Invocation

## El patrón de invocación de funciones

Cuando una función no es propiedad de un objeto, se invoca como una función:

var sum = add(3, 4); // sum is 7

Cuando se invoca una función con este patrón, this se vincula al objeto global. Esto fue un error en el diseño del lenguaje. Si el lenguaje se hubiera diseñado correctamente, al invocar la función interna, this seguiría estando ligada a la variable this de la función externa. Una consecuencia de este error es que un método no puede emplear una función interna para que le ayude a hacer su trabajo, porque la función interna no comparte el acceso del método al objeto, ya que su this está ligado al valor incorrecto. Afortunadamente, existe una solución fácil. Si el método define una variable y le asigna el valor de this, la función interna tendrá acceso a this a través de esa variable. Por convención, el nombre de esa variable es that:

// Augment myObject with a double method.  
  
myObject.double = function ( ) {  
 var that = this; // Workaround.  
  
 var helper = function ( ) {  
 that.value = add(that.value, that.value);  
 };  
  
 helper( ); // Invoke helper as a function.  
};  
  
// Invoke double as a method.  
  
myObject.double( );  
document.writeln(myObject.value); // 6

[anterior](ch04s03_2.html)[Subtema 3 de 5: (Ver todo)](ch04s03.html)[siguiente](ch04s03_4.html)