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.	C# is strongly typed. let a = 10; let name = "Alyssa"; var x;//(for older browsers) x = 15; const pi = 3.14;
Data Types List all of the data types (and ranges) supported by this language.	Data Types: String Number (all double - 64 bit floating) Bigint (store int value too big for normal JS number) Boolean (true/false) Undefined (variable without value, type undefined) Null Symbol Object {}
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.	1. if if (Number > 2) { document.write("This is greater than 2."} 2. if else if(Number>2) { document.write("This is greater than 2."} else { document.write("This is not greater than 2.")} 3. switch switch(Number) { default: document.write("number"); break; case 1: document.write("Number 1"); break; case 2: document.write("Number 2"); break; case 3: document.write("Number 3"); break; case 4:

	document.write("Number 4");
	break;
	}
Repetition Structures	1. while
Provide examples of all repetition structures supported	while(a>10) {
by this language (loops, etc.) Don't just list them, show	document.write("This is greater than 10.");
code samples of how each would look in a real	a++;
program.	
	2. for
	for (let $i = 1$; $i < a$; $i + +$) {
	console.log("Count" + i); }
	3. do-while
	do {
	document.write("Hello.");
	a++;
	} while (a<10);
A mmaxic	const integers = $[1,3,5,7]$;
Arrays	Const Integers [1,5,5,7],
If this language supports arrays, provide at least two	const names = [];
examples of creating an array with a primitive or	names[1] = "Alyssa";
String data types (e.g. float, int, String, etc.)	names[2] = "John";
	names[3] = "Ron";
Data Structures	Objects(hash-tables)
	Insertion - O(1)
If this language provides a standard set of data	Removal - O(1)
structures, provide a list of the data structures and their	Search - O(1)
Big-Oh complexity.	
	Stacks:
	Insertion - O(1)
	Removal - O(1)
	Searching - O(n)
	Access - O(n)
	Ougues
	Queues Insertion - O(1)
	Removal - O(1)
	Searching - O(n)
	Access - O(n)
	Linked Lists
	Liller Fig.

Insertion - O(1) Removal - O(n) Search - O(n) Access - O(n) Doubly Linked Lists Insertion - O(1) Removal - O(1) Search - O(n) Access - O(n) Trees Insertion - O(log(n)) Removal - O(log(n)) Removal - O(log(n)) Search - O(log(n)) Access - O(log(n) Access - O(log(n)) Access - O(lo		,
Search - O(n) Access - O(n) Doubly Linked Lists Insertion - O(1) Removal - O(1) Search - O(n) Access - O(n) Trees Insertion - O(log(n)) Removal - O(log(n)) Search - O(log(n)) Access - O(log(n)) Heaps Insertion - O(log n) Removal - O(log n) Search - O(n) Sort - O(n log n) Heapify - O(n) Graphs Insertion - O(1) Removal - O(n) Sort - O(n log n) Heapify - 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. Tooling the first of the provided an example of how you would write a simple object with a default constructor and then how you would instantiate it. Function Person(name, age, eyeColor) { this name = name; this age = age; this eyeColor = eyeColor; } Runtime Environment		
Access - O(n) Doubly Linked Lists Insertion - O(1) Removal - O(1) Search - O(n) Access - O(n) Trees Insertion - O(log(n)) Removal - O(log(n)) Search - O(log(n)) Access - O(log(n) Access - O		
Doubly Linked Lists Insertion - O(1) Removal - O(1) Search - O(n) Access - O(n) Trees Insertion - O(log(n)) Removal - O(log(n)) Search - O(log(n)) Search - O(log(n)) Access - O(log(n) Access - O(log(n)) Access - O(log(n)		Search - O(n)
Doubly Linked Lists Insertion - O(1) Removal - O(1) Search - O(n) Access - O(n) Trees Insertion - O(log(n)) Removal - O(log(n)) Search - O(log(n)) Search - O(log(n)) Access - O(log(n) Access - O(log(n)) Access - O(log(n)		Access - O(n)
Insertion - O(1) Removal - O(1) Search - O(n) Access - O(n) Trees Insertion - O(log(n)) Removal - O(log(n)) Search - O(log(n)) Access - O(log(n) Access - O(log(n)) Access - O(log(n) Acces		
Insertion - O(1) Removal - O(1) Search - O(n) Access - O(n) Trees Insertion - O(log(n)) Removal - O(log(n)) Search - O(log(n)) Access - O(log(n) Access - O(log(n)) Access - O(log(n) Acces		Doubly Linked Lists
Removal - O(1) Search - O(n) Access - O(n) Trees Insertion - O(log(n)) Removal - O(log(n)) Access - O(log(n)) Access - O(log(n)) Access - O(log(n)) Heaps Insertion - O(log n) Removal - O(log n) Search - O(n) Sort - O(n log n) Heapify - O(n) Graphs Insertion - O(1) Removal - O(n) Search - O(n) Sort - O(n log n) Heapify - O(n) Graphs Insertion - O(1) Removal - O(n) Search - O(n) Search - O(n) Search - O(n) Const human = {name: "Alyssa", age: 86, eyeColor: "purple"}; bool = new Boolean(); //bool is a Boolean object function Person(name, age, eyeColor) { this.name = name; this.age = age; this.eyeColor = eyeColor; } Runtime Environment		
Search - O(n) Access - O(n) Trees Insertion - O(log(n)) Removal - O(log(n)) Search - O(log(n)) Access - O(log(n)) Access - O(log(n)) Heaps Insertion - O(log n) Removal - O(log n) Search - O(n) Sort - O(n log n) Heapify - O(n) Graphs Insertion - O(1) Removal - O(n) Search - O(n) Coraphs Insertion - O(1) Removal - O(n) Search - O(n) Sea		
Access - O(n) Trees Insertion - O(log(n)) Removal - O(log(n)) Search - O(log(n)) Access - O(log(n)) Access - O(log(n)) Access - O(log(n)) Heaps Insertion - O(log n) Removal - O(log n) Search - O(n) Sort - O(n log n) Heapify - O(n) Graphs Insertion - O(1) Removal - O(n) Search - O(n) Sort - O(n log n) Heapify - O(n) Corphs Insertion - O(1) Removal - O(n) Search -		
Trees Insertion - O(log(n)) Removal - O(log(n)) Search - O(log(n)) Access - O(log(n)) Access - O(log(n)) Access - O(log n) Removal - O(log n) Removal - O(log n) Removal - O(log n) Search - O(n) Sort - O(n log n) Heapify - O(n) Graphs Insertion - O(1) Removal - O(n) Search - O(n) 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. Runtime Environment Trees Insertion - O(log(n)) Removal - O(log n) Removal - O(n) Search - O(n) Sear		
Insertion - O(log(n)) Removal - O(log(n)) Search - O(log(n)) Access - O(log(n)) Heaps Insertion - O(log n) Removal - O(log n) Removal - O(log n) Search - O(n) Sort - O(n log n) Heapify - O(n) Graphs Insertion - O(1) Removal - O(n) Search		Access - O(II)
Insertion - O(log(n)) Removal - O(log(n)) Search - O(log(n)) Access - O(log(n)) Heaps Insertion - O(log n) Removal - O(log n) Removal - O(log n) Search - O(n) Sort - O(n log n) Heapify - O(n) Graphs Insertion - O(1) Removal - O(n) Search		Trees
Removal - O(log(n)) Search - O(log(n)) Access - O(log(n)) Heaps Insertion - O(log n) Removal - O(log n) Removal - O(log n) Search - O(n) Sort - O(n log n) Heapify - O(n) Graphs Insertion - O(1) Removal - O(n) Search - O(n) Search - O(n) Const human = {name: "Alyssa", age: 86, eyeColor: "purple"}; bool = new Boolean(); //bool is a Boolean object function Person(name, age, eyeColor) { this.name = name; this.age = age; this.eyeColor = eyeColor; } Runtime Environment Runtime Environment Runtime Environment Removal - O(log(n)) Removal - O(log n) Removal - O(n) Search		
Search - O(log(n)) Access - O(log(n)) Heaps Insertion - O(log n) Removal - O(log n) Search - O(n) Sort - O(n log n) Heapify - O(n) Graphs Insertion - O(l) Removal - O(n) Search - O(n) Search - O(n) Graphs Insertion - O(l) Removal - O(n) Search - O(n) Search - O(n) const human = {name: "Alyssa", age: 86, eyeColor: "purple"}; bool = new Boolean(); //bool is a Boolean object function Person(name, age, eyeColor) { this name = name: this age = age; this eyeColor = eyeColor; } Runtime Environment -There are two runtime environments for JavaScript		
Access - O(log(n)) Heaps Insertion - O(log n) Removal - O(log n) Search - O(n) Sort - O(n log n) Heapify - O(n) Graphs Insertion - O(1) Removal - O(n) Search - O(n) Sea		
Heaps Insertion - O(log n) Removal - O(n) Search - O(n) Sort - O(n log n) Heapify - O(n) Graphs Insertion - O(1) Removal - O(n) Search - O(n) Search - O(n) Graphs Insertion - O(1) Removal - O(n) Search - O(n) Search - O(n) const human = {name: "Alyssa", age: 86, eyeColor: "purple"}; 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. bool = new Boolean(); //bool is a Boolean object function Person(name, age, eyeColor) { this.name = name; this.age = age; this.eyeColor = eyeColor; } Runtime Environment -There are two runtime environments for JavaScript		
Insertion - O(log n) Removal - O(log n) Search - O(n) Sort - O(n log n) Heapify - O(n) Graphs Insertion - O(1) Removal - O(1) Removal - O(n) Search - O(n) Const human = {name: "Alyssa", age: 86, eyeColor: "purple"}; 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. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object with a default construction of how you would instantiate it.		Access - O(log(li))
Insertion - O(log n) Removal - O(log n) Search - O(n) Sort - O(n log n) Heapify - O(n) Graphs Insertion - O(1) Removal - O(1) Removal - O(n) Search - O(n) Const human = {name: "Alyssa", age: 86, eyeColor: "purple"}; 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. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object-orientation, provide an example of how you would instantiate it. In this language support object with a default construction of how you would instantiate it.		Heaps
Removal - O(log n) Search - O(n) Sort - O(n log n) Heapify - O(n) Graphs Insertion - O(1) Removal - O(n) Search - 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. Runtime Environment Removal - O(log n) Search - O(n) Coraphs Insertion - O(1) Removal - O(n) Search - O(n)		
Search - O(n) Sort - O(n log n) Heapify - O(n) Graphs Insertion - O(1) Removal - O(n) Search - 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. Puntime Environment Search - O(n) Graphs Insertion - O(1) Removal - O(n) Search - O		
Sort - O(n log n) Heapify - O(n) Graphs Insertion - O(1) Removal - O(n) Search - O(n) Const human = {name: "Alyssa", age: 86, eyeColor: "purple"}; 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. bool = new Boolean(); //bool is a Boolean object function Person(name, age, eyeColor) { this.name = name; this.age = age; this.age = age; this.eyeColor = eyeColor; } Runtime Environment -There are two runtime environments for JavaScript		
Heapify - O(n) Graphs Insertion - O(1) Removal - O(n) Search - O(n) 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. Function Person(name, age, eyeColor) { this.name = name; this.name = name; this.name = name; this.age = age; this.eyeColor = eyeColor; } Runtime Environment -There are two runtime environments for JavaScript		
Graphs Insertion - O(1) Removal - O(n) Search - O(n) Search - O(n) Search graphs Insertion - O(1) Removal - O(n) Search - O(n) Search graphs Insertion - O(1) Removal - O		
Insertion - O(1) Removal - O(n) Search - O(n) Cobjects 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 human = {name: "Alyssa", age: 86, eyeColor: "purple"}; bool = new Boolean(); //bool is a Boolean object function Person(name, age, eyeColor) { this.name = name; this.age = age; this.eyeColor = eyeColor; } Runtime Environment -There are two runtime environments for JavaScript		neapity - O(ii)
Insertion - O(1) Removal - O(n) Search - O(n) Cobjects 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 human = {name: "Alyssa", age: 86, eyeColor: "purple"}; bool = new Boolean(); //bool is a Boolean object function Person(name, age, eyeColor) { this.name = name; this.age = age; this.eyeColor = eyeColor; } Runtime Environment -There are two runtime environments for JavaScript		Graphs
Removal - O(n) Search - 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. bool = new Boolean(); //bool is a Boolean object function Person(name, age, eyeColor) { this.name = name; this.age = age; this.eyeColor = eyeColor; } Runtime Environment removal - O(n) Search -		
Search - 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. bool = new Boolean(); //bool is a Boolean object function Person(name, age, eyeColor) { this.name = name; this.age = age; this.eyeColor = eyeColor; } Runtime Environment -There are two runtime environments for JavaScript		
Objectsconst human = {name: "Alyssa", age: 86, eyeColor: "purple"};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.bool = new Boolean(); //bool is a Boolean objectit.function Person(name, age, eyeColor) { this.name = name; this.age = age; this.eyeColor = eyeColor; }Runtime Environment-There are two runtime environments for JavaScript		
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. function Person(name, age, eyeColor) { this.name = name; this.age = age; this.eyeColor = eyeColor; } Runtime Environment -There are two runtime environments for JavaScript	Objects	
Example of how you would write a simple object with a default constructor and then how you would instantiate it. bool = new Boolean(); //bool is a Boolean object	, o	const numari marie. Aryssa, age. 60, eyecolor. purple 3,
default constructor and then how you would instantiate it. function Person(name, age, eyeColor) { this.name = name; this.age = age; this.eyeColor = eyeColor; } Runtime Environment -There are two runtime environments for JavaScript		hool = now Poolean(): //hool is a Poolean abject
function Person(name, age, eyeColor) {		DOOI - HEW DOOIEAH(), //DOOI IS A DOOIEAH OUJECT
this.name = name; this.age = age; this.eyeColor = eyeColor; } Runtime Environment -There are two runtime environments for JavaScript		function Demontrary and accordance (
this.age = age; this.eyeColor = eyeColor; } Runtime Environment -There are two runtime environments for JavaScript	it.	
this.eyeColor = eyeColor; } Runtime Environment -There are two runtime environments for JavaScript		
Runtime Environment -There are two runtime environments for JavaScript		
		tnis.eyeColor = eyeColor;
1. browsers runtime environment	Runtime Environment	
front end application		front end application

TTT	
What runtime environment does this language compile	window object
to? For example, Java compiles to the Java Virtual	Chrome: V8 JS engine, Firefox: SpiderMonkey,
Machine.	2. the Node runtime enviornment
Do other languages also compile to this runtime?	allows JS code to be run on server instead of browser
	back end application
	file system, databases, networks attached to server
Libraries/Frameworks	React
What are the popular libraries or frameworks used by	- base for mobile or single page applications
programmers for this language? List at least three (3)	- user interfaces while developing web app/ interactive sites handle high traffic
and describe what they are used for	- Facebook, Instagram, Whatsapp
and describe multiney are assurption.	
	Ember.js
	- support two way binding
	- complicated user inter faces
	- LinkedIn, Netflix, Nordstrom
	Node.js
	- backend framework but can go full stack
	- good for real time apps with communication
	- infrastructure and web server work
	- IBM, Microsoft, Netflix, Walmart
Domains	Web development, web applications presentations, server app, web servers, games(with HTML5),
What industries or domains use this programming	art, smartwatch app, mobile apps
language? Provide specific examples of companies	Netflix:
that use this language and what they use it for. E.g.	Uses Javascript both as their front and back end. Switched from JS front end and Java backend to
Company X uses C# for its line of business	get rid of need for two languages, better performance, easier debugging.
	Facebook:
applications.	Facebook interface is actually collection of different JavaScript apps
	Facebook create React, very popular JS framework
	1 accook create react, very popular 35 numework
	All companies listed in libraries/frameworks section obviously use JAvaScript in their chosen framework