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.
	Variable declaration is strongly typed.
	int myVariable = 12;
	string myString = “This is a string!”;
	bool isTrue = true;

Data Types *List all of the data types (and ranges) supported by this language.*	byte 0 to 255 sbyte -128 to 127 short -32,768 to 32,767 ushort 0 to 65,535 int -2,147,483,648 to 2,147,483,647 uint 0 to 4,294,967,295 long -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 ulong 0 to 18,446,744,073,709,551,615 float -3.402823e38 to 3.402823e38 double -1.79769313486232e308 to 1.79769313486232e308 decimal (+ or -)1.0 x 10e-28 to 7.9 x 10e28 char Any valid character, e.g. a,*, \x0058 (hex), or\u0058 (Unicode) bool True or False object Base type of all other types. String A sequence of Unicode characters DateTime 0:00:00am 1/1/01 to 11:59:59pm 12/31/9999

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

```
if (number> 5) {
    Console.WriteLine(“Number is greater than 5”);
}
```

If-Else

```
if (number> 5) {
    Console.WriteLine(“Number is greater thatn 5”);
}
Else {
    Console.WriteLine(“Number is not greater than 5”);
}
```

Else-If Ladder

```
If (number > 10) {
    Console.WriteLine(“Number is greater than 12”);
}
Else if (number > 5) {
    Console.WriteLine(“Number is greater than 5 but not 10”);
}
else {
    Console.WriteLine(“number is 5 or less”);
}
```

Switch Statement

```
switch (dayOfWeek) {
    Case 1:
        Console.WriteLine(“Sunday”);
        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.***

For Loop

```
for (int i = 0; i < 5; i++) {
    Console.WriteLine(“Iteration “ + i);
```

While Loop

```
while (i < 5) {
    Console.WriteLine(“Iteration “ + i);
    i++;
}
```

Do-While Loop

```
do  {
    Console.WriteLine(“Iteration “ + i);
    i++;
} while (i < 5);
```

Foreach Loop

```
foreach (int number in numbers) {
    Console.WriteLine(“Number: “ + number);
}
```

Nested Loops

```
for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 2; j++) {
        Console.WriteLine("i = " + i + ", j = " + j);
   }
}
```

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.) If the language supports declaring arrays in multiple ways, provide an example of way.*
	int[] numbers = { 1, 2, 3, 4, 5 };
	
	string[] names = new string[3];
	names[0] = “Sara”;
	names[1] = “Jim”;
	names[2] = “Matt”;

Data Structures *If this language provides a standard set of data structures, provide a list of the data structures and their Big-Oh complexity (identify what the complexity represents).*	Arrays: Access: O(1) Search (unsorted): O(n) Search (sorted with binary search): O(log n) Insertion (at the end): O(1) Insertion (at a specific index): O(n) Deletion (at the end): O(1) Deletion (at a specific index): O(n) Lists (List<T>): Access by index: O(1) Search: O(n) Insertion (at the end): O(1) Insertion (at a specific index): O(n) Deletion (at the end): O(1) Deletion (at a specific index): O(n) Dictionaries (Dictionary<TKey, TValue>): Access by key: O(1) on average (can degrade to O(n) in worst case) Insertion: O(1) on average (can degrade to O(n) in worst case) Deletion: O(1) on average (can degrade to O(n) in worst case) Sets (HashSet<T>): Add: O(1) on average (can degrade to O(n) in worst case) Remove: O(1) on average (can degrade to O(n) in worst case) Contains: O(1) on average (can degrade to O(n) in worst case) Queues (Queue<T>): Enqueue (add to the end): O(1) Dequeue (remove from the front): O(1) Peek (access the front element): O(1) Stacks (Stack<T>): Push (add to the top): O(1) Pop (remove from the top): O(1) Peek (access the top element): O(1)

	Linked Lists (LinkedList<T>): Access by index: O(n) Search: O(n) Insertion (at the end): O(1) Insertion (at a specific index): O(1) with reference to the location (O(n) for searching) Deletion (at the end): O(1) Deletion (at a specific index): O(1) with reference to the location (O(n) for searching) Sorted Sets (SortedSet<T>): Add: O(log n) Remove: O(log n) Contains: O(log n) Doubly Linked Lists (LinkedList<T>): Access by index: O(n) Search: O(n) Insertion (at the end): O(1) Insertion (at a specific index): O(1) with reference to the location (O(n) for searching) Deletion (at the end): O(1) Deletion (at a specific index): O(1) with reference to the location (O(n) for searching) HashTables (Hashtable, Dictionary, etc.): Access by key: O(1) on average (can degrade to O(n) in worst case) Insertion: O(1) on average (can degrade to O(n) in worst case) Deletion: O(1) on average (can degrade to O(n) in worst case)

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.
	class person {
	    public string Name {get; set; }
	    public int Age { get; set; }
	
	    public Person() {
	        Name = “Matt”;
	        Age = 26;
	}
	    public void Display() {
	        Console.WriteLine($”Name: {Name}, Age: {Age}”);
	    }
	} //end class

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? If so, what these other languages?*	C# compiles to the Common Language Runtime which is apart of the Microsoft.NET framework. Other languages that us the CLR framework are Visual Basic .NET, and F#
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.*	.NET Framework – Developing a wide variety of applications, including Windows desktop applications, web applications, cloud-based services, and cross-platform ap ASP.NET Core - Developing web applications and RESTful APIs that can run on Windows, macOS, or Linux. ASP.NET Core is known for its speed, scalability, and support for modern web development patterns. Entity Framework - Simplifying data access and database management in C# applications. Developers use Entity Framework to interact with databases, perform CRUD operations, and work with data using object-oriented paradigms.
Domains *What industries or domains use this programming language? Provide at least three 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.***	Microsoft – Microsoft is the developer of C# and is used in the development of Microsoft's own software products, including the Windows operating system, Office suite, and Azure cloud services. Additionally, C# plays a significant role in Microsoft's development tools and frameworks, such as Visual Studio and .NET. Unity Technologies - C# is a popular choice for game development, particularly in the Unity game engine. C# is used for scripting gameplay, creating game logic and building interactive experiences. Philips Healthcare - C# is employed in the healthcare sector for developing medical software, controlling medical devices, and managing patient data