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 language, which means that variables can hold values of different types at run time. You do not have to declare the data type of a value explicitly like in Java.
	• Examples of variable declaration
	○ var – declare variables
	▪ var name = “Megan”; //string
	▪ var age = 23; //number
	▪ isWinter = false; //boolean
	○ let – allows you to declare variables with block scope.
	▪ let city = “Chicago”;
	▪ city = “Miami”; //Updates the value
	○ const – declare a constant variable (cannot be reassigned)
	▪ const PI = 3.14159;

Data Types *List all of the data types (and ranges) supported by this language.*	• Primitive Data Types ○ String ▪ Range: 0-2^53 – 1 character ○ Number ▪ Range: -2^1024 to 2^1024 ○ Bigint ▪ Represents integers with arbitrary precision. ▪ Does not have a fixed range. Size is limited by available memory. ○ Boolean ▪ No numerical range. Just true or false. ○ Undefined ▪ Represents a variable that has not been assigned a value. ▪ Doesn’t have a range because it is a single value. ○ Null ▪ Represents an intentional absence of a value. ▪ Doesn’t have a range because it is a single value. ○ Symbol ▪ Does not have a range. Each symbol is unique and immutable. • Non-Primitive/Object Types ○ Object ▪ Can store any number of key-value pairs. Range is just limited by system memory. ○ Array ▪ Max array length: 2^32 -1

	○ Function ▪ No fixed range applies. Functions can be arbitrarily complex. ○ Data ▪ Range od data from: -8,640,000,000,000,000 milliseconds to 8,640,000,000,000,000 milliseconds relative to January 1, 1970 • Can go ~285,616 years to past of future

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 score = 85;
	if (score >= 70) {
	    console.log("You passed the test!");
	}
	
	• if-else statement
	
	let score = 65;
	if (score >=70) {
	    console.log("You passed the test!");
	}else{
	    console.log("You did not pass the test.");
	}
	
	• if-else-if statement
	
	let score = 85;
	if (score >=90) {
	    console.log("Your grade is an A");
	}else if(score >=80) {
	    console.log("Your grade is a B. ");
	} else if (score >=70) {
	    console.log("Your grade is an C");
	} else if (score >=60) {
	    console.log("Your grade is an D");
	} else{
	    console.log("Your grade is an F");
	}
	
	• switch statement
	
	let day = "Tuesday";

```
switch (day){
      case “Monday”:
                  console.log(“Today is laundry day.”);
                  break;
      case “Thursday”:
                  console.log(“Today is bathroom cleaning day.”);
                  break;
      default:
                  console.log(“No chores today.”);
      }
```

- Ternary Operator

```
      let age = 18;

      let eligibility = (age >=18) ? “You can vote.” : “You are too young to vote.”;
      console.log(eligibility);
```

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=1; I <=5; i++){
                  console.log(‘3 x ${i} = ${3 * i}’);
      }
```

- while loop

```
      let count = 5;

      while (count > 0){
                  console.log(‘Countdown: ${count}’);
                  count--;
      }
```

- do while loop

```
let selection;

do {
      console.log(“Menu:”);
```

	console.log(“Start Game”);
	console.log(“Quit);
	
	selection = Math.floor(Math.random() *3);
	console.log(‘User selected: ${selection”’”);
	}while (selection !== 3);
	
	• for in loop
	
	let pet = {
	name: “Drake”,
	age: 1,
	type: “cat”
	};
	
	for (let key in student) {
	console.log(‘${key}: $student[key]}’);
	}
	
	• for of loop
	
	let fruits = [“apple”, “banana”, “orange”];
	
	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.)
	• Example 1: Array of Numbers
	
	let scores = [85, 90, 77, 92, 81]
	
	console.log(scores); //output the scores
	console.log(scores[3]); //access the 4th element
	console.log(scores.length); //get number of elements
	
	• Example 2: Array of Strings
	
	let fruit = [“apple”, “banana”, “orange”];
	
	console.log(fruits); //output the fruits

	console.log(fruit[1]); //access the 2nd element
	console.log(fruits.length); //get number of elements

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 ○ Access: O(1) ○ Search: O(n) ○ Insert: O(1)/O(n) ○ Delete: O(1)/O(n) • Object ○ Access: O(1) ○ Search: O(n) ○ Insert: O(1) ○ Delete:O(1) • Map ○ Access: O(1) ○ Search: O(n) ○ Insert: O(1) ○ Delete: O(1) • Set ○ Access: O(1) ○ Search: O(n) ○ Insert: O(1) ○ Delete: O(1) • WeakMap/WeakSet ○ Access: O(1) ○ Search: O(n) ○ Insert: O(1) ○ Delete: O(1) • Stack ○ Access: O(1) ○ Search: O(n) ○ Insert: O(1) ○ Delete: O(1) • Queue ○ Access: O(1) ○ Search: O(n) ○ Insert: O(1) ○ Delete: O(n) • Priority Queue

  - Access: O(n)
  - Search: O(log n)
  - Insert: O(log n)
  - Delete: O(log n)
- Graph
  - Access: O(1)
  - Search: O(log n)
  - Insert: O(1)
  - Delete: O(1)
- Linked List
  - Access: O(n)
  - Search: O(n)
  - Insert: O(1)/O(n)
  - Delete: O(1)/O(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.

```
// Define a class with a constructor

class Person {
//Constructor to initialize properties
        constructor(firstName, lastName, alternativeName, shoeColor) {
                this.firstName = firstName;
                this.lastName = lastName;
                this.alternativeName = alternativeName;
                this.shoeColor = shoeColor;
        }

        //method to display full name
        getFullName() {
                return ‘${this.firstName} ${this.lastName}’;
        }
}

//Instantiate the object using the constructor
const person = new Person(“Mickey”, “Mouse”, “Steamboat Willie”, “Yellow”);

//Access properties and call methods
console.log(person.firstName);
console.log(person.lastName);
console.log(person.alternativeName);
console.log(person.shoeColor);
```

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?*	• Runtime Environment: o Browser Runtime Environment ▪ V8 – chrome and Node.js ▪ SpiderMonkey – Mozilla Firefox ▪ JavaScriptCore – Apple Safari ▪ Chakra – Microspft edge o Node.js Runtime Environment • Other language with this runtime: o TypeScript, CoffeeScript, Dart, Elm, Kotlin, Rust
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 o Developed by Facebook. Used for building user interfaces, especially single-page applications. It allows developed to build reusable UI components that efficiently update and render in response to data changes. • Node.js o Built for server-side applications. Used for building fast, scalable network application and is ideal for building real-time applications like chat apps and RESTful APIs. • Express.js o A web application framework for Node.js that is typically used for building APIs and web apps. Simplifies the process of building web application and APIs and provides a set of robust features such as routing middleware, and templating engines.
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.***	• All types of industries use JavaScript including web development, E-commerce, Financial Technology, Social Media, Gaming, Education Technology, and Healthcare. o Khan Academy (EdTech) uses JavaScript for building interactive learning tools, quizzes, and video contents. It helps allow students interact with lessons in real time. o Twitter (social media) uses JavaScript in front end and backend development. This is how Twitter can do real-time notifications and live streaming. o PayPal (Financial Technology) uses JavaScript to do dynamic form processing, transaction handling, and integration with APIs for payment processing.

Resources:

https://code-and-cozy.medium.com/javascript-fundamentals-understanding-why-it-is-a-dynamic-and-weak-language-7e31b06b3794#:~:text=JavaScript%20is%20a%20dynamic%20language,type%20of%20a%20variable%20explicitly.

https://www.w3schools.com/js/js_datatypes.asp

https://www.w3schools.com/js/js_object_definition.asp

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Data_structures

https://levelup.gitconnected.com/learning-javascript-selection-structures-7ea61b076d63

https://www.geeksforgeeks.org/loops-in-javascript/

https://stackoverflow.com/questions/9543518/creating-arrays-in-javascript

https://www.freecodecamp.org/news/data-structures-in-javascript-with-examples/

https://dev.to/rahmanmajeed/javascript-the-runtime-environment-35a2

https://www.nobledesktop.com/learn/javascript/industries-and-professions#:~:text=and%20Computer%20Programmers.-,Industries%20such%20as%20financial%20services%2C%20marketing%2C%20healthcare%2C%20and%20ecommerce,engaging%20and%20interactive%20user%20interfaces.