

4.A. Estructuras definidas por el usuario en JavaScript

2.- Creación de funciones.

2.2.- Ámbito de las variables.

Ha llegado la hora de distinguir entre las variables que se definen fuera de una función, y las que se definen dentro de las funciones.

Las variables que se definen fuera de las funciones se llaman **variables globales**. Las **variables** que se definen dentro de las funciones, con la palabra reservada **var**, se llaman variables locales.

Una **variable global** en JavaScript tiene una connotación un poco diferente, comparando con otros lenguajes de programación. Para un script de JavaScript, el alcance de una variable global, se limita al documento actual que está cargado en la ventana del navegador o en un frame. Sin embargo cuando inicializas una variable como variable global, quiere decir que todas las instrucciones de tu script (incluidas las instrucciones que están dentro de las funciones), tendrán acceso directo al valor de esa variable. Todas las instrucciones podrán leer y modificar el valor de esa variable global.

En el momento que una página se cierra, todas las variables definidas en esa página se eliminarán de la memoria para siempre. Si necesitas que el valor de una variable persista de una página a otra, tendrás que utilizar técnicas que te permitan almacenar esa variable (como las cookies, o bien poner esa variable en el documento frameset, etc.).

Aunque el uso de la palabra reservada **var**, para inicializar variables es opcional, te recomiendo que la uses ya que así te protegerás de futuros cambios en las próximas versiones de JavaScript.

En contraste a las variables globales, una **variable local será definida dentro de una función**. Antes viste que podemos definir variables en los parámetros de una función (sin usar **var**), pero también podrás definir nuevas variables dentro del código de la función. En este caso, **si que se requiere el uso de la palabra reservada var cuando definimos una variable local**, ya que de otro modo, esta variable será reconocida como una variable global.

El alcance de una variable local está solamente dentro del ámbito de la función. Ninguna otra función o instrucciones fuera de la función podrán acceder al valor de esa variable.

Reutilizar el nombre de una variable global como local es uno de los bugs más sutiles y por consiguiente más difíciles de encontrar en el código de JavaScript. La variable local en momentos puntuales ocultará el valor de la variable global, sin avisarnos de ello. Como recomendación, no reutilices un nombre de variable global como local en una función, y tampoco declares una variable global dentro de una función, ya que podrás crear fallos que te resultarán difíciles de solucionar.

Ejemplo de variables locales y globales:

```
// Uso de variables locales y globales no muy recomendable, ya que estamos empleando el mismo
// nombre de variable en global y en local.

var chica = "Aurora";           // variable global

var perros = "Lucky, Samba y Ronda"; // variable global

function demo()
{
    // Definimos una variable local (fíjate que es obligatorio para las variables locales usar
    // var)

    // Esta variable local tendrá el mismo nombre que otra variable global pero con distinto
    // contenido.

    // Si no usáramos var estaríamos modificando la variable global chica.

    var chica = "Raquel";        // variable local

    document.write( "<br/>" + perros + " no pertenecen a " + chica + ".");
}
```

```
// Llamamos a la función para que use las variables locales.  
  
demo();  
  
// Utilizamos las variables globales definidas al comienzo.  
  
document.write( "<br/>" + perros + " pertenecen a " + chica + ".");
```

Como resultado obtenemos:

Lacky, Samba y Ronda no pertenecen a Raquel.

Lacky, Samba y Ronda pertenecen a Aurora.

Créditos de la imagen

Autoría: leonardoparada.

Licencia: CC BY 2.0.

◀ Solución a la tarea para DWEC04

Ir a...

4.B. Objetos JavaScript en W3Schools ►