

Uso de .call()

Problema I (15 puntos)

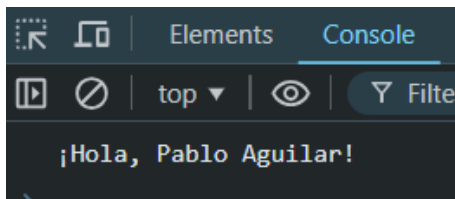
Cambiar el contexto de `this`

Escribe una función llamada `saludar` que tome un parámetro `nombre` y devuelva un saludo personalizado. Luego, crea un objeto `persona` con una propiedad `nombre`. Utiliza el método `call()` para llamar a la función `saludar` con el contexto del objeto `persona`.

Respuesta:

Agregue una segunda propiedad para persona "Apellido" y también puede llamar esta propiedad usando el mismo método call

```
function saludar(nombre, apellido) {  
  return `¡Hola, ${nombre} ${apellido}!`;  
}  
  
const persona = {  
  nombre: "Pablo",  
  apellido: "Aguilar",  
};  
  
const saludo = saludar.call(persona, persona.nombre, persona.apellido);  
  
console.log(saludo);
```



Problema II (15 puntos)

Cambiar el contexto de `this` en un método de objeto

Crea un objeto `auto` con una propiedad `marca` y un método `mostrarMarca` que devuelva un mensaje con la marca del auto. Luego, crea otro objeto `moto` con una propiedad `marca`.

Utiliza el método `call()` para llamar al método `mostrarMarca` del objeto `auto` con el contexto del objeto `moto`.

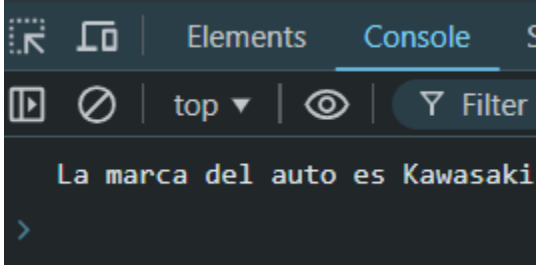
Respuesta:

```
const auto = {
  marca: "Honda",
  mostrarmarca: function () {
    return `La marca del auto es ${this.marca}`;
  },
};

const moto = {
  marca: "Kawasaki",
};

const mensaje = auto.mostrarmarca.call(moto);

console.log(mensaje);
```



The screenshot shows the browser's developer console with the 'Console' tab selected. It displays the output of the JavaScript code: 'La marca del auto es Kawasaki'. The console interface includes standard navigation icons and a filter input.

Problema III (10 Puntos)

Crea dos objetos, `persona1` y `persona2`, cada uno con una propiedad `nombre`. Define una función `saludar` que devuelva un mensaje de saludo utilizando la propiedad `nombre`. Usa `call()` para invocar la función `saludar` con el contexto de `persona2`.

Respuesta:

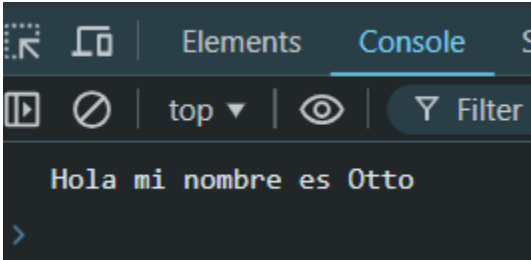
```
const persona1 = {
  nombre: "Eduardo",
};

const persona2 = {
  nombre: "Otto",
};

function saludar() {
  return `Hola mi nombre es ${this.nombre}`;
}

const saludo = saludar.call(persona2);

console.log(saludo);
```



The screenshot shows the browser's developer console with the 'Console' tab selected. It displays the output of the JavaScript code: 'Hola mi nombre es Otto'. The console interface includes standard navigation icons and a filter input.

Problema IV (10 Puntos)

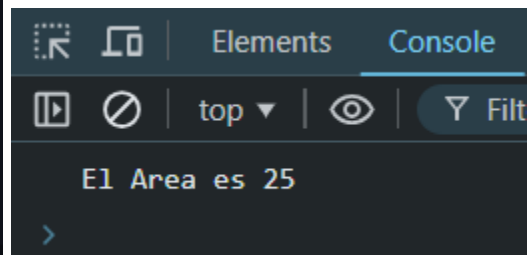
Crea un objeto `rectangulo` con propiedades `ancho` y `alto`, y un método `area` que calcule el área del rectángulo. Luego, crea un objeto `cuadrado` con una propiedad `lado`. Usa `call()` para invocar el método `area` del objeto `rectangulo` con el contexto del objeto `cuadrado`.

Respuesta:

```
const rectangulo = {
  ancho: 0,
  alto: 0,
  area: function () {
    return this.ancho * this.alto;
  },
};

const cuadrado = {
  lado: 5,
};

const areaCuadrado = rectangulo.area.call({
  ancho: cuadrado.lado,
  alto: cuadrado.lado,
});
console.log(`El Area es ${areaCuadrado}`);
```



Uso de .apply()

Problema V (15 puntos)

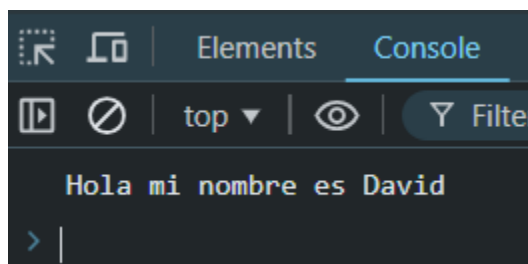
Crea dos objetos, `persona1` y `persona2`, cada uno con una propiedad `nombre`. Define una función `presentar` que devuelva un mensaje de presentación utilizando la propiedad `nombre`. Usa `apply()` para invocar la función `presentar` con el contexto de `persona2`.

Respuesta:

```
const persona1 = {
  nombre: "Luis",
};
const persona2 = {
  nombre: "David",
};

function presentar() {
  return `Hola mi nombre es ${this.nombre}`;
}

const presentacion = presentar.apply(persona2);
console.log(presentacion);
```



Problema VI (15 puntos)

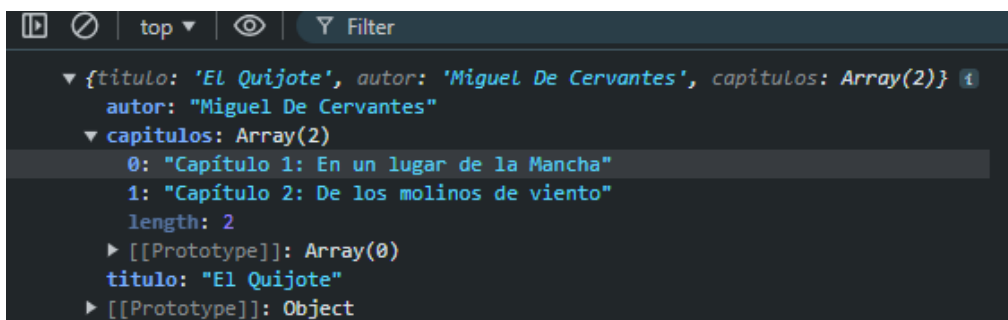
Expansión de Objeto con Array Crea un objeto `libro` con propiedades `titulo` y `autor`. Define una función `agregarCapitulos` que tome un array de capítulos y los agregue como una nueva propiedad `capitulos` al objeto `libro`. Usa `apply()` para invocar la función `agregarCapitulos` con el contexto del objeto `libro` y un array de capítulos.

```
const libro = {
  titulo: "El Quijote",
  autor: "Miguel De Cervantes",
};

function agregarCapitulos(capitulos) {
  this.capitulos = capitulos;
}

const capitulos = [
  "Capítulo 1: En un lugar de la Mancha",
  "Capítulo 2: De los molinos de viento",
];

agregarCapitulos.apply(libro, [capitulos]);
console.log(libro);
```



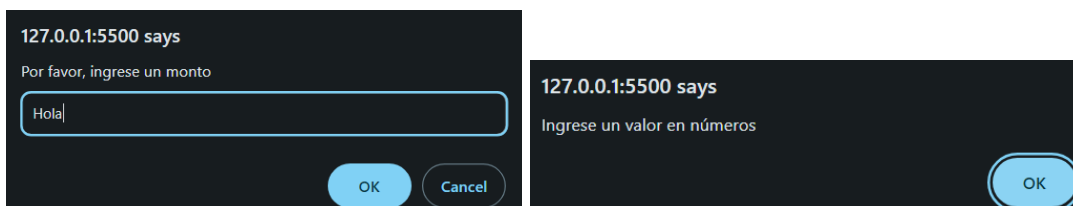
Problema VII (20 puntos)

Actualización de Saldo en Cuenta Bancaria

Crea un objeto `cuentaBancaria` con propiedades `titular` y `saldo`. Define una función `actualizarSaldo` que tome un monto y lo sume al saldo actual de la cuenta. Usa `apply()` para invocar la función `actualizarSaldo` con el contexto del objeto `cuentaBancaria` y un array que contenga el monto a agregar.

Respuesta:

Agregue una función para que el usuario pueda ingresar el monto y un validador para que lo que el usuario ingrese sea un número, y luego le coloque un "alert" para que me de el nuevo saldo.



127.0.0.1:5500 says

Por favor, ingrese un monto

OK

Cancel

127.0.0.1:5500 says

Su nuevo saldo es 5600

OK

```
top Filter
▶ {titular: 'Alejandro Cardona ', saldo: 5600}
>
```

```
cuentaBancaria = {
  titular: "Alejandro Cardona ",
  saldo: 5000,
};

function actualizarSaldo(monto) {
  this.saldo += monto;
}

function ingresoSaldo() {
  let valor;
  while (true) {
    valor = prompt("Por favor, ingrese un monto");
    const numero = Number(valor);
    if (!isNaN(numero)) {
      return numero;
    } else {
      alert("Ingrese un valor en números");
    }
  }
}

const monto = ingresoSaldo();

actualizarSaldo.apply(cuentaBancaria, [monto]);

console.log(cuentaBancaria);
alert(`Su nuevo saldo es ${cuentaBancaria.saldo}`);
```