## **Tipos de Datos Primitivos en JavaScript**

Los tipos de datos primitivos son los más básicos y no son objetos. Estos incluyen:

* **Number:** Representa números tanto enteros como decimales.
* Ejemplo: let edad = 30;, let pi = 3.14;
* **String:** Representa cadenas de texto.
* Ejemplo: let nombre = "Juan Pérez";, let saludo = 'Hola, mundo!';
* **Boolean:** Representa valores lógicos, verdadero (true) o falso (false).
* Ejemplo: let esMayorDeEdad = true;, let estaLloviendo = false;
* **Null:** Representa la ausencia intencional de cualquier valor.
* Ejemplo: let valorNulo = null;
* **Undefined:** Indica que una variable ha sido declarada pero aún no se le ha asignado un valor.
* Ejemplo: let variableSinValor; // variableSinValor tiene el valor undefined

## **Tipos de Datos Complejos en JavaScript**

Los tipos de datos complejos son objetos. Estos incluyen:

* **Object:** Es la base para crear objetos personalizados.
* Ejemplo:
* JavaScript
* let persona = { nombre: "Ana", edad: 25, ciudad: "Madrid" };
* Use code [with caution.](file:///C:/faq#coding)
* **Array:** Es una colección ordenada de elementos.
* Ejemplo: let frutas = ["manzana", "banana", "pera"];
* **Function:** Es un bloque de código reutilizable.
* Ejemplo:
* JavaScript
* function saludar(nombre) { console.log("Hola, " + nombre + "!"); }
* Use code [with caution.](file:///C:/faq#coding)

## **Cómo determinar el tipo de una variable**

Para conocer el tipo de una variable, puedes utilizar el operador typeof:

JavaScript

console.log(typeof edad); // Output: "number"console.log(typeof nombre); // Output: "string"console.log(typeof persona); // Output: "object"

Use code [with caution.](file:///C:/faq#coding)

## **¿Por qué son importantes los tipos de datos?**

* **Operaciones:** Cada tipo de dato tiene operaciones específicas. Por ejemplo, puedes sumar dos números, pero no puedes sumar un número y una cadena.
* **Comparaciones:** Al comparar variables, JavaScript verifica si son del mismo tipo y si sus valores son iguales.
* **Conversiones de tipo:** A veces es necesario convertir un tipo de dato a otro. Por ejemplo, puedes convertir una cadena numérica a un número utilizando parseInt() o parseFloat().

## **Ejemplo práctico:**

JavaScript

let nombre = "Carlos"; let edad = 32; let esEstudiante = false; console.log("Hola, mi nombre es " + nombre + " y tengo " + edad + " años."); console.log("¿Soy estudiante? " + esEstudiante); let calificaciones = [9, 8, 7]; console.log("Mis calificaciones son: " + calificaciones);

Use code [with caution.](file:///C:/faq#coding)

**En resumen:**

Los tipos de datos en JavaScript son fundamentales para entender cómo funciona el lenguaje. Al conocer los diferentes tipos y cómo utilizarlos, podrás escribir código más eficiente y evitar errores comunes.