

# Los tipos de datos

# Índice

1. [Tipos de datos](#)
2. [Tipos de datos especiales](#)

# 1 | Tipos de datos

“

Los **tipos de datos** le **permiten** a JavaScript **conocer** las **características** y **funcionalidades** que estarán disponibles para ese **dato**.

”



## Numéricos (number)

```
{  
  let edad = 35; // número entero  
  let precio = 150.65; // decimales  
}
```



Como JavaScript está escrito en inglés usaremos un punto para separar los decimales.

## Cadenas de caracteres (string)

```
{  
  let nombre = 'Mamá Luchetti'; // comillas simples  
  let ocupacion = "Master of the sopas"; // comillas  
  dobles tienen el mismo resultado  
}
```

## Lógicos o booleanos (boolean)

```
{  
  let laCharlaEstaReCopada = true;  
  let hayAsadoAlFinal = false;  
}
```

## Objetos (object)

A diferencia de otros tipos de datos que pueden contener un solo dato, los objetos son **colecciones** de datos y en su interior pueden existir todos los anteriores.

Los podemos reconocer porque se declaran con llaves `{ }`.

```
{ }  
  
let persona = {  
  nombre: 'Javier', // string  
  edad: 34, // number  
  soltero: true // boolean  
}
```

## Array

Al igual que los objetos, los arrays son colecciones de datos. Los podemos reconocer porque se declaran con corchetes [ ].

Los arrays son un tipo especial de objetos, por eso **no los consideramos como un tipo de dato más**.

Los mencionamos de manera especial porque son muy comunes en todo tipo de código.

```
{  
  let comidasFavoritas = ['Milanesa napolitana',  
    'Ravioles con bolognesa', 'Pizza calabresa'];  
  
  let numerosSorteados = [12, 45, 56, 324, 452];  
}
```

# 2 | Tipos de datos especiales



“

Los **tipos de datos especiales** le permiten a JavaScript determinar **estados especiales** que pueden tener los **datos**.



”

## NaN (Not a Number)

Indica que el valor no puede ser parseado como un número

```
{ } let malaDivision = "35" / 2; // NaN no es un número
```

## Null (valor nulo)

Lo asignamos nosotros para indicar un valor vacío o desconocido.

```
{ } let temperatura = null; // No llegó un dato, algo falló
```

## Undefined (valor sin definir)

Indica la ausencia de valor.

Las variables tienen un valor indefinido hasta que les asignamos uno.

```
{}  
let saludo; // undefined, no tiene valor  
saludo = "¡Hola!"; // Ahora si tiene un valor
```



Los **comentarios** son partes de nuestro código que **no se ejecutan**.

Siempre comienzan con dos barras inclinadas //

Los usamos para explicar lo que estamos haciendo y **dejar información útil** para nuestro equipo o para nuestro yo del futuro.



```
// Math.round() retorna el valor redondeado al entero  
más cercano.
```

```
let redondeado = Math.round(20.49);
```

DigitalHouse>  
Coding School