

LOS TIPOS DE DATOS

“

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



NUMÉRICOS (number)

```
{}
```

```
var edad = 35; // número entero  
var precio = 150.65; // con decimales
```



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

CADENAS DE CARACTERES (string)

```
{}
```

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

LÓGICOS O BOOLEANOS (boolean)

```
{}
```

```
var laCharlaEstaReCopada = true;  
var hayAsadoAlFinal = false;
```

OBJETO (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 { }.

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

{ }

OBJETO (object)

Para acceder a un elemento dentro de un objeto, utilizamos la notación de punto. Es decir el nombre del objeto y el nombre de la propiedad separado con un punto.

```
var persona = {  
  nombre = 'Teresa', // string  
  edad = 23 // number  
}
```

```
persona.nombre // Teresa
```

```
persona.edad // 23
```

{}

ARRAY (object)

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.

```
var comidasFavoritas = ['Milanesa napolitana',  
  'Ravioles con bolognesa', 'Pizza calabresa'];
```

```
let numerosSorteados = [12, 45, 56, 324, 452];
```



ARRAY (object)

Para acceder a un elemento dentro de un array, utilizamos el operador de índice (que se escribe con corchetes []) . Es decir el nombre del array y el número de índice al que queremos acceder.

```
let numerosSorteados = [12, 45, 56, 324, 452];
```

```
numerosSorteados[0] // 12
```

```
numerosSorteados[1] // 45
```

{ }

LOS 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)

```
{}
```

```
var malaDivision = "35" / 2; // NaN no es un número
```

NULL (VALOR NULO)

Lo asignamos nosotros para indicar un valor vacío o desconocido.

```
{}
```

```
var temperatura = null; // No llegó un dato, algo falló
```

UNDEFINED (valor sin definir)

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

```
{}
```

```
var otraVariable; // undefined, no tiene valor  
otraVariable = "¡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.  
var redondeado = Math.round(20.49);
```