Scope

# Ámbito

*Scope* en un lenguaje de programación controla la visibilidad y el tiempo de vida de las variables y los parámetros. Es un servicio importante para el programador porque reduce las colisiones de nombres y proporciona una gestión automática de la memoria:

var foo = function ( ) {  
 var a = 3, b = 5;  
  
 var bar = function ( ) {  
 var b = 7, c = 11;  
  
// At this point, a is 3, b is 7, and c is 11  
  
 a += b + c;  
  
// At this point, a is 21, b is 7, and c is 11  
  
 };  
  
// At this point, a is 3, b is 5, and c is not defined  
  
 bar( );  
  
// At this point, a is 21, b is 5  
  
};

La mayoría de los lenguajes con sintaxis C tienen ámbito de bloque. Todas las variables definidas en un bloque (una lista de sentencias entre llaves) no son visibles desde fuera del bloque. Las variables definidas en un bloque pueden liberarse cuando finaliza la ejecución del bloque. Esto es bueno.

Por desgracia, JavaScript no tiene ámbito de bloque, aunque su sintaxis de bloque sugiere que sí lo tiene. Esta confusión puede ser fuente de errores.

JavaScript sí tiene ámbito de función. Esto significa que los parámetros y variables definidos en una función no son visibles fuera de la función, y que una variable definida en cualquier lugar de una función es visible en cualquier lugar de la función.

En muchos lenguajes modernos, se recomienda declarar las variables lo más tarde posible, en el primer punto de uso. Eso resulta ser un mal consejo para JavaScript, porque carece de ámbito de bloque. Por tanto, lo mejor es declarar todas las variables utilizadas en una función al principio del cuerpo de la función.