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.
- Variables are declared with a keyword, a name for the data type, an equals sign, the value, and ends with a semicolon
- Keywords determine permanence of declaration
- var – keyword that is used especially for older browsers (1995-2015)
- const – keyword that is used to a set a variables value that cannot be changed
- let – keyword that is used to denote a variable's value that can be changed
- Examples: var x = 5; const str1 = "Hello World"; let total = 5 + 5;

Data Types

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

TYPE	DESCRIPTION	RANGE
Number	Double precision 64-bit binary format IEEE 754 value	Floating point: 2^{-1074} - 2^{1024} Integers: $(2^{-53} – 1) – (2^{53} – 1)$
String	Elements of 16-bit unsigned integer values Once a string is created you cannot modify it	Sequence of Unicode characters
Boolean	True or false value	True or False
Symbol	A unique and immutable primitive value	Sequence of Unicode characters
Object	A value in memory referenced by an identifier	
undefined	A variable that has been assigned no value	undefined
Null	Represents the absence of any object value	null
BigInt	64-bit unsigned integer	Integers larger than $(2^{-53} – 1) – (2^{53} – 1)$

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.***

SELECTION STRUCTURE	EXAMPLE

if	if (x <10)
	{
	x = x+1;
	}

if-else	if (x < 10)
	{
	  x = x+1;
	}
	else
	{
	  int y = x;
	}

else-if	if (x < 10)
	{
	  x = x+1;
	}
	else if (x > 10)
	{
	  x = x-1;
	}
	else
	{
	  int y = x;
	}

Conditional	(x > y) ? print(“X is greater than Y!”) : print(“X is not greater than Y!”);

Switch	int value;
	string output = “”;
	switch (value)
	{
	  case 1:
	  output = “A”;
	  break;
	  case 2:
	  output = “B”;
	  break;
	  case 3:
	  output = “C”;

		break; case 4: output = “D”; break; case 5: output = “F”; break; default: output = “default”; 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.***

REPETITION STRUCTURE	EXAMPLE

while loop	int num = 0;
	while (num < 5)
	{
	num +=1;
	}

for loop	for (int i = 0; i < 10; i++)
	{
	print(i);
	}

for-in loop	const employee = {id:1234, lname:”Doe”, fname:”John”, age: 35};
	
	let text = “”;
	for (let x in employee)
	{
	Text+= person [x];
	}

For-of loop	const employee = {id:1234, lname:”Doe”, fname:”John”, age: 35};

```
let text = “”;
for (let x of employee)
{
    Text+= person [x];
}
```

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.)

```
var thisArray = {“this”, “is”, “an”, “array”};

OR

var thatArray = new Array (4);
var thatArray[0] = 0;
var thatArray[1] = 1;
var thatArray[2] = 2;
var thatArray[3] = 3;
var thatArray[4] = 4;
```

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	Time Complexity								Space Complexity
	Average				Worst				Worst
	Access	Search	Insertion	Deletion	Access	Search	Insertion	Deletion	
Array	O(1)	O(n)	O(n)	O(n)	O(1)	O(n)	O(n)	O(n)	O(n)
Stack	O(n)	O(n)	O(1)	O(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Singly-Linked List	O(n)	O(n)	O(1)	O(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Doubly-Linked List	O(n)	O(n)	O(1)	O(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Skip List	O(n log(n))	O(n log(n))	O(n log(n))	O(n log(n))	O(n)	O(n)	O(n)	O(n)	O(n log(n))
Hash Table		O(1)	O(1)	O(1)		O(n)	O(n)	O(n)	O(n)

Binary Search Tree	O(n log(n))	O(n log(n))	O(n log(n))	O(n log(n))	O(n)	O(n)	O(n)	O(n)	O(n)
Cartesian Tree		O(n log(n))	O(n log(n))	O(n log(n))		O(n)	O(n)	O(n)	O(n)
B-Tree	O(n log(n))	O(n log(n))	O(n log(n))	O(n log(n))	O(n log(n))	O(n log(n))	O(n log(n))	O(n log(n))	O(n)
Red-Black Tree	O(n log(n))	O(n log(n))	O(n log(n))	O(n log(n))	O(n log(n))	O(n log(n))	O(n log(n))	O(n log(n))	O(n)
Splay Tree		O(n log(n))	O(n log(n))	O(n log(n))		O(n log(n))	O(n log(n))	O(n log(n))	O(n)
AVL Tree	O(n log(n))	O(n log(n))	O(n log(n))	O(n log(n))	O(n log(n))	O(n log(n))	O(n log(n))	O(n log(n))	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.

```
const employee =
{
    Fname: “John”;
    Lname: “Doe”;
    employeeID: “1234”;
    age: 35;
}
```

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?

A browser’s runtime environment

OR

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..

Angular
- Implemented to use for developing a Single Page Application (SPA)
- Extends HTML into the application and interprets the attributes to perform data binding

React
- Used to develop and operate the dyamic User Interface of webpages with high incoming traffic
- Makes using a virtual DOM and the integration of the same with any application straightforward

Vue
- Dual integration mode

	- Creating high-end SPA - Reliable platform for developing cross-platform
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.***	eBay - Front end development - Back end development Google - Front end development - Back end development - Web application development Walmart - Front end development - Back end development - Web application development - Mobile application development