

# Destructuring



Nos permite **extraer datos** de **arrays** y **objetos literales** de una manera más sencilla y fácil de implementar.



# Sin usar destructuring

Para extraer datos de un **array** es necesario crear una variable y asignarle un elemento del mismo, usando el operador de **índice**.

```
{}
```

```
let colores = ['Rojo', 'Azul', 'Amarillo'];  
let azul = colores[1];
```

Para extraer datos de un **objeto**, es necesario crear una variable y asignarle una propiedad específica del mismo.

```
{}
```

```
let auto = {marca: 'Ford', anio: 1998};  
let marcaAuto = auto.marca;
```

# Desestructurando arrays

Para desestructurar un **array** declaramos una variable (podemos usar `var`, `let` o `const`) y, entre **corchetes**, escribimos el nombre que queremos. Podemos declarar más de una variable, separando cada una con una coma `,`.

Luego, igualamos esa estructura al array del cual queremos extraer los datos.

```
{  
  let colores = ['Rojo', 'Azul', 'Amarillo'];  
  let [rojo, azul, amarillo] = colores;  
}
```

# Desestructurando arrays


Partiendo de un **array** previamente definido, se transfiere cada dato a las variables que definimos nosotros.

JavaScript le asignará a cada variable el dato extraído de la estructura que elijamos **respetando el orden original**.

```
{ } let array = ['Rojo', 'Azul', 'Amarillo'];
```

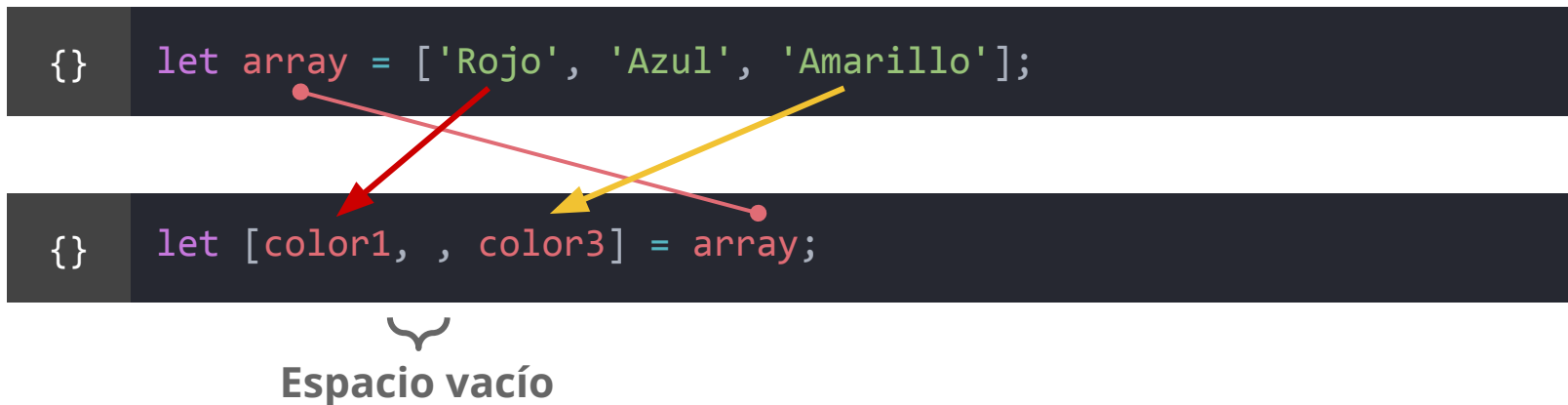


```
{ } let [color1, color2, color3] = array;
```



# Desestructurando arrays

Si queremos saltar un valor, podemos dejar vacío el nombre de la variable que correspondería con esa posición.



# Desestructurando objetos

Para desestructurar un **objeto literal**, creamos una variable (podemos usar `var`, `let` o `const`) y, entre **llaves**, declaramos el o los **nombres** de las **propiedades** que queremos extraer.

A esa estructura la igualamos al objeto del cual queremos extraer los datos.

```
{  
  let persona = {nombre: 'Laura', edad: 31, faltas: 3};  
  let {nombre, edad} = persona;
```

# Desestructurando objetos

Partiendo de un **objeto** previamente definido, se transfiere cada propiedad o método a una o más variables que definamos.

JavaScript le asignará a cada variable el valor de la propiedad que hayamos elegido.

```
{ } let persona = { nombre: 'Laura', edad: 31, faltas: 3};
```



```
{ } let { nombre, faltas } = persona;
```





# Desestructurando objetos

Es posible que en algún caso necesitemos **cambiarle el nombre a la variable** que estamos creando.

En ese caso, a continuación de la propiedad que estamos extrayendo **colocamos dos puntos :** seguidos del nuevo nombre.

```
{ } let persona = { nombre: 'Laura', edad: 31, faltas: 3 };  
  
{ } let { nombre, faltas: totalFaltas } = persona;
```

**Nuevo nombre**

“

La desestructuración **no modifica** el array u objeto literal de origen.

Su único objetivo es **copiar los valores** de una manera más práctica y rápida.

”



DigitalHouse>  
Coding School