Programador JavaScript Avanzado

Unidad 7: Trabajo con Clases

Contactos: tecnologia@elearning-total.com
Web: www.elearning-total.com

Indice

Unidad 7: Trabajo con Clases

Objetivos

Que el alumno logre:

- Implementar el paradigma de programación orientada a objetos

Prototipos

En JavaScript, los prototipos son un mecanismo fundamental para la herencia y la compartición de propiedades y métodos entre objetos. Cada objeto en JavaScript tiene un prototipo, excepto el objeto raíz (Object.prototype), que es el punto final de la cadena de prototipos.

Cuando accedes a una propiedad o método en un objeto, JavaScript busca primero esa propiedad en el propio objeto. Si no la encuentra, busca en el prototipo del objeto y luego en el prototipo del prototipo, y así sucesivamente, siguiendo la cadena de prototipos hasta llegar al objeto Object.prototype.

La relación entre objetos y prototipos se establece a través del enlace __proto__ (antes de ES6) o el método Object.setPrototypeOf() y Object.getPrototypeOf().

Por ejemplo:

```
// Definir un objeto persona
let persona = {
  nombre: 'Ana',
  edad: 25
};

// Acceder a una propiedad del objeto persona
console.log(persona.nombre); // Salida: Ana

// Agregar un método al prototipo de persona
persona.saludar = function() {
  console.log(`Hola, soy ${this.nombre}`);
};

// Crear un nuevo objeto que tiene a persona como prototipo
let otraPersona = Object.create(persona);
otraPersona.nombre = 'Carlos';

// Acceder a un método del prototipo de persona a través de otraPersona
otraPersona.saludar(); // Salida: Hola, soy Carlos
```

En este ejemplo, otraPersona se crea con persona como su prototipo. Cuando llamamos al método saludar() en otraPersona, JavaScript busca en otraPersona, no lo encuentra, y luego busca en su prototipo (persona) y lo encuentra allí.

Los prototipos son esenciales para la herencia en JavaScript y permiten la reutilización de propiedades y métodos entre objetos, lo que hace que el código sea más eficiente y fácil de mantener. Sin embargo, con la introducción de las clases en ES6, el uso directo de los prototipos es menos común, ya que las clases proporcionan una sintaxis más familiar para trabajar con la herencia y los objetos.

Resumen

En esta Unidad…

Trabajamos con Paradigmas

En la próxima Unidad…

Trabajaremos con clases