

Ámbito de las Variables

JavaScript

Ámbito de las Variables

El ámbito de una variable (llamado "*scope*" en inglés) es la zona del programa en la que se define la variable. **JavaScript** define dos ámbitos para las variables: **global** y **local**.

El siguiente ejemplo ilustra el comportamiento de los ámbitos:

```
function creaMensaje() {  
  var mensaje = "Mensaje de prueba";  
}  
  
creaMensaje();  
alert(mensaje);
```




Ámbito de las Variables

El ejemplo anterior define en primer lugar una función llamada **creaMensaje** que crea una variable llamada mensaje.

A continuación, se ejecuta la función mediante la llamada **creaMensaje()**; y seguidamente, se muestra mediante la función **alert()** el valor de una variable llamada mensaje.

Sin embargo, al ejecutar el código anterior no se muestra ningún mensaje por pantalla. La razón es que la variable **mensaje** se ha definido dentro de la función **creaMensaje()** y por tanto, es una **variable local** que solamente está definida dentro de la función.



Ámbito de las Variables

Cualquier instrucción que se encuentre dentro de la función puede hacer uso de esa variable, pero todas las instrucciones que se encuentren en otras funciones o fuera de cualquier función no tendrán definida la variable mensaje.

De esta forma, para mostrar el mensaje en el código anterior, la función **alert()** debe llamarse desde dentro de la función **creaMensaje()**:

```
function creaMensaje() {  
  var mensaje = "Mensaje de prueba";  
  alert(mensaje);  
}  
creaMensaje();
```



Ámbito de las Variables

Además de variables locales, también existe el concepto de **variable global**, que está definida en cualquier punto del programa (incluso dentro de cualquier función).

```
var mensaje = "Mensaje de prueba";  
function muestraMensaje() {  
  alert(mensaje);  
}
```




Ámbito de las Variables

El código anterior es el ejemplo inverso al mostrado anteriormente. Dentro de la función **muestraMensaje()** se quiere hacer uso de una variable llamada mensaje y que no ha sido definida dentro de la propia función.

Sin embargo, si se ejecuta el código anterior, sí que se muestra el mensaje definido por la variable mensaje.

El motivo es que en el código **JavaScript** anterior, la variable mensaje se ha definido fuera de cualquier función. Este tipo de variables automáticamente se transforman en variables globales y están disponibles en cualquier punto del programa (incluso dentro de cualquier función).




Ámbito de las Variables

De esta forma, aunque en el interior de la función no se ha definido ninguna variable llamada mensaje, la *variable global* creada anteriormente permite que la instrucción **alert()** dentro de la función muestre el mensaje correctamente.

Si una variable se declara fuera de cualquier función, automáticamente se transforma en *variable global* independientemente de si se define utilizando la palabra reservada **var** o no. Sin embargo, las variables definidas dentro de una función pueden ser *globales o locales*.

Si en el interior de una función, las variables se declaran mediante **var** se consideran *locales* y las variables que no se han declarado mediante **var**, se transforman automáticamente en variables *globales*.



Ámbito de las Variables

Por lo tanto, se puede rehacer el código del primer ejemplo para que muestre el mensaje correctamente. Para ello, simplemente se debe definir la variable dentro de la función sin la palabra reservada **var**, para que se transforme en una variable global:

```
function creaMensaje() {  
    mensaje = "Mensaje de prueba";  
}  
  
creaMensaje();  
alert(mensaje);
```



Ámbito de las Variables

¿Qué sucede si una función define una variable local con el mismo nombre que una variable global que ya existe? En este caso, las variables *locales* prevalecen sobre las *globales*, pero sólo dentro de la función:

```
var mensaje = "gana la de fuera";  
function muestraMensaje() {  
  var mensaje = "gana la de dentro";  
  alert(mensaje);  
}  
alert(mensaje);  
muestraMensaje();  
alert(mensaje);
```



Ámbito de las Variables

El código anterior muestra por pantalla los siguientes mensajes:

gana la de fuera

gana la de dentro

gana la de fuera

Dentro de la función, la variable *local* llamada mensaje tiene más prioridad que la variable *global* del mismo nombre, pero solamente dentro de la función.



Ámbito de las Variables

¿Qué sucede si dentro de una función se define una variable global con el mismo nombre que otra variable global que ya existe? En este otro caso, la variable global definida dentro de la función simplemente modifica el valor de la variable global definida anteriormente:

```
var mensaje = "gana la de fuera";  
function muestraMensaje() {  
  mensaje = "gana la de dentro";  
  alert(mensaje);  
}  
alert(mensaje);  
muestraMensaje();  
alert(mensaje);
```



Ámbito de las Variables

En este caso, los mensajes mostrados son:

gana la de fuera

gana la de dentro

gana la de dentro

La recomendación general es definir como ***variables locales*** todas las variables que sean de uso exclusivo para realizar las tareas encargadas a cada función. Las ***variables globales*** se utilizan para compartir variables entre funciones de forma sencilla.

