

Language Map for JavaScript

Variable Declaration <i>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.</i>	JavaScript is a dynamically typed language meaning variable types can change during runtime. let numberVar = 42; // Number const stringVar = "Hello, World"; // String let booleanVar = true; // Boolean
Data Types <i>List all of the data types (and ranges) supported by this language.</i>	Null Undefined String Boolean – true and false Number – double-precision 64 bit. Capable of storing positive floating-point numbers between 2^{-1074} and 2^{1024} BigInt Symbol – a unique and immutable primitive value and may be used as the key of an Object property. Non-primitive Data Types Objects Arrays Functions
Selection Structures <i>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.</i>	If Statement: const condition = true; if (condition) { console.log("The condition is true."); } If-Else Statement: const condition = false; if (condition) { console.log("The condition is true."); } else { console.log("The condition is false"); }

	<p>If-Else If-Else const value = 2;</p> <pre> if (value === 1) { console.log("The value is 1."); } else if (value === 2) { console.log("The value is 2."); } else { console.log("The value is neither 1 nor 2."); } </pre> <p>Switch Statement let fruit = "Apple";</p> <pre> switch (fruit) { case "Banana": console.log("It's a banana."); break; case "Apple": console.log("It's an apple."); break; default: console.log("It's something else."); } </pre>
<p>Repetition Structures <i>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.</i></p>	<p>For Loop for (let i = 0; i < 5; i++) { console.log(i); }</p> <p>While Loop let i = 0;</p> <pre> while (i < 5) { console.log(i); i++; } </pre> <p>Do-While Loop</p>

	<pre>let i = 0; do { console.log(i); i++; } while (i < 5);</pre> <p>For-In Loop</p> <pre>const person = { name: "John", age: 30, job: "Developer" }; for (let key in person) { console.log(`\${key}: \${person[key]}`) }</pre> <p>For-Of Loop</p> <pre>const colors = ["red", "green", "blue"]; for (let color of colors) { console.log(color); }</pre>
<p>Arrays</p> <p><i>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.)</i></p>	<pre>let numberArray = [1, 2, 3, 4, 5]; console.log(numberArray); let stringArray = ["apple", "banana", "orange", "grape"]; console.log(stringArray);</pre>
<p>Data Structures</p> <p><i>If this language provides a standard set of data structures, provide a list of the data structures and their Big-Oh complexity.</i></p>	<p>Arrays:</p> <p>Access: $O(1)$</p> <p>Insertion/Deletion at the end: $O(1)$</p> <p>Insertion/Deletion at the beginning: $O(n)$</p> <p>Search (unsorted): $O(n)$</p> <p>Search (sorted): $O(\log n)$</p> <p>Objects (Hash Tables):</p> <p>Insertion/Deletion/Access: $O(1)$ and $O(n)$ in some cases</p> <p>Linked Lists:</p>

	<p>Access: $O(n)$ Insertion/Deletion at the beginning: $O(1)$ Insertion/Deletion at the end: $O(n)$ Search: $O(n)$</p> <p>Sets: Insertion/Deletion/Search: $O(1)$</p> <p>Maps: Insertion/Deletion/Access: $O(1)$</p> <p>Stacks: Push/Pop: $O(1)$</p> <p>Queues: Enqueue/Dequeue: $O(1)$</p> <p>Trees (Binary Search Trees): Search/Insertion/Deletion: $O(\log n)$ and can degrade to $O(n)$ in worst cases for unbalanced trees.</p> <p>Heaps: Insertion/Deletion: $O(\log n)$ Find min/max: $O(1)$</p> <p>Graphs: Search/Insertion/Deletion ?</p>
<p>Objects <i>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.</i></p>	<pre> 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. `); } } let person1 = new Person("Lawrence", 34); console.log(person1.name); </pre>

	<pre>console.log(person1.age) person1.greet();</pre>
Runtime Environment <i>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?</i>	V8: Used in Google Chrome and Node.js Spider Monkey: Used in Mozilla Firefox JavaScriptCore(Nitro): Used in Safari Chakra: Previously used in Microsoft Edge TypeScript CoffeeScript Dart
Libraries/Frameworks <i>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..</i>	React: Developing complex, enterprise-level web applications with a structured and scalable architecture. Vue.js: Developing interactive and dynamic user interfaces, especially in scenarios where you want a framework that is easy to integrate into existing projects. Angular: Developing complex, enterprise-level web applications with a structured and scalable architecture.
Domains <i>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.</i>	Facebook: JavaScript is a primary technology for Facebook's web applications, and they are the maintainers of the React library. Twitter: Twitter uses JavaScript to create a dynamic and responsive user interface for its microblogging platform. They also use technologies like React for certain aspects of their front-end. PayPal: JavaScript is used extensively on PayPal's website for creating a seamless and interactive payment experience. Amazon: JavaScript is a key technology for Amazon's website, facilitating dynamic and interactive user interfaces for online shopping.