Day 1:

- HTTP vs HTTPS
    - HTTPS is HTTP with encryption. The difference between the two protocols is that HTTPS uses TLS (SSL) to encrypt normal HTTP requests and responses. As a result, HTTPS is far more secure than HTTP.
- Restful API (**RE**presentational **S**tate **T**ransfer)
    - uses HTTP requests to access and use data
    - **GET** – retrieve resource
    - **POST** – create resource
    - **PUT** – change/update resource
    - **HEAD** - method returns info about resource (version/length/type)
    - **DELETE** – remove resource
    - **PATCH** – change/update resource
    - **OPTIONS** - method returns info about API (methods/content type)
- HTTP Response Codes
    - Informational Responses (100 – 199)
    - Successful Responses (200 – 299)
    - Redirection Messages (300 – 399)
    - Client Error Responses (400 – 499)
    - Server Error Responses (500 – 599)
- Local Storage vs Session vs Cookie
    - Local storage – Stores data on client's computer. Saves keys and stores data
    - Session storage – Data stored only for a session until the browser or tab is closed
    - Cookies – Server sided. Stores data that has to be sent back to server with requests
- HTML defer vs async
    - Async – will wait for the element inside JS element
    - Defer – waits for JS code and runs at the end
    - The script with async attribute will be executed once it is downloaded. While the script with defer attribute will be executed after completing the DOM parsing
    - The scripts loaded with async doesn't guarantee any order. While the scripts loaded with defer attribute maintains the order in which they appear on the DOM.

- Search Engine Optimization (SEO)
    - SEO stands for "search engine optimization." In simple terms, it means the process of improving your site to increase its visibility
    - Meta data tag in HTML helps with result findings in google
        - The <meta> tag defines metadata about an HTML document. Metadata is data (information) about data.
- What is iframe?
    - An iFrame (Inline Frame) is an HTML document embedded inside another HTML document on a website
- What is character encoding?
    - A character encoding tells the computer how to interpret raw zeroes and ones into real characters. It usually does this by pairing numbers with characters
- What are semantic elements?
    - In HTML there are some semantic elements that can be used to define different parts of a web page:
- What does DOCTYPE do?
    - Doctype stands for Document Type Declaration. It informs the web browser about the type and version of HTML used in building the web document. This helps the browser to handle and load it properly.
- What's new in HTML5?
    - Video and audio tags
    - Storage
    - Web browser support
    - Header and footer elements
- Canvas vs SVG
    - The HTML <svg> element is a container for SVG graphics. SVG stands for Scalable Vector Graphics. SVG and useful for defining graphics such as boxes, circles, text, etc. SVG stands for Scalable Vector Graphics and is a language for describing 2D-graphics and graphical applications in XML and the XML is then rendered by an SVG viewer. Most of the web browsers can display SVG just like they can display PNG, GIF, and JPG.
    - The HTML <canvas> element is used to draw graphics, via JavaScript. The<canvas> element is a container for graphics.
- Can a web page contain multiple <header> /<footer> elements?
    - Yes, but with a catch. The W3 documents state that the tags represent the header and footer areas of their nearest ancestor section. I would

recommend having as many as your want, but only 1 of each for each "section" of your page, i.e. body, section etc.

- Why is it generally a good idea to position CSS <link>s between <head></head> and JS <script>s just before </body>?
    - Placing <link>s in the <head>— Putting <link>s in the head is part of the specification. Besides that, placing at the top allows the page to render progressively which improves user experience. The problem with putting stylesheets near the bottom of the document is that it prohibits progressive rendering in many browsers, including Internet Explorer. Some browsers block rendering to avoid having to repaint elements of the page if their styles change. The user is stuck viewing a blank white page. It prevents the flash of unstyled contents.
    - Placing <scripts>s just before </body> — <script>s block HTML parsing while they are being downloaded and executed. Downloading the scripts at the bottom will allow the HTML to be parsed and displayed to the user first.
- What is DOM?
    - Document Object Model
    - Tree structure wherein each node is an object representing a part of the document
- HTML5 vs XHTML
    - HTML5 is compatible with all browsers. XHTML is not.
    - HTML5 is compatible with all browsers. XHTML is not.
    - XHTML need closing tags and not all HTML5 elements need closing tags
- Where should we put <link> and <script> and why
    - In the <head> tag
    - Alerts the browser to start interpreting all the text between tags as a script
- 508 policy / ARIA (Accessible Rich Internet Applications)
    - set of attributes that define ways to make web content and web applications (especially those developed with JavaScript) more accessible to people with disabilities.

Day 2:

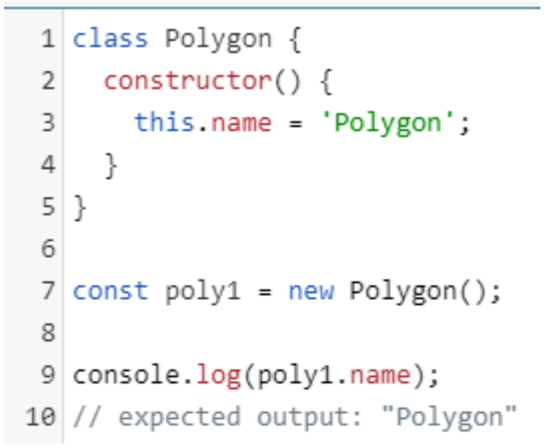
- Regular expression (REGEX)
    - CSS selectors cheat sheet, JS cheat sheet
        - https://frontend30.com/css-selectors-cheatsheet/
- What is CSS selector? Name some.
    - In CSS, selectors are used to target the HTML elements on our web pages that we want to style.
    - Type, class, Id selectors

    - Attribute selectors such as
        - a:hover{}
- Positions:
    - Static
        - Static positioned elements are not affected by the top, bottom, left, and right properties.
        - An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of the page:
    - Relative
        - An element with position: relative; is positioned relative to its normal position.
        - Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.
    - Absolute
        - An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).
        - However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.
    - Sticky
        - An element with position: sticky; is positioned based on the user's scroll position.
    - Fixed
        - An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.
        - A fixed element does not leave a gap in the page where it would normally have been located.
- Explain the three main ways to apply CSS styles to a web page
    - linked, embedded, and inline

- How would you approach fixing browser-specific styling issues?
    - After identifying the issue and the offending browser, use a separate style sheet that only loads when that specific browser is being used. This technique requires server-side rendering though.

    - Use libraries like Bootstrap that already handles these styling issues for you.
- Describe pseudo-elements and discuss what they are used for
    - A CSS pseudo-element is a keyword added to a selector that lets you style a specific part of the selected element(s)

Day 3:

- JS vs ECMAScript vs node
    - **ECMAScript** – syntax standard
    - **JavaScript** – ECMAScript + web API
    - **Node** – ECMAScript + node API
- Primitive type data is pass by value (making a copy)
    - String, number, Boolean, undefined
- Non-primitive type data is pass by reference (pointing to the address)
    - Objects, function
- Deep copy vs shallow copy
    - Shallow copy – Makes a copy and points to the same memory address
    - Deep copy – Makes a copy with a different memory address
- Class key in ES6
    - The Object.keys() method returns an array of a given object's own enumerable property names
- Constructor function in ES5
    - The constructor method is a special method of a class for creating and initializing an object instance of that class.
    - 

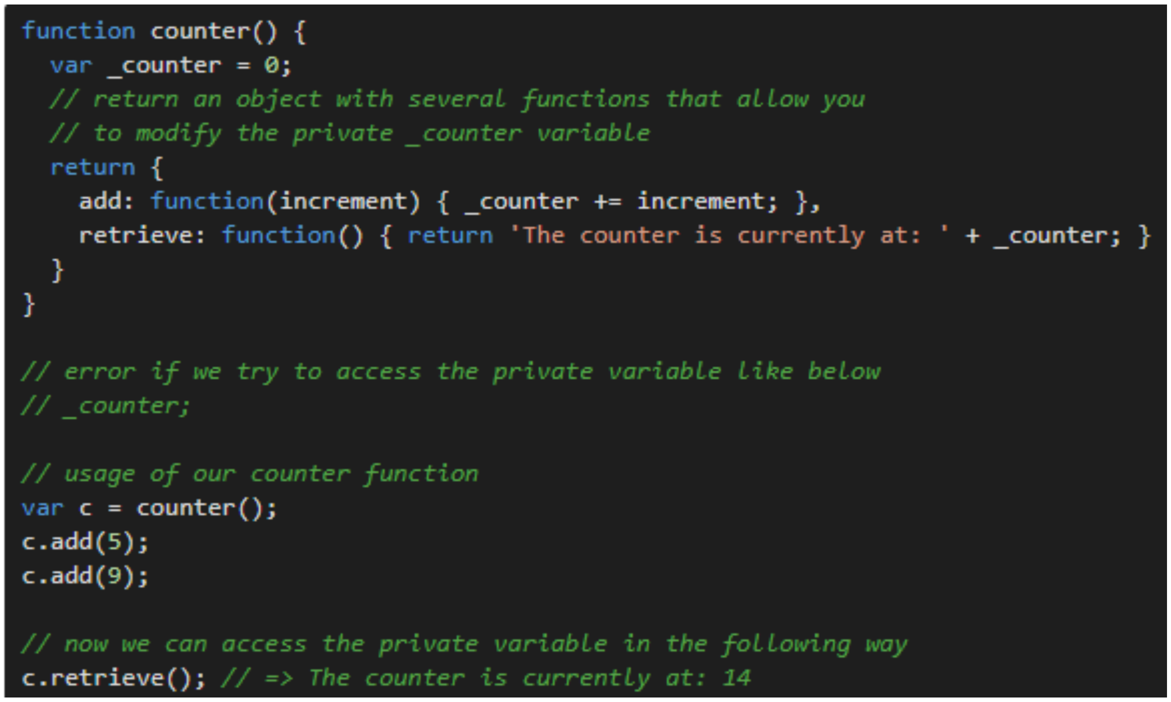
```
class Polygon {
  constructor() {
    this.name = 'Polygon';
  }
}

const poly1 = new Polygon();

console.log(poly1.name);
// expected output: "Polygon"
```

- JS coercion

- Type Coercion refers to the process of automatic or implicit conversion of values from one data type to another. This includes conversion from Number to String, String to Number, Boolean to Number etc. when different types of operators are applied to the values.

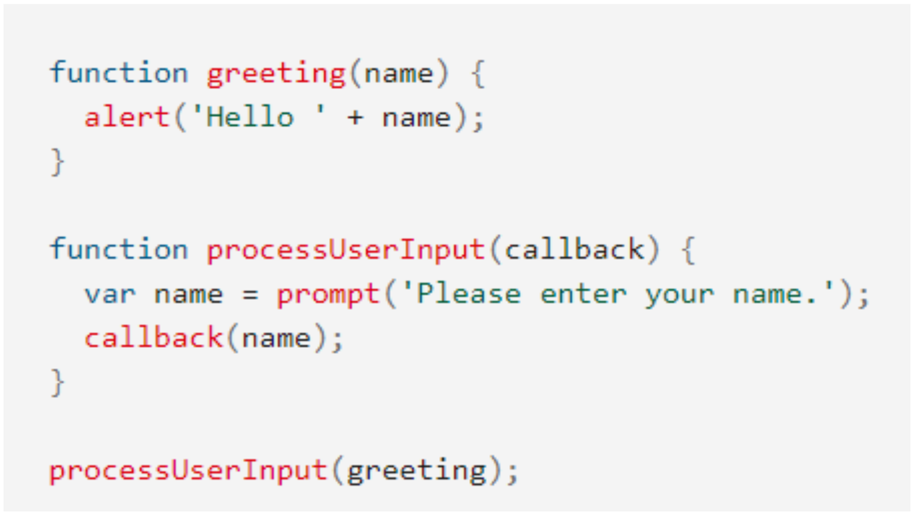
- JS Scope
    - The scope is an important concept that manages the availability of variables
    - Block (let/const)
    - Function (local scope)
    - Global (variables outside functions)
- JS equality
    - "==" compares value
    - "===" compares value and type
- Null vs undefined
    - Undefined – variable is declared but not given a value; Type undefined
    - Null is an assignment value meaning no value; Type object
- What is strict mode in JS?
    - With strict mode, you cannot, for example, use undeclared variables.
- What is polyfill?
    - A polyfill is a browser fallback, made in JavaScript, that allows functionality you expect to work in modern browsers to work in older browsers, e.g., to support canvas (an HTML5 feature) in older browsers.
- Var/let/const
    - Var – pre ES6 for declaring variables
    - Const/let – ES6 syntax
        - Let – able to reassign variable value
        - Const – variable value will not be reassigned
- Explain event bubbling and how one may prevent it.
    - Event Bubbling is the event starts from the deepest element or target element to its parents, then all its ancestors which are on the way to bottom to top. At present, all the modern browsers have event bubbling as the default way of event flow.
    - If you want to stop the event bubbling, this can be achieved by the use of the event.stopPropagation() method. If you want to stop the event flow from event target to

top element in DOM, event.stopPropagation() method stops the event to travel to the bottom to top.

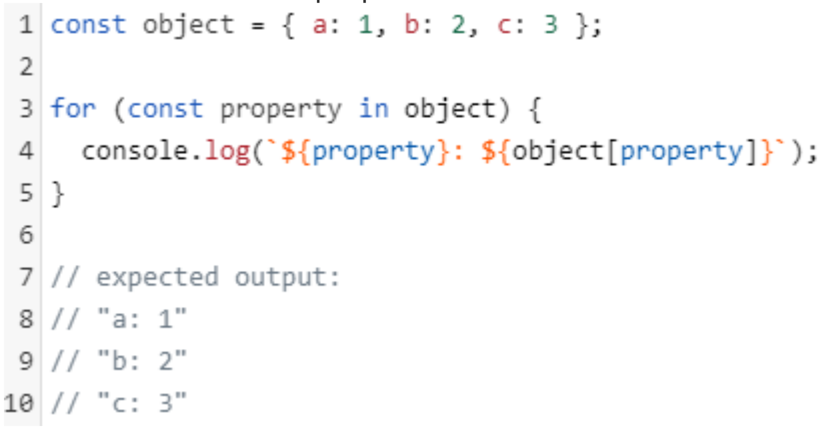
- How to empty an array in JavaScript
  - Assign it to an empty array
  - Setting the length of the array to zero
  - Remove each element in the array with pop() method
- How to check if an object is an array
  - You can use typeof (console.log(typeof arr))
  - Or you can use Array.isArray(arr)
- How would you use a closure to create a private counter?
  - How would you use a closure to create a private counter
  - You can create a function within an outer function (a closure) that allows you to update a private variable but the variable wouldn't be accessible from outside the function without the use of a helper function.
  - 

```
function counter() {
  var _counter = 0;
  // return an object with several functions that allow you
  // to modify the private _counter variable
  return {
    add: function(increment) { _counter += increment; },
    retrieve: function() { return 'The counter is currently at: ' + _counter; }
  }
}

// error if we try to access the private variable like below
// _counter;

// usage of our counter function
var c = counter();
c.add(5);
c.add(9);

// now we can access the private variable in the following way
c.retrieve(); // => The counter is currently at: 14
```

- Call back function example:
  - A callback function is a function passed into another function as an argument, which is then invoked inside the outer function to complete some kind of routine or action.

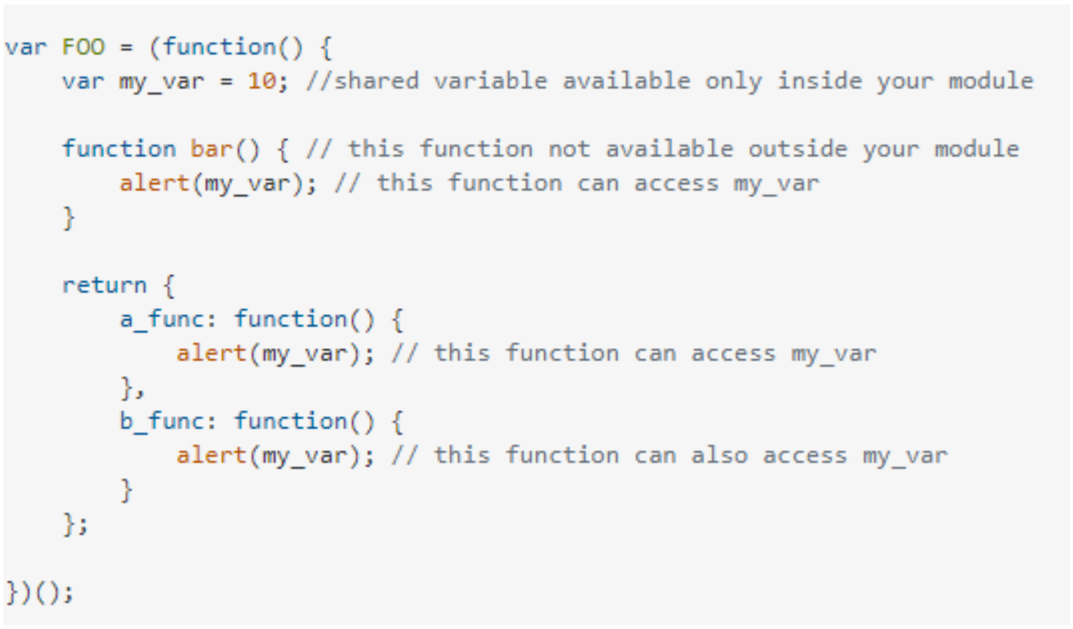
- 

```
function greeting(name) {
  alert('Hello ' + name);
}

function processUserInput(callback) {
  var name = prompt('Please enter your name.');
  callback(name);
}

processUserInput(greeting);
```

- How to iterating over object properties and array items?
    - The for...in statement iterates over all enumerable properties of an object that are keyed by strings (ignoring ones keyed by Symbols), including inherited enumerable properties.
    - 

```
const object = { a: 1, b: 2, c: 3 };

for (const property in object) {
  console.log(`${property}: ${object[property]}`);
}

// expected output:
// "a: 1"
// "b: 2"
// "c: 3"
```

- Why we need to avoid touching global scope and how to avoid it?
    - The primary reason why global variables are discouraged in javascript is because, in javascript all code share a single global namespace, also javascript has implied global variables ie. variables which are not explicitly declared in local scope are automatically added to global namespace. Relying too much on global variables can result in collisions between various scripts on the same page
    - One way to reduce global variables is to use the YUI module pattern. The basic idea is to wrap all your code in a function that returns an object which contains functions that needs to be accessed outside your module and assign the return value to a single global variable.



```
var FOO = (function() {
    var my_var = 10; //shared variable available only inside your module

    function bar() { // this function not available outside your module
        alert(my_var); // this function can access my_var
    }

    return {
        a_func: function() {
            alert(my_var); // this function can access my_var
        },
        b_func: function() {
            alert(my_var); // this function can also access my_var
        }
    };

})();
```

- DOM event DOMContentLoaded
    - The DOMContentLoaded event fires when the initial HTML document has been completely loaded and parsed, without waiting for stylesheets, images, and subframes to finish loading.
- Undefined vs not defined
    - If the variable name which is being accessed doesn't exist in memory space then it would be **not defined**
    - if exists in memory space but hasn't been assigned any value till now, then it would be **undefined**.
- What is currrying in js?
    - currying is the technique of converting a function that takes multiple arguments into a sequence of functions that each takes a single argument
- What is high order function?
    - In Javascript, functions can be assigned to variables in the same way that strings or arrays can. They can be passed into other functions as parameters or returned from them as well.
    - A "higher-order function" is a function that accepts functions as parameters and/or returns a function.
- What is hoisting?
    - Hoisting is JavaScript's default behavior of moving declarations to the top.
- Rest operator vs spread operator

- Rest- The rest operator is used to put the rest of some specific user-supplied values into a JavaScript array.

```
// Define a function with two regular parameters and one rest parameter:
function myBio(firstName, lastName, ...otherInfo) {
  return otherInfo;
}

// Invoke myBio function while passing five arguments to its parameters:
myBio("Oluwatobi", "Sofela", "CodeSweetly", "Web Developer", "Male");

// The invocation above will return:
["CodeSweetly", "Web Developer", "Male"]
```

- Spread - The spread operator (...) helps you expand iterables into individual elements.

```
const myName = ["Sofela", "is", "my"];
const aboutMe = ["Oluwatobi", ...myName, "name."];

console.log(aboutMe);

// The invocation above will return:
[ "Oluwatobi", "Sofela", "is", "my", "name." ]
```