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.
	
	let x = 5;
	const y = 20;
	var z = “banana”;

Data Types *List all of the data types (and ranges) supported by this language.*	String - ~9e15 Number – 2^-1074 to 2^1024 Bigint - -9223372036854775808 to +9223372036854775807 Boolean – true and false Undefined - undefined Null - null Symbol -n/a Object – n/a

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 (x < 10){
	  console.log(‘True’);}
	
	if (y > x){
	  console.log(‘True’);}
	else {
	  console.log(‘False);}
	
	if (y < x){
	  console.log(‘True);}
	else if (x > 10){
	  console.log(‘True);}
	else {
	  console.log(‘No condition true’)}

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.***
	while (x < 10){
	  x = x + 2;}
	
	for (i = 0; i < 10; i + 1){
	  console.log(‘i equals ’ + i);}
	do{
	  console.log(x);
	  x = x + 1;} while (x < 10);

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.)*	const names = [“Ben”, “Steve”, “Joe”]; const numbers = [5, 23, 6];
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 - accessing a specified index is O(1) since it does not matter what size the array is. Adding to the end of an array is O(1) unless the array is full, then it’s O(n). Searching the array is O(n) since the complexity grows with the size of the array. Hashtable – assuming no collisions it is O(1) since the hashkey will point directly to the index in the array regardless of size. Stack – push and pop are O(1) since the size of the stack does not matter. Popping all elements and accessing a specific element are O(n) since it depends on the size of the stack. Queue – enqueue is O(n) since the queue is traversed until it reaches the end. Dequeue is O(1) since it always removes the item from the front of the list. Linked List - When accessing elements of a linked list, speed is proportional to size of the list with Big O(n). Since we must traverse the entire list in order to get to the desired element. Tree/Heap – Searching the tree or heap is O(n) because the complexity grows with its size.

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.
	const person = {
	firstName: "John",
	lastName : "Doe",
	id : 5566,
	fullName : function() {
	return this.firstName + " " + this.lastName;
	}
	};

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 executed in either a browser’s runtime environment or the Node runtime environment. I could not find any other languages that use these runtime environments.
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..*	Node.js is used for its capacity for command-line tools and server-side scripting. Angular JS dynamically loads content from the webserver rather than the web browser. React simplifies building interactive UIs.
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.***	Microsoft uses JavaScript to build its Edge web browser. PayPal uses it on the front end of their website. Google uses it in its search results, web client, Google Docs, and Chrome.