

Questionario Variables

1- ¿Qué es una variable en JavaScript?

Es un espacio en la memoria del programa que se asigna para guardar un valor específico, como un número, texto, un objeto o una función.

2- ¿Qué diferencia hay entre `var`, `let` y `const`?

Las diferencias entre `var`, `let`, y `const` radican en los siguientes items:

Scope (ámbito), en el caso de `var`, este tiene un tipo de scope por función, mientras que el de `let` y `const` es por bloque.

La posibilidad de reasignación, `var` y `let` podrán ser reasignada dentro de su ámbito, mientras que `const` no podrá ser reasignada.

La posibilidad de redeclaración, `var` podrá ser redeclarada de forma libre, mientras que `let` y `const` no podrán ser redeclaradas nunca dentro de un mismo bloque.

Hoisting, `var` podrá ser llamada o reasignada tanto como antes o después de la declaración de esta. En el caso de `let` y `const` se generará un Temporal Dead Zone (TDZ) y no se podrán acceder a ellas hasta luego de su declaración.

3- ¿Cuál es el ámbito de una variable declarada con `var`?

Funcion.

4- ¿Cuál es el ámbito de una variable declarada con `let`?

Bloque.

5- ¿Cuál es el ámbito de una variable declarada con `const`?

Bloque.

6- ¿Qué sucede si intentas redeclarar una variable con `var` en el mismo ámbito?

Será posible y se guardará en la variable la última declaración.

7- ¿Qué sucede si intentas redeclarar una variable con `let` en el mismo ámbito?

No será posible, se generará un error.

8- ¿Qué sucede si intentas cambiar el valor de una constante declarada con `const`?

No será posible, se generará un error.

9- ¿Es posible declarar una variable sin inicializar? Si es así, ¿qué valor tiene por defecto?

Es posible declarar una variable sin inicializar cuando son `var` o `let`, estas por defecto se les asignará el valor default. En el caso de una variable `const` no será posible.

10- ¿Qué tipo de error se genera al intentar usar una variable no declarada?

Se genera un error conocido como `ReferenceError`.

11- ¿Qué diferencia hay entre declarar una variable e inicializarla?

Declarar una variable implica definir su existencia y reservar un espacio en la memoria con un nombre específico. Inicializar una variable implica asignarle un valor por primera vez después de haber sido declarada.

12- ¿Qué es el hoisting en JavaScript?

El hoisting es un comportamiento donde las declaraciones de variables y funciones son movidas al principio de su ámbito actual durante la fase de compilación. Es decir que, aunque declares una variable o función más adelante en tu código las trata como si hubieran sido declaradas al inicio. En el caso de `let` y `const` las define como no inicializadas.

13- ¿Cómo afecta el hoisting a las variables declaradas con `var`?

Las inicializa con "undefined" al principio del código.

14- ¿Qué sucede si intentas acceder a una variable antes de declararla con `let` o `const`?

Se generará el error `ReferenceError`.

15- ¿Qué es una variable global?

Una variable global es aquella que se declara fuera de cualquier función o bloque de código. Esto significa que está disponible en todo el ámbito del programa.

16- ¿Es posible cambiar el tipo de datos de una variable en JavaScript? Da un ejemplo.

Puede cambiar dinámicamente debido a su naturaleza de lenguaje de tipado dinámico. Esto significa que una variable puede contener diferentes tipos de datos en diferentes momentos durante la ejecución del programa.

```
let numero = 18;  
let texto = años;  
let edad = numero + texto; // tipo string
```

17- ¿Cómo se puede verificar el tipo de una variable en JavaScript?

```
let booleano = numero >= 18;  
  
console.log(typeof numero); //tipo number  
console.log(typeof edad); //tipo string  
console.log(typeof booleano); //tipo boolean
```