact info, legal info, or other such info
  + <header> - A semantic element used to create a header that often contains a heading, navigation links, or other info
  + <main> - A semantic element used to contain the main content of a webpage
  + <nav> - A semantic element used to create a navbar
  + <section> - A semantic element used to define a “thematic grouping of content, typically with a heading”
  + <summary> - A semantic element used to create a visible heading for a <details> element
  + <time> - A semantic element used to define a date/time

**Module 6:**

* Character entities are used to create characters that would otherwise be reserved by HTML, like < or >. They can also be used to insert additional spaces into elements to prevent HTML from collapsing multiple spaces or lines into a single space. Character entities can be created using their entity name or their entity code. There are nearly 150,000 character entities, including foreign alphabets and emojis.
* Useful Character Entities:
  + Non-breaking Space - &nbsp; or &#160;
  + Less Than - &lt; or &#60;
  + Greater Than - &gt; or &#62;
  + Double Quote - &quot; or &#34;
  + Single Quote - &apos; or &#39;
* There are several ways to create output with HTML. One of them we have not used very often is document.write(), which can be used to change the output of an HTML doc to whatever argument was given to document.write().

**Module 7:**

* Forms have require inputs to function correctly. There are many types of inputs:
  + <input type=”button”> - Creates a button for use in a form (requires JS code to work correctly)
  + <input type=”checkbox”> - Creates a checkbox for use in a form
  + <input type=”color”> - Creates a color picker for use in a form
  + <input type=”date”> - Creates a date picker for use in a form
  + <input type=”datetime-local”> - Creates an input to pick a date and a time for use in a form
  + <input type=”email”> - Creates a text input with built in validation for entering an email
  + <input type=”file”> - Creates an input to upload a file for use in a form
  + <input type=”hidden”> - Creates an hidden input field that will be submitted with the rest of the form, but is not visible. This can be used to submit information without it being seen by the user.
  + <input type=”image”> - Creates a submit button using an image rather than a regular button
  + <input type=”month”> - Creates a month/year picker for use in a form
  + <input type=”number”> - Creates a text input that accepts a number.
  + <input type=”password”> - Creates a text input that accepts a password. The text that you entered will appear as dots
  + <input type=”radio”> - Creates a radio button input. You can only select a single radio button at a time.
  + <input type=”range”> - Creates a slider input. Great for selecting an imprecise value, but not for specific values.
  + <input type=”reset”> - Creates a reset button for a form. Resets the values of all inputs in the form.
  + <input type=”search”> - Creates a text input optimized for searches. Creates a different keyboard when used on a mobile device.
  + <input type=”submit”> - Creates a submit button. Used to submit an entire form.
  + <input type=”tel”> - Creates an input for a phone number. Supports the use of *regular expressions*, which are used to validate that a phone number was input.
  + <input type=”text”> - Creates a text input for a form.
  + <input type=”time”> - Creates an input that accepts a time.
  + <input type=”url”> - Creates an input that accepts a URL.
  + <input type=”week”> - Creates a week picker. Not supported on all browsers.
* The <fieldset> element can be used to group multiple form elements together.
* The <legend> element can be used to create a caption for a <fieldset>
* HTTP: Hyper Text Transfer Protocol is the protocol used to transfer most information over the internet. There are four main HTTP Methods, but we care the most about *GET* and *POST*. Get is designed to send receive information but can be used to send a small amount of unencrypted information. Post can be used to send any amount of information and all information is secure.

**Module 8:**

* The <canvas> element is used to draw graphics on an HTML page using JavaScript using the built-in getContent object
* <canvas> is able to draw lines with the moveTo(x,y) and lineTo(x,y) methods
* We can embed *SVGs (Scalable Vector Graphics)* into our HTML page using the <svg> element
* SVGs are well suited for displaying an image across any screen size, but is not well suited for game development
* SVGs are based on XML
* SVGs will never stretch or distort no matter how large or small they are displayed
* The <source> element can be used to define media resources for media elements, like for <video> and <audio> elements

**Module 9:**

* Browsers have access to Geolocation data of the user. They can access the geolocation data using the **getPosition()** method
* Web Workers allow JavaScript to run in the background, without affecting the performance of the website or the User Experience
* You can make elements draggable using the *draggable* attribute
* The sessionStorage object can be used to store user info for a single session
* The localStorage object can be used to store data for as long as you would like to store itl