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, meaning variables are not directly associated with any particular value type and any variable can be assigned (and reassigned) values of all types. JavaScript is also weakly typed, meaning it allows implicit type conversion when an operation involves mismatched types, instead of throwing type errors.

Variables are declared in JavaScript similarly to how they are declared in Java and C#, except you use "var", "let", or "const" instead of the data type. "Var" is used in all JavaScript code from 1995 – 2015, so you must use "var" if you want your code to run in older browsers. "Let" will not allow you to redeclare a variable, but "var" will. "Const" is used to declare constant references.

Examples:
1) var myNum = 10; 2) let myWord= "Hello"; 3) const PI = 3.14;

Data Types

List all of the data types (and ranges) supported by this language.

Data Type	Range
Boolean	True or false
null	Refers to some nonexistent or invalid object or address
undefined	Means a variable hasn't been assigned any value
number	$-2^{53} - 1$ to $2^{53} - 1$
BigInt	Outside the number range ($-2^{53} - 1$ to $2^{53} - 1$)
string	A sequence of zero or more characters
symbol	Used for unique identifiers in an object
object	Store keyed collections of various data and more complex entities

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.***

Examples of IF

```
if (condition)
{
    //code to be executed if true
}
```

```
if (billPaid)
{
    balance = 90;
}
```

Examples of IF ELSE

```
if (condition)
{
    //code to be executed if true
}
else
{
    //code to be executed if false
}
```

```
if (billPaid)
{
    balance = 90;
}
else
{
    balance = 200;
}
```

Examples of ELSE IF

```
if (condition1)
{
    //code to be executed if 1 is true
}
else if (condition2)
{
    //code to be executed if 1 is false & 2 is true
}
else
{
    //code to be executed if 1 is false & 2 is false
}
```

```
if (result = 10)
{
    level = "normal";
}
else if (result < 10)
{
    level = "low";
}
else
{
    level = "high";
}
```

Examples of SWITCH

```
switch(expression)
{
    case x:
        //code to be executed if x
        break;
    case y:
        //code to be executed if y
        break;
    default:
        //code to be executed if not x or y
        break;
}
```

```
switch(position)
{
    case 1:
        rank = "winner";
        break;
    case 2:
        rank = "runner up";
        break;
    default:
        rank = "contestant";
        break;
}
```

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.***

Examples of FOR LOOP – loops through a block of code a number of times

```
for (statement1; statement 2; statement 3)
{
    //code to be executed
}
```

```
for (let i = 0; i < 5; i++)
{
    text += "The number is " + i + "<br>";
}
```

Examples of FOR IN LOOP – loops through the properties of an object

```
for (key in object)
{
    //code to be executed
}
```

```
for (let x in person)
{
    text += person[x];
}
```

Don't use when index order is important!

```
for (variable in array)
{
    //code to be executed
}
```

```
for (let x in numbers)
{
    text += numbers[x];
}
```

Examples of FOR OF LOOP – loops through the values of an iterable object (arrays, strings, maps, etc)

```
for (variable of iterable)
{
    //code to be executed
}
```

```
for (let x of cars)
{
    text += x;
}
```

Examples of WHILE LOOP – loops through a block of code while a specified condition is true

```
while (condition)
{
    //code to be executed
}
```

```
while (i < 10)
{
    text += "The number is " + i;
    i++;
}
```

Examples of DO/WHILE LOOP – executes at least once and then loops through a block of code while a specified condition is true

```
do
{
    //code to be executed at least once
    //and while condition is true
}
while (condition);
```

```
do
{
    text += "The number is " + i;
    i++;
}
while (i < 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 myArray = [];
myArray[0] = "red";
myArray[1] = "blue";
myArray[2] = "green";

OR

const myArray = ["red", "blue", "green"];

OR

const myArray = new Array("red", "blue", "green");   ← less common and unecessary
```

Data Structures

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

Data Structure	Access	Search	Insert	Delete
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)
Singly Linked List	O(n)	O(n)	O(1)	O(1)
Doubly Linked List	O(n)	O(n)	O(1)	O(1)
Hash Table	N/A	O(1)	O(1)	O(1)
Binary Search Tree	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 Car (make, model, year)
{
    this.make = make;
    this.model = model;
    this.year = year;
}
```

```
const myCar = new Car("Ford", "Focus", 2017);
```

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 most commonly executed in the runtime environment of a browser, like Chrome or Firefox. More recently (in 2009), the Node runtime environment was created to execute JavaScript code without a browser for full-stack applications. JavaScript is the only language that Node.js supports natively, but many compile-to-JS languages, such as CoffeeScript, Dart, and TypeScript, can compile to the Node runtime environment.
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..*	A few popular libraries and frameworks used by JavaScript programmers are: • jQuery: event handling, CSS animation, and developing Ajax applications • React: created by Facebook and used by Twitter, can be used as a base in the development of single-page, mobile, or server-rendered applications • Angular: created by Google for YouTube and Gmail, can be used to reuse code for web, mobile web, native mobile, and native desktop applications
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 a dynamic programming language that is used for various purposes, including web development, web apps, game creation, and more. It enables companies to add dynamic features to websites that aren't possible with just HTML and CSS. JavaScript is used by the following companies: • Facebook: website and mobile applications • Google: provides suggestions when typing in the search box • Walmart: tailors content based on device type and browser capabilities • LinkedIn: development of its efficient mobile app • Uber: tracks driver and rider locations and incoming ride requests to match riders as fast as possible