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 dynamically typed. Examples of declared variables: let name = "Adam"; (use "let" for string) let age = 40; (use "let" for number) let allowed = false (use "let" for Booleans) let myArray = [2,4,6] (use "let" for arrays)
Data Types *List all of the data types (and ranges) supported by this language.*	Data types: String – any alphanumerics Number – any integer or floating point up to ($2^{53} - 1$); no commas Boolean – true/false BigInt – integers larger than ($2^{53} - 1$) Null – absence of any object value Undefined – declared but value not assigned Object – used to store collections of data and more complex entities Symbol – used to create unique identifiers for objects

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 capacity = 1000;
	if (capacity < 1000){
	  console.log("The jug is too full.");
	
	if-else statement
	let price = 100;
	if (wallet >= 100){
	  console.log("You can afford it.");
	} else{
	  console.log("Keep saving.");
	
	if-else, if-else statement
	let passingScore = 80;
	if (grade > 95){
	  console.log("You aced it.");
	} else if (grade >80){
	  console.log("You passed.");
	} else{
	  console.log("Try again.");

	Nested if statements
	if (fridgeEmpty = true){
	  if(wallet > 50){
	  console.log(“Let’s go out to eat.”);
	  } else{
	  console.log(“Stop at the store for bread.”);
	  }
	}
	
	Switch statements
	let weather = “sunny”;
	switch (weather){
	  case “snowy”:
	  console.log(“Wear a coat.”);
	  break;
	  case “rainy”:
	  console.log(“Bring an umbrella.”);
	  break;
	  case “sunny”:
	  console.log(“Wear your shades.”);
	  break;
	  case “breezy”:
	  console.log(“Wear a windbreaker.”);
	  break;

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
	let i = 0;
	while (i < 5) {
	  console.log(“Not there yet.”);
	  i++;
	}
	
	do while loop

	let i = 0;
	do {
	  console.log("Iteration number" + i);
	  i++;
	} while (i < 5);
	
	for in loop
	const person = { name: "Adam", age: 40, occupation: "Admin" };
	for (let key in person) {
	  console.log(`${key}: ${person[key]}`);
	}
	
	for of loop
	let shapes = ["circle", "square", "triangle"];
	for (let shape of shapes) {
	  console.log(shape);
	}

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.)
	Yes, JavaScript supports arrays.
	
	Example 1: Array of numbers
	let numbers = [1, 2, 3, 4.5, 5.6, 7.8];
	console.log(numbers);
	
	Example 2: Array of strings
	let fruits = ["apple", "banana", "cherry", "date"];
	console.log(fruits);

Data Structures *If this language provides a standard set of data structures, provide a list of the data structures and their Big-Oh complexity.*	**1. Array** • **Access (by index):** O(1) • **Insertion/Deletion (at the end):** O(1) • **Insertion/Deletion (at the beginning or middle):** O(n) • **Searching (Linear search):** O(n) Arrays in JavaScript are dynamic, allowing you to add or remove elements at any index. However, this can lead to O(n) complexity for operations that require shifting elements (like insertion or deletion at anywhere other than the end of the array). **2. Object** • **Access, Insertion, and Deletion (by key):** Average O(1); Worst O(n) • **Searching (for a value):** O(n)

	Objects in JavaScript are key-value pairs and are often used as hash tables. The average-case complexities for access, insertion, and deletion are constant time, but in the worst case (like when many keys hash to the same value), these operations can degrade to O(n). **3. Set** • **Insertion, Deletion, and Access:** Average O(1); Worst O(n) • **Searching (for a value):** O(1) A set is a collection of unique values. JavaScript's set object allows you to store unique values of any type. **4. Map** • **Insertion, Deletion, Access, and Searching (by key):** Average O(1); Worst O(n) A map holds key-value pairs and remembers the original insertion order of the keys. **5. Stack and Queue** • JavaScript does not have built-in stack or queue classes, but these can be implemented using arrays. • **Stack (LIFO) operations (push/pop):** O(1) • **Queue (FIFO) operations (enqueue/dequeue):** Enqueue (push) is O(1), but dequeue (shift) is O(n) if implemented with an array.

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.
	Yes, JavaScript supports object orientation. Here are two different approaches.
	
	The constructor function
	function Person(name, age) {
	this.name = name;
	this.age = age;
	}
	
	// Adding a method to the prototype
	Person.prototype.greet = function() {
	console.log(`Hello, my name is ${this.name} and I am ${this.age} years old.`);
	};
	
	//Then, create an instance of the 'Person' object:
	
	let person1 = new Person("Alice", 30);
	person1.greet(); // Output: Hello, my name is Alice and I am 30 years old.
	
	Class Syntax (another method)
	class Person {

	constructor(name, age) {
	this.name = name;
	this.age = age;
	}
	
	greet() {
	console.log(`Hello, my name is ${this.name} and I am ${this.age} years old.`);
	}
	}
	
	//Then, instantiate the 'Person' class:
	let person2 = new Person("Bob", 25);
	person2.greet(); // Output: Hello, my name is Bob and I am 25 years old.

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 is traditionally executed in a JavaScript engine, usually in web browsers. JavaScript can also be run on servers or other environments using Node.js, which is a JavaScript runtime built on the V8 engine. This allows JavaScript to be used for server-side scripting and other non-browser environments. Several other languages can be compiled into JavaScript to leverage its runtime environment, including TypeScript, CoffeeScript, ClojureScript, and others.
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..*	Three popular frameworks and libraries used by programmers for JavaScript: 1) React – used for building interfaces, especially for single-page applications where data changes over time. 2) Angular – used for building client-side single-page web apps 3) Vue.js – a progressive framework for building user interfaces
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 and Facebook use JavaScript for web development. Amazon and eBay use JavaScript for eCommerce. PayPal uses JavaScript for its online payment web app. Netflix uses JavaScript—particularly Node.js—for its server-side ops. Microsoft uses JavaScript for front-end development in their online services like Microsoft 365.