

Javascript Objects



Destructuring... Pero en Objetos

El "**destructuring**" en JavaScript es una característica que permite descomponer un array o un objeto en variables más pequeñas y accesibles de manera más conveniente. Esto facilita la extracción de datos y asignación de variables de manera concisa.

Destructuring... Pero en Objetos

Supongamos que tenemos este objeto:

```
const personaje = {  
  nombre: 'Shrek',  
  edad: 30,  
  especie: 'ogro',  
  amigos: ['Burro', 'Fiona']  
};
```

Destructuring... Pero en Objetos

Para extraer propiedades de este objeto y asignarlas a variables, puedes usar el destructuring:

```
const { nombre, edad, especie } = personaje;
```

```
console.log(nombre); // 'Shrek'
```

```
console.log(edad);   // 30
```

```
console.log(especie); // 'ogro'
```


Destructuring... Pero en Objetos

Puedes asignar una propiedad a una variable con un nombre diferente:

```
const { nombre: nombrePersonaje, edad: edadPersonaje } = personaje;
```

```
console.log(nombrePersonaje); // 'Shrek'
```

```
console.log(edadPersonaje);   // 30
```

Destructuring... Pero en Objetos

Si la propiedad no existe en el objeto, puedes asignar un valor predeterminado:

```
const { nombre, ocupacion = 'desconocida' } = personaje;
```

```
console.log(nombre);    // 'Shrek'
```

```
console.log(ocupacion); // 'desconocida'
```

Destructuring... Pero en Objetos



Puedes utilizar destructuring en objetos que contienen otros objetos o arrays:

```
const pelicula = {  
  titulo: 'Shrek',  
  año: 2001,  
  personajes: {  
    protagonista: 'Shrek',  
    compañero: 'Burro'  
  }  
};  
  
const { titulo, personajes: { protagonista, compañero } } = pelicula;  
  
console.log(titulo);      // 'Shrek'  
console.log(protagonista); // 'Shrek'  
console.log(compañero);   // 'Burro'
```


Destructuring... Pero en Objetos

También puedes usar destructuring directamente en los parámetros de una función:

```
function mostrarPersonaje({ nombre, edad }) {  
  console.log(`El personaje es ${nombre} y tiene ${edad} años.`);  
}  
  
mostrarPersonaje(personaje); // 'El personaje es Shrek y tiene 30 años.'
```


Ventajas del Destructuring

Simplicidad: Permite extraer datos de objetos y arrays de manera más concisa y legible.

Desacoplamiento: Facilita el trabajo con datos estructurados sin tener que acceder a ellos mediante una cadena de propiedades o índices.

Flexibilidad: Ofrece una sintaxis flexible para manejar valores predeterminados y propiedades anidadas.

Spread Operator... Pero en Objetos

El spread operator se usa para expandir las propiedades de un objeto en otro objeto. Esto es útil para copiar objetos, fusionar objetos, o modificar propiedades de un objeto sin mutar el original.

Veamos como...

Spread Operator

Podemos copiar un Objeto:

```
const obj1 = { name: 'Alice', age: 25 };  
const obj2 = { ...obj1 };  
console.log(obj2); // Output: { name: 'Alice', age: 25 }
```

Spread Operator

Podemos fusionar un Objeto:

```
const obj1 = { name: 'Alice' };  
const obj2 = { age: 25 };  
const merged = { ...obj1, ...obj2 };  
console.log(merged); // Output: { name: 'Alice', age: 25 }
```


Spread Operator

Podemos modificar Propiedades:

```
const obj = { name: 'Alice', age: 25 };  
const updated = { ...obj, age: 26 };  
console.log(updated); // Output: { name: 'Alice', age: 26 }
```