Stephen Brom

11/19/2023

CS311-ON

Basic Structures of the JavaScript Programming Language

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. This means you don't have to specify the data type of a variable when you declare it, and data types can be changed or converted automatically as needed.
	Number: In JavaScript, there's no distinction between integer values and floating-point values. You can declare a number like this:
	let num = 10; // an integer
	let floatNum = 10.5; // a floating-point number
	String: A sequence of characters used to represent text. You can declare a string like this:
	let str = "Hello, World!";
	Boolean: A logical entity that can have two values: true or false. You can declare a boolean like this:
	let isTrue = true;

Data Types *List all of the data types (and ranges) supported by this language.*	JavaScript supports the following data types: 1. **Number**: There's no distinction between integer values and floating-point values in JavaScript. The range of values is approximately ±5.00e-324 to ±1.79e+308. 2. **String**: A sequence of characters used to represent text. There's no theoretical limit to the length of a string in JavaScript. 3. **Boolean**: A logical entity that can have two values: true or false. 4. **Null**: It has one value: null. 5. **Undefined**: A variable that has not been assigned a value is of type undefined.

	6. **Object**: An individual instance of the data structure type. This includes arrays and functions, which are specific types of objects in JavaScript. 7. **Symbol**: A unique and immutable primitive value and may be used as the key of an Object property. 8. **BigInt**: It can represent integers in the arbitrary precision format. This is used for mathematically intense tasks, and also for handling large numbers.

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: Executes a statement if a specified condition is truthy. If the condition is falsy, another statement can be executed.
	let x = 10;
	if (x > 5) {
	console.log('x is greater than 5');
	}
	If-Else Statement: The if statement executes a statement if a specified condition is truthy. If the condition is falsy, another statement can be executed.
	let x = 10;
	if (x > 5) {
	console.log('x is greater than 5');
	} else {
	console.log('x is not greater than 5');
	}
	Else If Statement: The else if statement allows for multiple conditions to be tested in sequence.
	let x = 10;
	if (x > 10) {
	console.log('x is greater than 10');
	} else if (x < 10) {
	console.log('x is less than 10');
	} else {
	console.log('x is equal to 10');
	}
	Switch Statement: The switch statement evaluates an expression and executes the associated case statement.

	let fruit = 'Apple';
	switch (fruit) {
	case 'Banana':
	console.log('Banana is good!');
	break;
	case 'Apple':
	console.log('I like apples!');
	break;
	default:
	console.log('No fruit selected');
	}

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.***
	1. For Loop: A for loop repeats until a specified condition evaluates to false.
	
	for (let i = 0; i < 5; i++) {
	
	console.log(i);
	}
	
	2. While Loop: A while loop executes its statements as long as a specified condition evaluates to true.
	
	let i = 0;
	while (i < 5) {
	console.log(i);
	i++;
	}

	3. Do-While Loop: The do...while loop is similar to the while loop, except that the condition is tested at the end of the loop, so the loop will always be executed at least once.
	
	let i = 0;
	do {
	  console.log(i);
	  i++;
	} while (i < 5);
	
	4. For-In Loop: The for...in loop is used to iterate over the properties of an object (or elements of an array).
	
	let obj = {a: 1, b: 2, c: 3};
	for (let prop in obj) {
	  console.log(`obj.${prop} = ${obj[prop]}`);
	}
	
	5. For-Of Loop: The for...of loop is used to iterate over iterable objects like arrays, strings, etc.
	
	let arr = [1, 2, 3, 4, 5];
	for (let value of arr) {
	  console.log(value);
	}

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.)*	1. **Array of Numbers (Integer and Float)**: let numArray = [1, 2, 3, 4.5, 5.6]; // an array of numbers

	2. **Array of Strings**: let strArray = ["Hello", "World", "!"]; // an array of strings
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** o Access: O(1) o Insertion/Deletion at the end: O(1) o Insertion/Deletion at the beginning: O(n) 2. **Object (similar to Map)** o Access: O(1) o Insertion/Deletion: O(1) o Searching: O(n) 3. **Set** o Access: N/A (Sets do not support access by index) o Insertion/Deletion: O(1) o Searching: O(1) 4. **Map** o Access: O(1) o Insertion/Deletion: O(1) o Searching: O(1)

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 simple object with a constructor
	class MyObject {
	constructor() {
	this.message = "Hello, World!";
	}
	}
	
	// Instantiate the object
	let obj = new MyObject();

	// Access a property of the object console.log(obj.message); // Outputs: "Hello, World!"
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 typically run in a web browser, which serves as its runtime environment. The JavaScript code is interpreted and executed by the JavaScript engine in the browser. Different browsers use different JavaScript engines: for example, Google Chrome uses the V8 engine, Firefox uses SpiderMonkey, and Safari uses JavaScriptCore. However, JavaScript can also be run outside of a browser in environments like Node.js, which also uses the V8 engine. As for other languages compiling to this runtime, there are indeed languages that transpile to JavaScript so they can be run in the JavaScript runtime environment. These include but are not limited to TypeScript, CoffeeScript, and ClojureScript. These languages offer different features or syntax, but ultimately compile down to JavaScript code that can be executed in the browser or other JavaScript 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..*	1. **React**: React is a JavaScript library for building user interfaces, particularly single-page applications where you need a fast, interactive user interface. It allows developers to create reusable UI components. React was developed by Facebook and is widely used in production both at Facebook and at other major companies. 2. **Angular**: Angular is a platform for building web applications. It provides a framework for client-side MVC and MVVM architectures along with components commonly used in rich internet applications. Angular was developed and is maintained by Google. 3. **Vue.js**: Vue.js is an open-source JavaScript framework for building user interfaces and single-page applications. It was created by Evan You, a former Google employee. Vue is known for its simplicity and ease of use, especially for developers coming from an HTML/CSS background.

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.***	JavaScript is used across a wide range of industries and domains due to its versatility and the fact that it's the primary language for web development. Here are some specific examples of companies that use JavaScript and what they use it for: 1. **Microsoft**: Microsoft uses JavaScript to build its Edge web browser. All browsers need to process and execute JavaScript efficiently, so Microsoft has developed and maintains its own JavaScript engine for Edge. 2. **PayPal:** PayPal has been using JavaScript on the front end of their website for a long time. The online payment giant was one of the earliest adopters of NodeJS and during an overhaul of their account overview page, they decided to try building the page in Node at the same time as their usual JavaScript development 3. **Netflix:** Netflix started out using JavaScript for just about everything. Over time, Netflix moved away from its more traditional structure into the cloud and started to introduce NodeJS. With NodeJS, Netflix was able to break down pieces of their user interface into individual services. 4. **Groupon**: Groupon used to be infamously slow. Thanks to difficulties in speed and maintainability, they decided to change over to NodeJS. Node allowed Groupon to rebuild their entire US website by breaking down everything into individual NodeJS web applications. 5. **eBay:** eBay uses JavaScript for front end development and backend development. 6. **Facebook:** Facebook uses JavaScript for front end development and backend development. 7. **Google:** Google uses JavaScript for front end development and backend development.