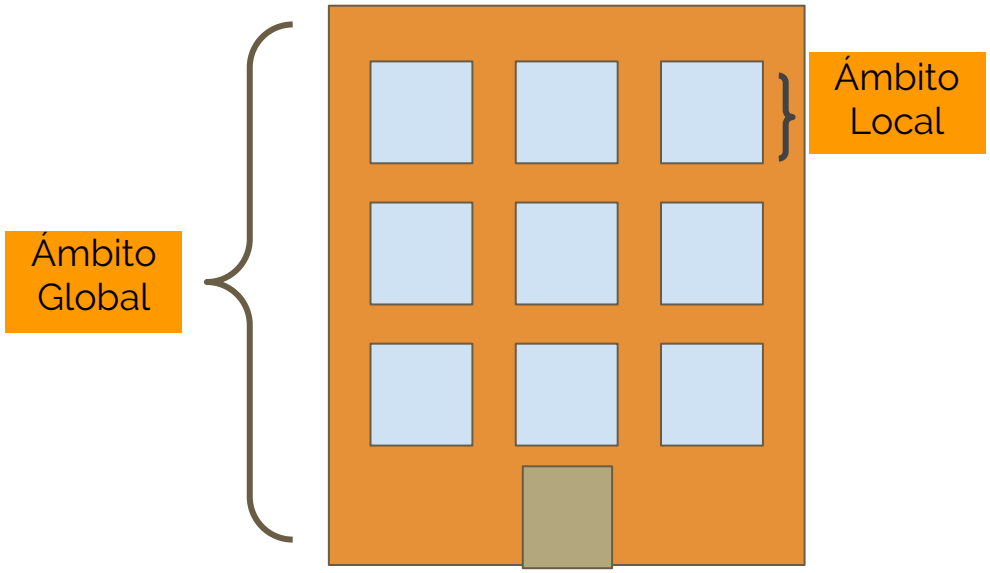
Lección 21

Scope

¿Qué es el scope?

¿Qué es el Scope?

Hablamos de scope cuando nos referimos al **contexto** en el cual una variable existe. El scope (también llamado *ámbito* o *alcance*) nos indica si tenemos acceso a una variable y/o en qué partes del programa podemos acceder a la misma.

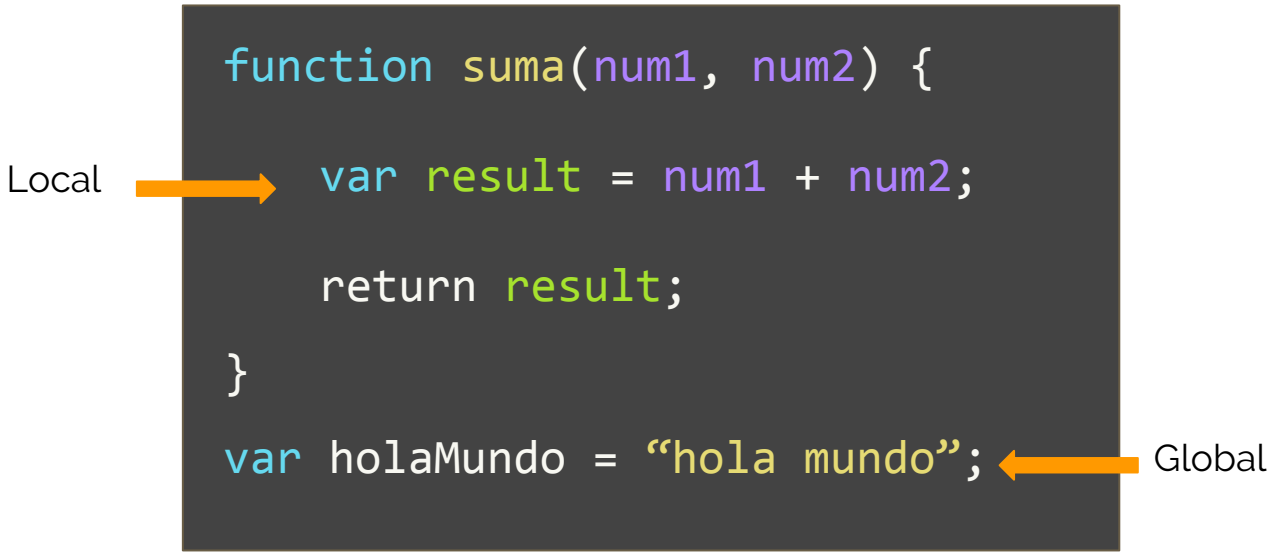


Tipos de Scope

En JavaScript existen dos tipos de scope:

Local: para las variables declaradas **dentro** de una función.

Global: para las variables declaradas **fuera** de una función.



Scope Global

Scope Global

Las variables declaradas en global scope son las que se definen y que pueden ser accedidas desde cualquier punto del programa. Toda variable declarada fuera de una función tiene un scope global.

La ventaja que tienen es que se pueden declarar de manera fácil y podemos compartir datos con diferentes componentes.

Scope Global

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.

Scope Global: Precaución.

Cuidado con los problemas: Un problema sería pensar que una variable es local cuando realmente es global: Esto puede ocurrir cuando declaramos una variable local pero olvidamos la palabra reservada `var`, en vez de que haya un error de variable no declarada como en otros lenguajes, esa variable se convierte en global sin darnos cuenta.

Scope Local

Scope Local

Este tipo de scope aplica para las variables declaradas dentro de una función y serán accesibles durante toda la ejecución de dicha función.

Cuando se declaren variables locales sólo podremos acceder a ellas dentro del lugar donde se ha declarado, es decir, si la habíamos declarado en una función solo podremos acceder a ella cuando estemos en esa función.

Scope Local

Al repetirse el nombre de una variable (entre una variable global y una local), dentro de una función la variable local tiene más prioridad que la variable global, pero sólo dentro de esta función.

¿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.

Scope: Recomendaciones.

La recomendación es definir como variables locales todas las variables que sean de uso exclusivo para realizar las tareas encargadas a cada función y definir como globales las que se utilizarán para compartir variables entre funciones.

Scope: Recomendaciones.

Conocer los tipos de scope y sus reglas nos permitirá detectar errores de una manera más efectiva.

Como resumen podemos decir:

-Una variable declarada dentro de una función es local.

-Una variable declarada fuera de una función es global.

-Nunca olvidar la palabra reservada `var`.

-Debemos preferir las variables locales sobre las globales.