



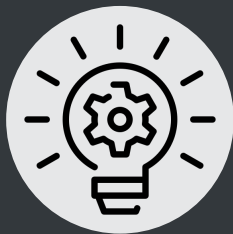
# Objetos literales

DigitalHouse>



**Certified Tech  
Developer**

The Ultimate Degree



Podemos decir que los objetos literales son la **representación en código** de un **elemento de la vida real**.

# Estructura básica

Un **objeto** es una estructura de datos que puede contener **propiedades** y **métodos**.

Para crearlo usamos llave de apertura y de cierre `{}`.

```
{} let auto = {  
    patente : 'AC 134 DD'  
};
```

## PROPIEDAD

Definimos el nombre de la **propiedad** del objeto.

## DOS PUNTOS

**Separa** el nombre de la propiedad de su valor.

## VALOR

Puede ser cualquier **tipo de dato** que conocemos.

## Propiedades de un objeto

Un objeto puede tener la cantidad de propiedades que queramos. Si hay más de una, las separamos con comas , .

Con la notación `objeto.propiedad` accedemos al valor de cada una de ellas.

```
{}  
  
let tenista = {  
  nombre: 'Roger',  
  apellido: 'Federer'  
};  
  
console.log(tenista.nombre) // Roger  
console.log(tenista.apellido) // Federer
```

# Métodos de un objeto

Una propiedad puede almacenar cualquier tipo de dato. Si una propiedad almacena una **función**, diremos que es un **método** del objeto. Con una estructura similar a la de las funciones expresadas, vemos que se crean mediante el nombre del método, seguido de una función anónima.

```
{}  
  
let tenista = {  
  nombre: 'Roger',  
  edad: 38,  
  activo: true,  
  saludar: function() {  
    return '¡Hola! Me llamo Roger';  
  }  
};
```

# Ejecución de un método de un objeto

Para ejecutar un método de un objeto usamos la notación `objeto.metodo()`. Los paréntesis del final son los que hacen que el método se ejecute.

```
let tenista = {  
  nombre: 'Roger',  
  apellido: 'Federer',  
  saludar: function() {  
    return '¡Hola! Me llamo Roger';  
  }  
};  
  
console.log(tenista.saludar()); // ¡Hola! Me llamo Roger
```

# Trabajando dentro del objeto

La palabra reservada **this** hace referencia al objeto en sí donde estamos parados. Es decir, el objeto en sí donde escribimos la palabra.

Con la anotación `this.propiedad` accedemos al valor de cada propiedad interna de ese objeto.

```
{  
  let tenista = {  
    nombre: 'Roger',  
    apellido: 'Federer',  
    saludar: function() {  
      return '¡Hola! Me llamo ' + this.nombre;  
    }  
  };  
  console.log(tenista.saludar()); // ¡Hola! Me llamo Roger
```

DigitalHouse>