

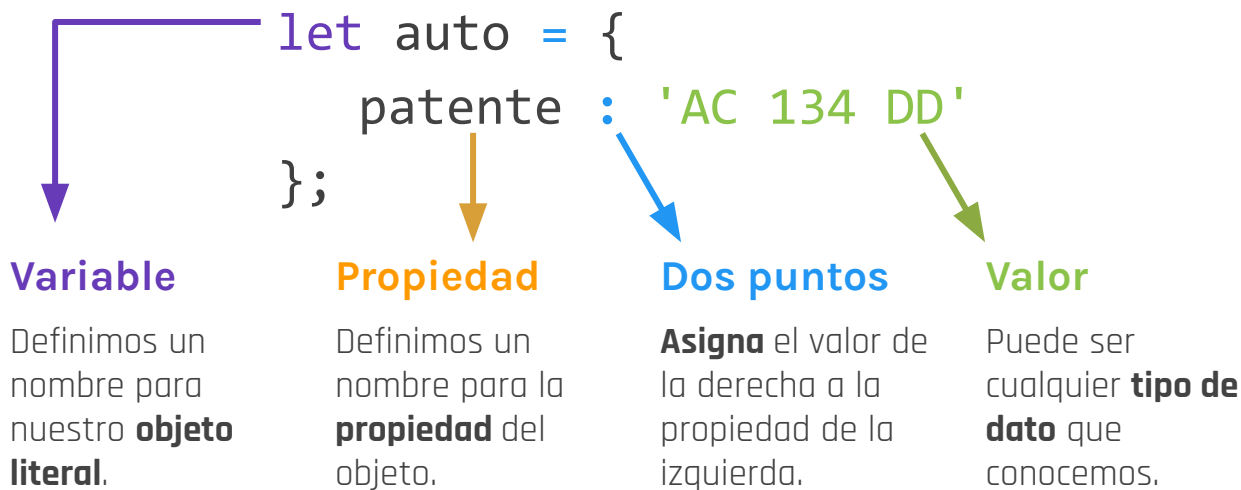
Objetos Literales

Los objetos literales son un tipo de dato que nos permiten una **representación en código** lo más cercana a un **elemento** de la **vida real**.

Estructura del Objeto Literal

Un **objeto literal** es una estructura de datos que contiene **propiedades** y **métodos**.

Para crearlo usamos la notación de llaves `{}`.



Estructura del Objeto Literal

Un **objeto literal** almacena información dentro de **propiedades**. Si hay más de una las separamos con comas (,).

Para acceder a la información de una propiedad utilizamos la notación de punto (.): ***variable.propiedad***

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

Métodos de un Objeto Literal

Una propiedad puede **almacenar** cualquier otro **tipo dato** del lenguaje.

Si una propiedad almacena **una función** diremos que es un **método** del objeto.

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

Métodos de un Objeto Literal

Para ejecutar el método de un objeto usamos la notación ***variable.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
```

Referenciando dentro del Objeto Literal

La palabra reservada **this** hace referencia al objeto en sí que la contiene. Con la notación **this.propiedad** accedemos al valor de **cada propiedad interna** de ese objeto.

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

PRACTIQUEMOS

