Language Map for JavaScript

<table>
<tr>
<td>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.</td>
<td>C# is a strongly typed language, meaning that the variable needs to be declared at compile time and it cannot be assigned a value of a different data type without a type conversion.

int number = 2;
string message = “Hello, world!”;
bool isTrue = true;</td>
</tr>
<tr>
<td>Data Types
List all of the data types (and ranges) supported by this language.</td>
<td>
<table>
<tr><td>sbyte: -128 to 127</td><td>float: ±1.5 × 10^-45 to ±3.4 × 10^38</td></tr>
<tr><td>short: -32768 to 32767</td><td>double: ±5.0 × 10^-324 to ±1.7 × 10^308</td></tr>
<tr><td>int: -2^31 to 2^31-1</td><td>decimal: ±1.0 × 10^-28 to ±7.9228 × 10^28</td></tr>
<tr><td>long: -2^63 to 2^63-1</td><td>char: U +0000 to U +ffff</td></tr>
<tr><td>byte: 0 to 255</td><td>bool: true/false</td></tr>
<tr><td>ushort: 0 to 65535</td><td></td></tr>
<tr><td>uint: 0 to 2^32</td><td></td></tr>
<tr><td>ulong: 0 to 2^63</td><td></td></tr>
</table>
</td>
</tr>
<tr>
<td>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.</td>
<td><u>if statement:</u>
int number = 10;
if (number > 0)
{
Console.WriteLine("The number is positive.");
}

<u>if-else statement:</u>
int number = -5;
if (number > 0)
{
Console.WriteLine("The number is positive.");
}
else
{
Console.WriteLine("The number is non-positive.");
}

<u>else-if:</u>
int number = 0;
if (number > 0)
{
Console.WriteLine("The number is positive.");</td>
</tr>
</table>

```
}
else if (number < 0)
{
   Console.WriteLine("The number is negative.");
}
else
{
   Console.WriteLine("The number is zero.");
}
```

nested if statement:

```
int num1 = 10, num2 = 5;
if (num1 > 0)
{
   if (num2 > 0)
   {
      Console.WriteLine("Both numbers are positive.");
   }
   else
   {
      Console.WriteLine("First number is positive, but second number is non-positive.");
   }
}
else
{
   Console.WriteLine("First number is non-positive.");
}
```

switch statement:

```
int day = 3;
switch (day)
{
   case 1:
      Console.WriteLine("Monday");
      break;
   case 2:
      Console.WriteLine("Tuesday");
      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 = 1; i <= 5; i++)
	{
	  Console.WriteLine("Iteration: " + i);
	}
	
	while loop
	int i = 1;
	while (i <= 5)
	{
	  Console.WriteLine("Iteration: " + i);
	  i++;
	}
	
	do-while loop
	int i = 1;
	do
	{
	  Console.WriteLine("Iteration: " + i);
	  i++;
	} while (i <= 5);
	
	foreach loop
	int[] numbers = { 1, 2, 3, 4, 5 };
	foreach (int number in numbers)
	{
	  Console.WriteLine("Number: " + number);
	}

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.*	**One-dimensional arrays** int[] intArray = { 1, 2, 3, 4, 5 }; double[] doubleArray = {2.4, 3.5, 8.21}; int[] intArray2 = new int[5]; **two-dimensional arrays** int[,] intArray = { { 1, 2 }, { 3, 4 }, { 5, 6 } };

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:** O(1) for access, deletion, insertion. O(n) for searching **Lists:** O(1) for access, and deletion/insertion at the end. O(n) for insertion/deletion in the middle and searching. **Dictionaries:** O(1) for insertion/deletion and access. **Sets:** O(1) for insertion, deletion, search. **Queues:** O(1) for enqueue and dequeue. **Stacks:** O(1) for push and pop. **Linked list:** O(n) for access, search, and insertion/deletion in middle. O(1) for insertion at beginning. **Trees:** O(log n) for all **Graphs:**

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 = "John Doe";
	Age = 30;
	}
	
	public void DisplayInfo()
	{
	System.Console.WriteLine($"Name: {Name}, Age: {Age}");
	}
	}
	
	class Program
	{
	static void Main()
	{
	Person person1 = new Person();
	person1.DisplayInfo();
	}
	}

Runtime Environment *What runtime environment does this language compile to? For example, Java compiles to the Java Virtual Machine.*	C# is complied into the IL(intermediate language) Some other languages also compile to IL such as F# and virtual basic.

Do other languages also compile to this runtime? If so, what these other languages?	
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.*	Entity Framework is popular in C# and is used to simplify database interaction by providing a convenient way to perform operations without writing raw SQL queries. ASP.Net core is another framework popular in C# that is used for web based applications and RESTful APIs ML.Net is a popular framework for building custom machine learning models
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 and many of the popular gaming companies use C# for game development. Siemens uses C# to create healthcare software such as medical imaging applications. JPMorgan uses C# to create trading platforms.