



# Colección de personas

DigitalHouse>



**Certified Tech  
Developer**

The Ultimate Degree

# Índice

1. [Objeto para una persona](#)
2. [Colección de personas](#)
3. [Acceder a propiedades](#)

# 1 | Objeto para una persona

# Objeto persona

Con el tipo de dato "objeto" podemos definir el valor para una persona:

JS

```
let persona = {  
  nombre: "Dua",  
  apellido: "Lipa",  
  ocupacion: "cantante",  
  estatura: 173,  
}
```

# Acceder a propiedades

Para leer valores del objeto persona, podemos usar la notación “punto” para acceder a las propiedades que queramos mostrar:

JS

```
console.log(persona.nombre) // Dua  
console.log(persona.ocupacion) // Cantante  
console.log(persona.estatura) // 173  
console.log(persona.edad) // undefined
```



Lo anterior nos sirve para una persona, pero,  
¿qué estructura necesitamos para almacenar  
una **lista** de personas?

# 2 | Colección de personas

## Objeto persona

Con el tipo de dato "array" podemos almacenar objetos para muchas personas.

```
let personas = [  
  {  
    nombre: "Dua",  
    apellido: "Lipa",  
    ocupacion: "cantante",  
    estatura: 173,  
  },  
  {  
    nombre: "Ariana",  
    apellido: "Grande",  
    ocupacion: "cantante",  
    estatura: 160,  
  },  
  {  
    nombre: "Taylor",  
    apellido: "Swift",  
    ocupacion: "cantante",  
    estatura: 180,  
  },  
];
```



# **3 | Acceder a propiedades**

# Acceder a propiedades

Para leer valores de cada objeto, dentro del array, tendremos que primero acceder al índice de la colección y luego utilizar la notación “punto” para acceder a la propiedad que queramos leer.

JS

```
console.log(personas[0].estatura) // 173  
console.log(personas[1].estatura) // 160  
console.log(personas[2].estatura) // 180
```

# Acceder a propiedades

También, podemos crear un loop que itere sobre todo el array y mostrar por consola alguna propiedad, como por ejemplo, la estatura.

JS

```
for (let index = 0; index < personas.length; index++) {  
  console.log(personas[index].estatura);  
}
```

# Acceder a propiedades

Además, podemos aplicar condicionales, o cualquier sintaxis válida de Javascript dentro de un ciclo. Observa en este ejemplo donde solo mostramos el nombre de las personas, si su estatura es mayor a 170.

JS

```
for (let index = 0; index < personas.length; index++) {  
  if (personas[index].estatura > 170) {  
    console.log(personas[index].nombre)  
  }  
}
```

DigitalHouse>