

Evitar el uso de variables globales

Cuando se hace uso de una variable global, normalmente es porque va a ser accedida por varias funciones.

Al declararse en un ámbito global, las funciones del programa pueden acceder a la variable global sin tener que pasarla como parámetro a dichas funciones.

El uso de variables globales no es aconsejable, a pesar de que aparentemente nos parezca muy útil. Esto se debe a varias razones fundamentales:

- Impiden predecir su valor en cada punto de ejecución del programa (ya que puede modificarse en cualquier punto del código).
- Existe la posibilidad de introducir errores con declaraciones locales de la misma variable.
- Ocuparán espacio hasta el final de la ejecución del programa, momento en el que son destruidas.
- El uso indiscriminado de variables globales produce efectos colaterales. Esto sucede cuando existe una alteración no deseada del contenido de una variable global dentro de una función, bien por invocación, bien por olvidar definir en la función una variable local o un parámetro formal con ese nombre. La corrección de dichos errores puede ser muy ardua.
- Atenta contra uno de los principios de la programación, la **modularidad**. El bajo acoplamiento supone no compartir espacios de memoria con otras funciones, y potenciar el paso de información (llamadas) para que la función trate la información localmente.
- Dificultad de realizar pruebas y detección de errores, porque hay que "rastrear" qué funciones las utilizan.

¿Cuándo es aceptable usarlas?

- 1.- Cuando las usamos como constantes (identificador en MAYÚSCULAS, para detectarla rápidamente).
- 2.- Cuando su uso reduce drásticamente la complejidad del código (caso muy difícil de encontrar).

REGLA GENERAL: Las variables deben tener siempre el menor alcance posible.