

Práctica

Enunciado

Una reserva de un hotel está representada por la siguiente estructura de datos:

```
{ price: 37.50, count: 3, room: "standard", prepaid: false, succeeded: true
}
```

Donde:

- id: identificador de la reserva
- price: precio por noche
- count: número de noches
- room: tipo de habitación
- prepaid: se ha pagado al reservar (true) o se pagará en el hotel (false)
- succeeded: indica si ha ido todo bien (true) o ha ocurrido un error (false)

A partir del siguiente listado de reservas:

```
const bookings = [
  { id: 1, price: 45.30, count: 3, room: "standard", prepaid: false,
    succeeded: true },
  { id: 2, price: 92.45, count: 4, room: "superior", prepaid: false,
    succeeded: true },
  { id: 3, price: 63.50, count: 7, room: "standard", prepaid: true,
    succeeded: false },
  { id: 3, price: 37.50, count: 2, room: "standard", prepaid: true,
    succeeded: true },
  { id: 5, price: 87.90, count: 5, room: "superior", prepaid: true,
    succeeded: false }
];
```

Realizar los siguientes ejercicios:

Listar todas las reservas del hotel

```
for (i = 0; i < bookings.length; i++) {
  imprimir(bookings[i]);
}

function imprimir(booking) {
  console.log("----RESERVA " + booking.id + "-----");
  console.log("    Price: " + booking.price * booking.count + " €");
  console.log("    Habitación: " + booking.room);
}
```

```
    console.log("    Pagada: " + booking.prepaid);
    console.log("    Error: " + !booking.succeeded);
}
```

```
for (booking of bookings) {
    imprimir(booking);
}

function imprimir(booking) {
    console.log("----RESERVA " + booking.id + "-----");
    console.log("    Price: " + booking.price * booking.count + " €");
    console.log("    Habitación: " + booking.room);
    console.log("    Pagada: " + booking.prepaid);
    console.log("    Error: " + !booking.succeeded);
}
```

Reto: modificar la función imprimir para que recorra los atributos de la estructura de la reserva:

```
for (booking of bookings) {
    imprimir(booking);
}

function imprimir(booking) {
    console.log("-----");
    for (attr in booking) {
        console.log(attr.toUpperCase() + ": " + booking[attr]);
    }
}
```

Crear una función que nos indique si al menos una reserva ha fallado

```
console.log("Error: " + existeError(bookings));

function existeError(bookings) {
    let succeeded = true;

    for (booking of bookings) {
        succeeded = succeeded && booking.succeeded;
    }

    return !succeeded;
}
```

Listar sólo las reservas que han tenido éxito

```
for (booking of bookings) {  
  if (booking.succeeded) imprimir(booking);  
}  
  
function imprimir(booking) {  
  console.log("-----");  
  for (attr in booking) {  
    console.log(attr.toUpperCase() + ": " + booking[attr]);  
  }  
}
```

Listar las reservas que ya están pagadas

```
for (booking of bookings) {  
  if (booking.prepaid) imprimir(booking);  
}  
  
function imprimir(booking) {  
  console.log("-----");  
  for (attr in booking) {  
    console.log(attr.toUpperCase() + ": " + booking[attr]);  
  }  
}
```

Calcular la suma total de los precios

```
let total = 0;  
for (booking of bookings) {  
  total += booking.price * booking.count;  
  // total += 5 es lo mismo que: total = total + 5  
}  
  
console.log("Total", total);
```

Aplicar un descuento del 10% a todas las reservas

```
for (booking of bookings) {  
  booking.price = booking.price * 0.9;  
}
```

Buscar reservas pagadas que han fallado

```
for (booking of bookings) {  
  if (!booking.succeeded && booking.prepaid) imprimir(booking);  
}  
  
function imprimir(booking) {  
  console.log("-----");  
  for (attr in booking) {  
    console.log(attr.toUpperCase() + ": " + booking[attr]);  
  }  
}
```

Buscar si existe algun identificador repetido (bucle anidado)

```
let repeated = false;  
  
for (i = 0; i < bookings.length; i++) {  
  for (x = i + 1; x < bookings.length; x++) {  
    repeated = repeated || bookings[i].id === bookings[x].id;  
  }  
}  
  
console.log("Identificadores repetidos: ", repeated);
```