

```
744         'error' => $quote['error'],
745     );
746 }
747
748 $sort_order = array();
749 foreach ($sources as $key => $value) {
750     $sort_order[$key] = $value['sort_order'];
751 }
752 array_multisort($sort_order, SORT_ASC, $sources);
753
754 $this->session->data['ip'] = $this->shipping_method;
755 $this->session->data['ip'] = $address;
756
757 if (empty($sources)) {
758     $json['error'] = $this->language->get(
759         'error_no_shipping_method');
760 } else {
761     $json['quotes'] = $sources;
762 }
763
764 if (isset($this->session->data['ip']["shipping_method"])) {
765     $json['selected'] = $this->session->data['ip']["shipping_method"];
766 } else {
767     $json['selected'] = '';
768 }
769
770 $this->response->addHeader('Content-Type: application/json');
```

Objetos en JavaScript

Aprende a trabajar con objetos en JavaScript

Empezar

Descripción general

En este curso aprenderás todo sobre los objetos en JavaScript, incluyendo cómo crear y manipular objetos, cómo acceder a sus propiedades y métodos, y cómo utilizarlos en tus aplicaciones

01 Introducción

A decorative horizontal bar at the top of the slide featuring a series of overlapping squares and circles in various shades of purple and blue.

Introducción a los objetos en JavaScript

01 | Introducción a los objetos en JavaScript

Los objetos en JavaScript son una pieza fundamental de este lenguaje de programación. Nos permiten organizar y representar datos, así como también manipular y acceder a ellos de manera eficiente. En este tema exploraremos los conceptos básicos de los objetos en JavaScript, incluyendo su creación, propiedades y métodos.

¿Qué es un objeto en JavaScript?

Un objeto en JavaScript es una estructura que nos permite almacenar y representar datos a través de propiedades y métodos. Las propiedades son variables que contienen valores, mientras que los métodos son funciones que pueden realizar acciones o procesos sobre esos valores.

A diferencia de otros lenguajes de programación, en JavaScript los objetos pueden ser creados de manera dinámica y no necesitan una definición de clase previa. Esto significa que podemos añadir o eliminar propiedades y métodos en cualquier momento.

Creación de objetos en JavaScript

Existen diferentes formas de crear objetos en JavaScript. Una de ellas es utilizando la sintaxis de objeto literal, donde se define el objeto mediante llaves `{}` y se especifican las propiedades y métodos dentro de ellas.

```
let persona = {  
  nombre: "Juan",  
  edad: 25,  
  saludar: function() {  
    console.log("¡Hola! Mi nombre es " + this.nombre);  
  }  
};
```

También podemos utilizar la palabra clave `new` junto con una función constructora para crear objetos. Esta función constructora actúa como una plantilla para la creación de nuevos objetos.

```
function Persona(nombre, edad) {  
  this.nombre = nombre;  
  this.edad = edad;  
  this.saludar = function() {  
    console.log("¡Hola! Mi nombre es " + this.nombre);  
  }  
}  
  
let persona = new Persona("María", 30);
```

Acceso a propiedades y métodos

Para acceder a las propiedades y métodos de un objeto en JavaScript, utilizamos la sintaxis de punto `objeto.propiedad` o `objeto.metodo()`. Por ejemplo, en el objeto `persona` creado anteriormente, podemos acceder a la propiedad `nombre` y llamar al método `saludar` de la siguiente manera:

```
console.log(persona.nombre); // Output: "María"  
persona.saludar(); // Output: "¡Hola! Mi nombre es María"
```

También es posible acceder a las propiedades y métodos utilizando la sintaxis de corchetes `objeto['propiedad']` o `objeto['metodo']()`. Esta sintaxis es útil cuando el nombre de la propiedad o método es una cadena o si queremos acceder de forma dinámica a las propiedades.

```
console.log(persona['nombre']); // Output: "María"  
persona['saludar'](); // Output: "¡Hola! Mi nombre es María"
```

Modificación de objetos

Los objetos en JavaScript son mutables, lo que significa que podemos modificar sus propiedades y métodos después de haberlos creado. Esto nos permite actualizar los valores de las propiedades o añadir nuevas propiedades o métodos cuando sea necesario.

```
persona.edad = 35; // Modificando la propiedad edad  
persona.hobby = "pintar"; // Añadiendo una nueva propiedad
```

También es posible eliminar propiedades de un objeto utilizando el operador `delete`.

```
delete persona.hobby; // Eliminando la propiedad hobby
```

Conclusiones

En este tema hemos explorado los conceptos básicos de los objetos en JavaScript. Hemos aprendido cómo crear objetos utilizando la sintaxis de objeto literal y la palabra clave `new`, así como también cómo acceder, modificar y eliminar propiedades y métodos de un objeto. Los objetos son una herramienta poderosa en JavaScript que nos permite organizar y manipular datos de manera eficiente. En los próximos temas, profundizaremos en temas más avanzados relacionados con los objetos en JavaScript.

Conclusión - Introducción a los objetos en JavaScript

En conclusión, los objetos en JavaScript son una parte fundamental del lenguaje, ya que nos permiten agrupar valores y funciones relacionadas en una única entidad. A lo largo de este curso hemos aprendido los conceptos básicos de los objetos en JavaScript, como su creación, sus propiedades y métodos, y cómo manipularlos. Espero que este conocimiento te sea útil en tus proyectos y te ayude a mejorar tus habilidades como desarrollador en JavaScript. ¡Sigue practicando y explorando nuevas formas de utilizar objetos en JavaScript!



Propiedades y métodos de los objetos

Los objetos en JavaScript son estructuras de datos que contienen propiedades y métodos. Las propiedades son valores asociados a un objeto, mientras que los métodos son funciones que pueden ser invocadas por el objeto. A continuación, veremos con mayor detalle qué son las propiedades y los métodos, cómo se accede a ellos y cómo se utilizan.

Propiedades de los objetos

Las propiedades de un objeto representan características o atributos del mismo. Pueden ser de diferentes tipos, como cadenas de texto, números, booleanos, matrices e incluso otros objetos. Cada propiedad tiene un nombre único que sirve como identificador.

Accediendo a las propiedades

Para acceder a una propiedad de un objeto, se utiliza la notación de punto o la notación de corchetes.

La notación de punto se utiliza escribiendo el nombre del objeto, seguido de un punto y el nombre de la propiedad:

```
// Definición del objeto
var persona = {
  nombre: "Juan",
  edad: 30,
  casado: false
};

// Accediendo a las propiedades con la notación de punto
```

```
console.log(persona.nombre); // Juan
console.log(persona.edad); // 30
console.log(persona.casado); // false
```

La notación de corchetes se utiliza colocando el nombre de la propiedad entre corchetes:

```
// Accediendo a las propiedades con la notación de corchetes
console.log(persona["nombre"]); // Juan
console.log(persona["edad"]); // 30
console.log(persona["casado"]); // false
```

Ambas notaciones son equivalentes y pueden ser utilizadas indistintamente. Sin embargo, la notación de corchetes es útil cuando se necesitan acceder a propiedades cuyo nombre es una cadena de texto almacenada en una variable.

Modificando propiedades

Para modificar el valor de una propiedad, se puede utilizar la misma notación de punto o de corchetes que se utiliza para acceder a ellas. Simplemente se asigna un nuevo valor a la propiedad:

```
// Modificando propiedades utilizando la notación de punto
persona.edad = 35;
persona.casado = true;

// Modificando propiedades utilizando la notación de corchetes
persona["edad"] = 35;
persona["casado"] = true;
```

Añadiendo propiedades

Es posible añadir nuevas propiedades a un objeto simplemente asignándoles un valor:

```
// Añadiendo propiedades
persona.profesion = "Ingeniero";
persona.sueldo = 5000;

console.log(persona.profesion); // Ingeniero
console.log(persona.sueldo); // 5000
```

Métodos de los objetos

Los métodos de un objeto son funciones asociadas a ese objeto. Al igual que las propiedades, cada método tiene un nombre único que lo identifica.

Definiendo métodos

Los métodos se definen dentro del objeto de la misma manera que se definen las propiedades, pero en lugar de asignarles un valor, se les asigna una función:

```
// Definición de un objeto con un método
var persona = {
  nombre: "Juan",
  edad: 30,
  casado: false,
  saludar: function() {
    console.log("¡Hola! Mi nombre es " + this.nombre);
  }
};

persona.saludar(); // ¡Hola! Mi nombre es Juan
```

En el ejemplo anterior, hemos definido un método llamado "saludar" que muestra un mensaje por consola.

Invocando métodos

Para invocar un método de un objeto, se utiliza la notación de punto, seguida del nombre del método y los paréntesis "()":

```
persona.saludar();
```

Uso de la palabra clave "this"

Dentro de un método, es común utilizar la palabra clave "this" para acceder a las propiedades y otros métodos del mismo objeto. La palabra clave "this" hace referencia al propio objeto en el que se encuentra el método.

```
var persona = {  
  nombre: "Juan",  
  edad: 30,  
  casado: false,  
  presentarse: function() {  
    console.log("¡Hola! Mi nombre es " + this.nombre + " y tengo " + this.edad);  
  }  
};  
  
persona.presentarse(); // ¡Hola! Mi nombre es Juan y tengo 30 años.
```

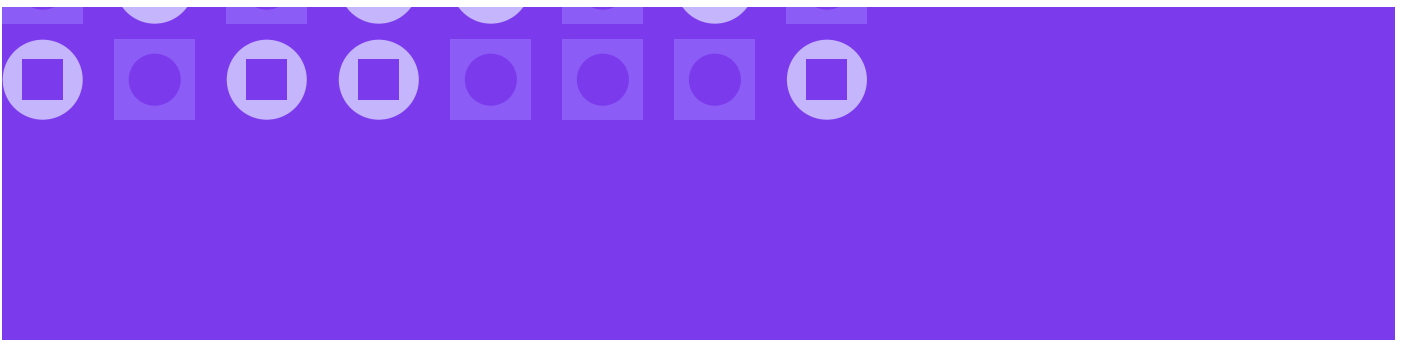
En el ejemplo anterior, utilizamos "this.nombre" y "this.edad" para acceder a las propiedades del objeto "persona" dentro del método "presentarse".

Conclusion

En este tema hemos explorado las propiedades y los métodos de los objetos en JavaScript. Hemos aprendido cómo acceder y modificar las propiedades de un objeto, así como cómo definir y llamar a los métodos. Las propiedades y métodos de un objeto proporcionan flexibilidad y funcionalidad, permitiéndonos modelar y manipular datos de manera eficiente en nuestros programas.

Conclusión - Propiedades y métodos de los objetos

En resumen, hemos visto que los objetos en JavaScript son contenedores de propiedades y métodos, que nos permiten representar entidades del mundo real. Durante este curso hemos aprendido a crear objetos, agregar propiedades y métodos, y acceder a ellos utilizando la notación de punto y corchetes. Además, también hemos explorado conceptos avanzados como la herencia y la manipulación de objetos. Espero que hayas disfrutado de este curso y que puedas aplicar tus conocimientos en tus proyectos futuros.



Manipulación y creación de objetos en JavaScript

03 | Manipulación y creación de objetos en JavaScript

En JavaScript, los objetos son una parte fundamental del lenguaje y proporcionan una manera poderosa de almacenar y manipular datos. Los objetos en JavaScript son estructuras de datos que contienen propiedades y métodos, lo que los hace muy flexibles y versátiles.

Creación de objetos

En JavaScript, hay varias formas de crear objetos. La forma más básica es utilizando la sintaxis de objeto literal. Un objeto literal se define utilizando llaves `{ }` y puede contener una lista de propiedades separadas por comas. Cada propiedad consiste en una clave y un valor, separados por dos puntos `:`. Por ejemplo:

```
let persona = {  
  nombre: "Juan",  
  edad: 25,
```

```
    profesion: "Desarrollador"
  };
```

También es posible crear objetos utilizando la función constructora `Object()`. Esta función crea un nuevo objeto vacío al que se le pueden añadir propiedades y métodos. Por ejemplo:

```
let coche = new Object();
coche.marca = "Toyota";
coche.modelo = "Corolla";
coche.año = 2020;
```

Otra forma de crear objetos es utilizando otras funciones como plantillas para generar objetos con propiedades y métodos predefinidos. Estas funciones se conocen como funciones constructoras y se utilizan junto con la palabra clave `new`. Por ejemplo:

```
function Persona(nombre, edad) {
  this.nombre = nombre;
  this.edad = edad;
  this.saludar = function() {
    console.log("Hola, mi nombre es " + this.nombre);
  };
}

let juan = new Persona("Juan", 25);
juan.saludar(); // Imprime "Hola, mi nombre es Juan"
```

Acceso y manipulación de propiedades

Una vez que se ha creado un objeto, es posible acceder y manipular sus propiedades utilizando la sintaxis de punto `objeto.propiedad` o la sintaxis de

corchetes `objeto["propiedad"]` . Por ejemplo:

```
console.log(persona.nombre); // Imprime "Juan"
console.log(coche["marca"]); // Imprime "Toyota"

persona.edad = 26; // Modifica el valor de la propiedad "edad"
coche["modelo"] = "Camry"; // Modifica el valor de la propiedad "modelo"
```

También es posible acceder a las propiedades de un objeto utilizando un bucle `for...in` . Este bucle itera sobre todas las propiedades enumerables de un objeto. Por ejemplo:

```
for (let propiedad in persona) {
  console.log(propiedad + ": " + persona[propiedad]);
}
```

Métodos de objetos

Los objetos en JavaScript también pueden contener métodos, que son funciones asociadas a un objeto. Estos métodos pueden ser utilizados para realizar acciones o modificar el estado del objeto.

Para añadir un método a un objeto, se puede hacer dentro de la definición del objeto utilizando una función anónima o utilizando una función declarada previamente. Por ejemplo:

```
let persona = {
  nombre: "Juan",
  edad: 25,
  saludar: function() {
    console.log("Hola, mi nombre es " + this.nombre);
  },
}
```

```
    envejecer: function(anos) {
      this.edad += anos;
    }
  };

  persona.saludar(); // Imprime "Hola, mi nombre es Juan"
  persona.envejecer(2);
  console.log(persona.edad); // Imprime "27"
```

Herencia de objetos

Una característica importante de los objetos en JavaScript es la posibilidad de heredar propiedades y métodos de otros objetos. Esto se logra utilizando el concepto de prototipos.

Cada objeto en JavaScript tiene un enlace interno a otro objeto llamado prototipo. Cuando se accede a una propiedad o método en un objeto, si no se encuentra en el objeto en sí, JavaScript buscará en su prototipo y seguirá buscando en la cadena de prototipos hasta encontrar el miembro deseado.

Para establecer una herencia, se puede utilizar la función `Object.create()` para crear un nuevo objeto que tenga como prototipo otro objeto existente. Por ejemplo:

```
let animal = {
  sonar: function() {
    console.log("El animal emite un sonido");
  }
};

let perro = Object.create(animal);
perro.ladRAR = function() {
```

```
console.log("El perro ladra");  
};  
  
perro.sonar(); // "El animal emite un sonido"  
perro.ladRAR(); // "El perro ladra"
```

En resumen, la manipulación y creación de objetos en JavaScript es una parte esencial del lenguaje. Los objetos permiten almacenar y organizar datos de manera estructurada, y los métodos proporcionan funcionalidades asociadas a los objetos. Además, la herencia de objetos brinda la posibilidad de reutilizar propiedades y métodos de otros objetos, lo que permite un desarrollo más eficiente y estructurado en JavaScript.

Conclusión - Manipulación y creación de objetos en JavaScript

En conclusión, los objetos en JavaScript son estructuras de datos fundamentales que nos permiten representar entidades complejas y organizar la lógica de nuestros programas de manera más eficiente. Durante este curso hemos explorado los conceptos básicos de los objetos en JavaScript, como la creación de objetos, la adición de propiedades y la definición de métodos. También hemos aprendido a manipular y acceder a los objetos utilizando la notación de punto y corchetes. Espero que este curso te haya brindado una base sólida para que puedas continuar

explorando y utilizando objetos en tus proyectos de JavaScript.



Ejercicios Practicos

Pongamos en práctica tus conocimientos

04 | Ejercicios Practicos

En esta lección, pondremos la teoría en práctica a través de actividades prácticas. Haga clic en los elementos a continuación para verificar cada ejercicio y desarrollar habilidades prácticas que lo ayudarán a tener éxito en el tema.

Creación de un objeto



Crea un objeto llamado persona con las siguientes propiedades: nombre, edad, ciudad. Asigna valores a las propiedades y muestra por consola la información del objeto.

Manipulación de propiedades



Crea un objeto llamado libro con las propiedades: título, autor y año de publicación. Luego, cambia el valor de la propiedad año de publicación y muestra por consola la información actualizada del libro.

Creación de varios objetos



Crea varios objetos de la clase Estudiante con las propiedades: nombre, edad y carrera. Asigna valores a las propiedades y muestra por consola la información de cada objeto.



Resumen

Repasemos lo que acabamos de ver hasta ahora

05 | Resumen

- ✓ En conclusión, los objetos en JavaScript son una parte fundamental del lenguaje, ya que nos permiten agrupar valores y funciones relacionadas en una única entidad. A lo largo de este curso hemos aprendido los conceptos básicos de los objetos en JavaScript, como su creación, sus propiedades y métodos, y cómo manipularlos. Espero que este conocimiento te sea útil en tus proyectos y te ayude a mejorar tus habilidades como desarrollador en JavaScript. ¡Sigue practicando y explorando nuevas formas de utilizar objetos en JavaScript!
- ✓ En resumen, hemos visto que los objetos en JavaScript son contenedores de propiedades y métodos, que nos permiten representar entidades del mundo real. Durante este curso hemos aprendido a crear objetos, agregar propiedades y métodos, y acceder a ellos utilizando la notación de punto y corchetes. Además, también hemos explorado conceptos avanzados como la herencia y la manipulación de objetos. Espero que hayas disfrutado de este curso y que puedas aplicar tus conocimientos en tus proyectos futuros.

- ✓ En conclusión, los objetos en JavaScript son estructuras de datos fundamentales que nos permiten representar entidades complejas y organizar la lógica de nuestros programas de manera más eficiente. Durante este curso hemos explorado los conceptos básicos de los objetos en JavaScript, como la creación de objetos, la adición de propiedades y la definición de métodos. También hemos aprendido a manipular y acceder a los objetos utilizando la notación de punto y corchetes. Espero que este curso te haya brindado una base sólida para que puedas continuar explorando y utilizando objetos en tus proyectos de JavaScript.



Prueba

Comprueba tus conocimientos respondiendo unas preguntas

06 | Prueba

Pregunta 1/6

¿Qué es un objeto en JavaScript?

- ☐ Una función
 - ☐ Un tipo de dato primitivo
 - ☐ Una colección de propiedades y métodos
-

Pregunta 2/6

¿Cuál de las siguientes afirmaciones es verdadera sobre los objetos en JavaScript?

- ☐ Los objetos no pueden contener funciones
 - ☐ Los objetos pueden tener propiedades y métodos
 - ☐ Los objetos solo pueden contener propiedades
-

Pregunta 3/6

¿Cómo se crea un objeto en JavaScript?

- ☐ Usando la palabra clave 'class'
 - ☐ Utilizando la función 'object'
 - ☐ Usando la palabra clave 'new' seguida de una función constructora
-

Pregunta 4/6

¿Qué es una propiedad en un objeto JavaScript?

- ☐ Una función asociada al objeto
- ☐ Un valor que describe una característica del objeto

☐ Una variable local

Pregunta 5/6

¿Cuál es la sintaxis para acceder a la propiedad de un objeto en JavaScript?

- ☐ objeto.propiedad
 - ☐ objeto(\$propiedad)
 - ☐ propiedad.objeto
-

Pregunta 6/6

¿Cómo se define un método en un objeto JavaScript?

- ☐ Asignando una función a una propiedad del objeto
 - ☐ Declarando una función con la palabra clave 'method'
 - ☐ Creando un objeto separado para cada método
-

Entregar

Conclusión

Felicidades!

¡Felicitaciones por completar este curso! Has dado un paso importante para desbloquear todo tu potencial. Completar este curso no se trata solo de adquirir conocimientos; se trata de poner ese conocimiento en práctica y tener un impacto positivo en el mundo que te rodea.



Comparte este curso

Created with [LearningStudioAI](#)

v0.5.63