# Variables en JavaScript: Guía de Partes Importantes

## 1. Declarar variables (var, let, const)

**Explicación:** Puedes declarar variables con var, let o const.

**Ejemplo:**

var x = 10;  
let y = 20;  
const z = 30;

## 2. Diferencias entre var, let y const

* var: Declaración tradicional, alcance de función, permite redeclarar y reasignar.
* let: Alcance de bloque, permite reasignar pero no redeclarar en el mismo bloque.
* const: Alcance de bloque, no permite reasignar ni redeclarar.

**Ejemplo:**

var nombre = "Ana";  
let edad = 25;  
const PI = 3.1416;

## 3. Ámbito de las variables (scope)

**Explicación:** Determina desde dónde se puede acceder a una variable.

**Ejemplo:**

function prueba() {  
 var a = 1;  
 let b = 2;  
 if(true) {  
 let c = 3;  
 }  
 // 'a' está disponible aquí, 'b' también, 'c' no  
}

## 4. Variables globales y locales

**Explicación:**

* Global: Declarada fuera de cualquier función/bloque.
* Local: Declarada dentro de una función o bloque.

**Ejemplo:**

let global = "afuera";  
function test() {  
 let local = "adentro";  
}

## 5. Inicialización de variables

**Explicación:** Asignar un valor inicial al declarar la variable.

**Ejemplo:**

let a = 5; // Inicializada  
let b; // No inicializada

## 6. Reasignación de valores

**Explicación:** Cambiar el valor de una variable (menos const).

**Ejemplo:**

let a = 10;  
a = 20; // Reasignación

## 7. Hoisting de variables

**Explicación:**

* var es elevada (hoisting) y su valor es undefined antes de la asignación.
* let y const también se elevan, pero no se pueden usar antes de su declaración.

**Ejemplo:**

console.log(a); // undefined  
var a = 5;  
//console.log(b); // Error  
let b = 10;

## 8. Constantes y mutabilidad

**Explicación:**

* const no permite cambiar la referencia, pero sí modificar objetos o arrays a los que apunta.

**Ejemplo:**

const persona = {nombre: "Ana"};  
persona.nombre = "Juan"; // Permitido  
//persona = {}; // Error

## 9. Nombres válidos para variables (naming conventions)

**Explicación:**

* Letras, dígitos (no al inicio), $, \_
* Sensibles a mayúsculas y minúsculas
* Convención: camelCase

**Ejemplo:**

let edadUsuario;  
let $total;  
let \_contador;

## 10. Tipos de datos primitivos almacenados en variables

**Explicación:**

* number, string, boolean, undefined, null, symbol, bigint

**Ejemplo:**

let n = 5; // number  
let texto = "Hola"; // string  
let activo = true; // boolean  
let nada = null; // null  
let noDefinido; // undefined

## 11. Declarar múltiples variables en una sola línea

**Ejemplo:**

let a = 1, b = 2, c = 3;

## 12. Variables no inicializadas (undefined)

**Explicación:** Si una variable se declara pero no se inicializa, su valor es undefined.

**Ejemplo:**

let x;  
console.log(x); // undefined

## 13. Shadowing de variables

**Explicación:** Una variable local puede “ocultar” una variable global con el mismo nombre.

**Ejemplo:**

let x = 1;  
function test() {  
 let x = 2; // Oculta la global  
 console.log(x); // 2  
}  
test();  
console.log(x); // 1

## 14. Acceso y modificación del valor de una variable

**Explicación:** Acceder es simplemente usar su nombre, modificar es reasignar.

**Ejemplo:**

let color = "azul";  
color = "rojo"; // Modificación  
console.log(color);