

Language Map for JavaScript

Variable Declaration <i>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.</i>	It is a strongly typed language. Every variable and constant has a type, and so does every expression that evaluates a value. string name = "Ben"; int x = 10; bool myBool = true;
Data Types <i>List all of the data types (and ranges) supported by this language.</i>	byte 0 to 255 sbyte -128 to 127 short -32768 to 32767 ushort 0 to 65535 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 string DateTime 0:00:00am 1/1/01 to 11:59:59pm 12/31/9999
Selection Structures <i>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.</i>	if if (value < 0 value > 100) { Console.WriteLine("Warning: not acceptable value! "); } if else if (tempInCelsius < 20.0) { Console.WriteLine("Cold."); } else { Console.WriteLine("Perfect!"); } switch switch (measurement)

	<pre> { case < 0.0: Console.WriteLine(\$"Measured value is {measurement}; too low."); break; default: Console.WriteLine(\$"Measured value is {measurement}."); break; } </pre>
<p>Repetition Structures</p> <p><i>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.</i></p>	<pre> while int number = 0; while(number < 5) { Console.WriteLine(number); number = number + 1; } do int number = 0; do { Console.WriteLine(number); number = number + 1; } while(number < 5); for for(int i = 0; i < number; i++) Console.WriteLine(i); foreach ArrayList list = new ArrayList(); list.Add("John Doe"); list.Add("Jane Doe"); list.Add("Someone Else"); foreach(string name in list) Console.WriteLine(name); </pre>

<p>Arrays <i>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.</i></p>	<pre>string[] names = {"Ben", "Steve", "Bob", "Jeff"}; int[] numbers = new int[4] {5, 3, 21, 76};</pre>
<p>Data Structures <i>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).</i></p>	<p>Array – accessing a specified index is $O(1)$ since it does not matter what size the array is. Adding to the end of an array is $O(1)$ unless the array is full, then it's $O(n)$. Searching the array is $O(n)$ since the complexity grows with the size of the array.</p> <p>Stack – push and pop are $O(1)$ since the size of the stack does not matter. Popping all elements is $O(n)$ since it depends on the size of the stack.</p> <p>Queue – enqueue is $O(n)$ since the queue is traversed until it reaches the end. Dequeue is $O(1)$ since it always removes the item from the front of the list.</p> <p>Hashtable – assuming no collisions it is $O(1)$ since the hashkey will point directly to the index in the array regardless of size.</p> <p>Dictionary - $O(1)$, because the Dictionary class is implemented as a hash table.</p> <p>Linked List - When accessing elements of a linked list, speed is proportional to size of the list with Big $O(n)$. Since we must traverse the entire list in order to get to the desired element.</p>
<p>Objects <i>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.</i></p>	<pre>class Car { string color = "blue"; } static void Main(string[] args) { Car myObj = new Car(); }</pre>

<p>Runtime Environment <i>What runtime environment does this language compile to? For example, Java compiles to the Java Virtual Machine.</i> <i>Do other languages also compile to this runtime? If so, what these other languages?</i></p>	<p>C# is compiled into a .NET common language runtime routine. Managed C++, Visual Basic, and J# do as well.</p>
<p>Libraries/Frameworks <i>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.</i></p>	<p>Entity Framework Core - an object-relational mapping (ORM) framework that allows developers to work with databases using C# objects, instead of SQL queries. It supports various database providers, including SQL Server, MySQL, and PostgreSQL.</p> <p>ASP.NET. Core - a web framework that builds on top of .NET Core, providing a set of features for building web applications and APIs. It includes a rich middleware pipeline, routing, authentication, and dependency injection.</p> <p>AutoMapper - A mapping library that simplifies the process of mapping objects between different types. It eliminates the need for writing repetitive mapping code and reduces the risk of errors.</p>
<p>Domains <i>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.</i></p>	<p>Microsoft uses C# for web and game development. City National Bank uses C# for building cloud-based applications. Trustpilot uses C# for web services and app development.</p>