Language Map for JavaScript

Variable Declaration *Is this language strongly typed or dynamically typed? Provide at least three examples (with different data types or keywords) of how variables are declared in this language.*	JavaScript is a dynamically typed, meaning that variable types are determined at runtime and can change as the program executes. Explicitly: let age = 30; // Number Implicitly: var name = "Alice"; // String Constant: const isStudent = true; // Boolean
Data Types *List all of the data types (and ranges) supported by this language.*	**Primitive Data Types** Null – intentionally absent of any object value Undefined – a variable that has been declared but not yet assigned value Boolean – local entities that can have one of two values; true or false. Number – represents both integer and floating-point numbers. • Range: From approximately $$-2^{53} + 1$$ to $$2^{53} - 1$$ String – textual data. Symbol – represents a uniqie and immutable identifier. BigInt – represents integers with arbitrary precision. **Complex Data Type** Object – collection of properties, where each property is defined as a key-value pair.
Selection Structures *Provide examples of all selection structures supported by this language (if, if else, etc.)* ***Don't just list them, show code samples of how each would look in a real program.***	**if Statement** `let age = 21;` `if (age >= 21) {` `console.log("You can purchase this.");` `}` **if else Statement** `let age = 18;` `if (age >= 21) {` `console.log("You can purchase this.");` `} else {` `console.log("You cannot purchase this.");` `}`

	If else if else Statement ```if (score >= 90) { console.log("Grade: A"); } else if (score >= 80) { console.log("Grade: B"); } else if (score >= 70) { console.log("Grade: C"); } else { console.log("Grade: F"); }``` **Switch Statement** ```switch (day) { case 1: dayName = "Monday"; break; case 2: dayName = "Tuesday"; break;``` **Ternary Operator** ```let canVote = (age >= 18) ? "Yes" : "No"; console.log(canVote); // Output: Yes```
Repetition Structures *Provide examples of all repetition structures supported by this language (loops, etc.)* ***Don't just list them, show code samples of how each would look in a real program.***	**For Loop** ```for (let i = 0; i < 5; i++) { console.log("Iteration number: " + i); }``` **While Loop** ```while (count < 5) { console.log("Count is: " + count); count++; }``` **Do while Loop** ```do { console.log("Count is: " + count); count++; } while (count < 5);```

	For in Loop let person = {firstName: "John", lastName: "Doe", age: 25}; for (let key in person) { console.log(key + ": " + person[key]); } **For of Loop** let fruits = ["Apple", "Banana", "Cherry"]; for (let fruit of fruits) { console.log(fruit); }
Arrays *If this language supports arrays, provide at least two examples of creating an array with a primitive or String data types (e.g. float, int, String, etc.)*	**Array of Numbers:** let numbers = [1, 2, 3, 4, 5]; console.log(numbers); // Output: [1, 2, 3, 4, 5] **Array of Strings** let fruits = ["Apple", "Banana", "Cherry"]; console.log(fruits); // Output: ["Apple", "Banana", "Cherry"]
Data Structures *If this language provides a standard set of data structures, provide a list of the data structures and their Big-Oh complexity.*	**Array :** O(n) **Queue:** O(n) **Stack:** O(n) **Linked List:** O(n) **Doubly Linked List:** O(n) **Hash Table:** O(1) **Binary Search Trees:** O(log n) **Skip List:** O(log n)

Objects *If this language support object-orientation, provide an example of how you would write a simple object with a default constructor and then how you would instantiate it.*	//default constructor class Person { constructor() { this.name = "John Doe"; this.age = 21; } // Method to display person details displayInfo() { console.log(`Name: ${this.name}, Age: ${this.age}`); } } // Instantiate let person1 = new Person(); // Display person1.displayInfo(); // Output: Name: John Doe, Age: 30
Runtime Environment *What runtime environment does this language compile to? For example, Java compiles to the Java Virtual Machine.* *Do other languages also compile to this runtime?*	JavaScript typically runs in a JavaScript runtime environment, which is provided by JavaScript engines. The most well-known JavaScript engine is Google's V8 engine, which powers Google Chrome and Node.js JavaScript engines use a combination of interpretation and Just-In-Time (JIT) compilation to execute code. This means that JavaScript code is parsed, compiled to machine code, and executed on the fly. Other languages that compile to this runtime: TypeScript, CoffeeScript, Elm, Dart
Libraries/Frameworks *What are the popular libraries or frameworks used by programmers for this language? List at least three (3) and describe what they are used for..*	**React** is a JavaScript library developed by Facebook for building user interfaces, particularly single-page applications. It allows developers to create reusable UI components and manage the state of their applications efficiently. React uses a virtual DOM to optimize updates and rendering, making it highly performant for dynamic and interactive web applications **Angular** is a comprehensive framework developed by Google for building web applications. It provides a robust set of tools and features for developing large-scale applications, including two-way data binding, dependency injection, and a powerful CLI (Command Line Interface). Angular is well-suited for enterprise-level applications and offers a complete solution for both front-end and back-end development

	Vue.js is a progressive JavaScript framework for building user interfaces. It is designed to be incrementally adoptable, meaning you can use as much or as little of Vue as you need. Vue is known for its simplicity and flexibility, making it a great choice for both small and large projects. It offers features like reactive data binding and a component-based architecture, similar to React.
Domains *What industries or domains use this programming language? Provide specific examples of companies that use this language and what they use it for**. E.g. Company X uses C# for its line of business applications.***	**Google:** Uses JavaScript extensively in its advertising platforms and analytics tools. JavaScript enables the creation of interactive ads and the tracking of user activities to improve marketing strategies. **Facebook:** Uses JavaScript, particularly the React library, to build and manage user interface components for its web applications. React helps in creating dynamic and responsive user experience **Amazon:** Utilizes JavaScript for both front-end and back-end development. JavaScript frameworks like Node.js are used to handle server-side operations, while front-end libraries like React and Angular enhance the user interface. **PayPal:** Employs JavaScript and Node.js to build scalable network applications and handle real-time transactions. JavaScript helps in creating secure and efficient payment processing systems