Rheann L. Vera

CS-300-ON

November 19, 2023

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, you're able to declare variables in three ways. The same variable can be used to hold different data types. Mainly using let and const. The third keyword is var but should be used when supporting old browsers. const is used when a value cannot be changed or should not be changed like with arrays or objects. If you're not needing to use const, "let" should be used to declare variables. let name = "Rheann"; const age = 30; var car = "Toyota";
Data Types *List all of the data types (and ranges) supported by this language.*	There are 87 primitive datatypes. **Null** type having only one value, null. Representing the absence of an object. **Undefined** type also has one value, undefined, the absence of a value. **Boolean** has two values, true and false in conditional operations. **Number** type, there is only one type and it's a double 64-bit floating point, they can be written with or without decimals, between 2^{-1074} and 2^{1024}. **BigInt** type, represents integers as arbitrary figures. This type can store and operate large integers past the number type's range. **String** type represents textual data as a sequence of 16-bit integer values. **Symbol** type is an immutable primitive value, used as the key to an object property. They are called "atoms" they create unique property keys, that are meant to not clash with keys from other code.

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 wear;
	if (temp > 70) {
	wear = "shorts";
	}
	
	If-else statement
	if (temp > 70) {
	wear = "shorts";
	} else { wear = "pants";
	}
	
	Else-if statement (two conditions)
	if (temp <30) {
	wear = "heavy coat";
	} else if (temp < 60) {
	wear = " light coat";

	} else {
	wear = “sweater”;
	}
	Switch statement
	const food = “fruit”;
	switch(fruit) {
	case ‘apple’: {
	console.log(‘harvested from a tree’);
	break;
	}
	case ‘grape’:{
	console.log(‘harvested from a vine’);
	break;
	}
	case ‘strawberry’:{
	console.log(‘harvested from the ground’);
	break;
	}
	default: {
	console.log(‘No fruits to harvest’);
	}

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 – loops through a block of code a number of times
	for (let i = 0; i > 0; i++) {
	console.log(i);
	}
	
	For-in loop – loops through the properties of an object
	const nums = [1, 2, 3, 4, 5];
	for (let num in nums) {
	console.log(nums[num]);
	}
	
	For-of – loops through the values of an iterable object (array)
	const nums =[1, 2, 3, 4, 5];
	for (let num of nums){
	console.log(num);
	}

While – loops through a block of code while a specified condition is true

```
let x = 0;
while (x > 10){
x--;
} console.log(x);
```

Do-while – loops through a block of code while a specified condition is true

```
let x = 0;
do {
x--;
console.log(x);
} 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 cars = [“Toyota”, “Honda”, “Subaru”];

let z = new Array (1, 2, 3);

const fruit = []
fruit[0] = “mango”;
fruit[1] = “banana”;
fruit[2]= “pineapple”;
```

Data Structures

If this language provides a standard set of data structures, provide a list of the data structures and their Big-Oh complexity.

	Access	Search	Insertion	Deletion
Array	O(1)	O(n)	O(n)	O(n)
Stack	O(n)	O(n)	O(1)	O(1)
Queue	O(n)	O(n)	O(1)	O(1)
Hash table	O(n)	O(n)	O(n)	O(n)
Dictionary	O(n)	O(n)	O(n)	O(n)
Linked list	O(n)	O(n)	O(n)	O(n)
Sets	O(log n)	O(log n)	O(log n)	O(1)
Binary Trees	O(log n)	O(log n)	O(log n)	O(log 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.

```
function Person(first, last, age, eye) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eye;
}
```

	// Create a Person object const myHusband = new Person("Jorge", "Vera", 29, "brown");
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 has two runtime environments, a browser such as Chrome or FireFox and Node runtime environment created for the purpose of executing JavaScript code without a browser.
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. ReactJs – They're based on a reuseable component, they are code blocks classified as either classes or functions. Each section represents a certain part of a page, such as a button, search bar or logo. 2. Angular – Designed to develop single page application (SPA) giving developers parameters to combine JavaScript with HTML And CSS. 3. Vue.js – A framework for creating a creative UI, designed to be adaptable. Uses a command-line interface (CLI) that is basic tooling to speed up development.
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.***	1. PayPal, uses NodeJS in their production which is the runtime environment specific to executing JavaScript, utilized for their backend development. 2. Instagram – uses ReactJS library to implement many of their features such as video delivery, uploading and/or Geolocations. 3. Uber, utilizes NodeJS to develop and maintain its implementations such as matching rides and drivers and provides quick iterations.