

Node.JS

Clase 4: Objetos, Clases y Operadores Avanzados en JavaScript

¡Les damos la bienvenida! 🚀

 Vamos a comenzar a grabar la clase.

Índice

1

Objetos, clases y operadores avanzados

- **Objetos:**
 - Definición/ Características de los objetos
 - Tipos de objetos
- **Clases:**
 - Sugar Syntax
 - Similitud con objetos funcionales
 - Método constructor
- **Operadores avanzados:**
 - Destructuring operator - Spread operator

2

Módulos y gestores de paquetes

- **Gestores de paquetes:**
- **NPM - Node Package Manager:**
 - Instalación de paquetes
 - Creación de scripts
 - Gestión de dependencias.
- **Inicio de un proyecto nuevo con Node Js y NPM**
- **Módulos Nativos**
- **Módulos de terceros**
- **Dirname: Gestión de rutas absolutas y acceso a archivos del servidor**

Objetivos de la Clase

1 Comprender Objetos

Aprenderemos a crear y manipular objetos literales y funcionales en JavaScript.

2 Introducir Clases

Exploraremos la sintaxis y uso de clases en JavaScript.

3 Operadores Avanzados

Estudiaremos el destructuring y spread operator para mejorar nuestro código.

Objetos

Objetos

Los objetos son variables especiales que contienen más variables en su interior.

Estructura

Cada **propiedad** o elemento es un par **clave/valor** separado de otro a través de una coma.

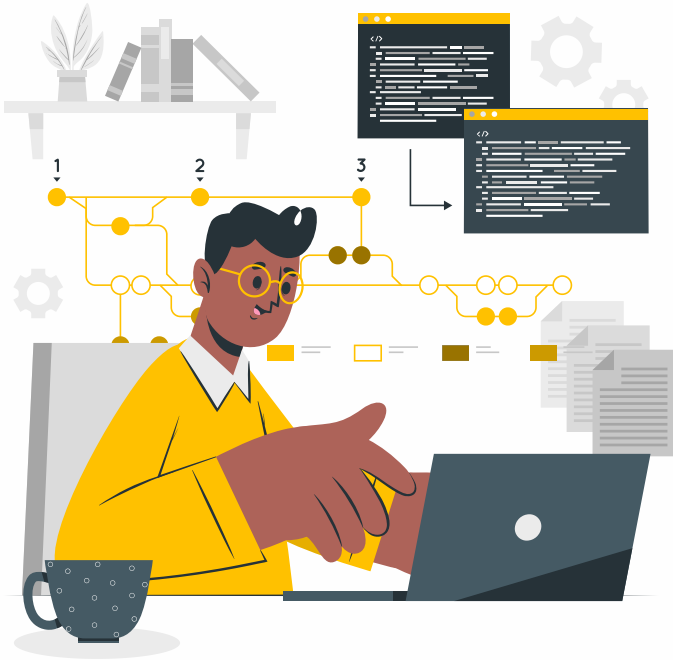
Ventaja

Excelente para crear colecciones con información relacionada e identificada por claves específicas.

```
const hero = {  
  alias: 'Cody',  
  universe: 'TechLab',  
  powers: ['fly', 'nightvision', 'stregth'],  
  hit: 128,  
  vitality: 100,  
  decreaseVitality: function(damage) {  
    return this.vitality - damage  
  }  
}
```

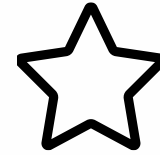


Características de los Objetos



Pares Clave/Valor

Cada dato está asociado a una clave única llamada propiedad.



Dinamismo

Los objetos pueden expandirse o modificarse en tiempo de ejecución.



Métodos

Los objetos pueden incluir funciones como valor de sus propiedades.

Objetos Literales

Definición

Se declaran asignando a una variable un par de llaves {} que contienen propiedades.

Acceso

Se accede a las propiedades usando el . (punto) o corchetes ['propiedad'].

Objetos Funcionales

Se declaran como una función de JavaScript tradicional, actuando como un molde para crear múltiples objetos.

Función Constructora

Utiliza la palabra reservada 'this' para definir propiedades y métodos.

Creación de Instancias

Se utilizan con la palabra reservada 'new' para crear nuevos objetos basados en el molde.



Classes

Clases en JavaScript

Las clases en JavaScript ofrecen una sintaxis más clara para la programación orientada a objetos.

Definición de Clase

Usa la palabra reservada 'class' y el método constructor().

Herencia

Permite extender clases con 'extends'.

Instanciación

Crea objetos con 'new'.



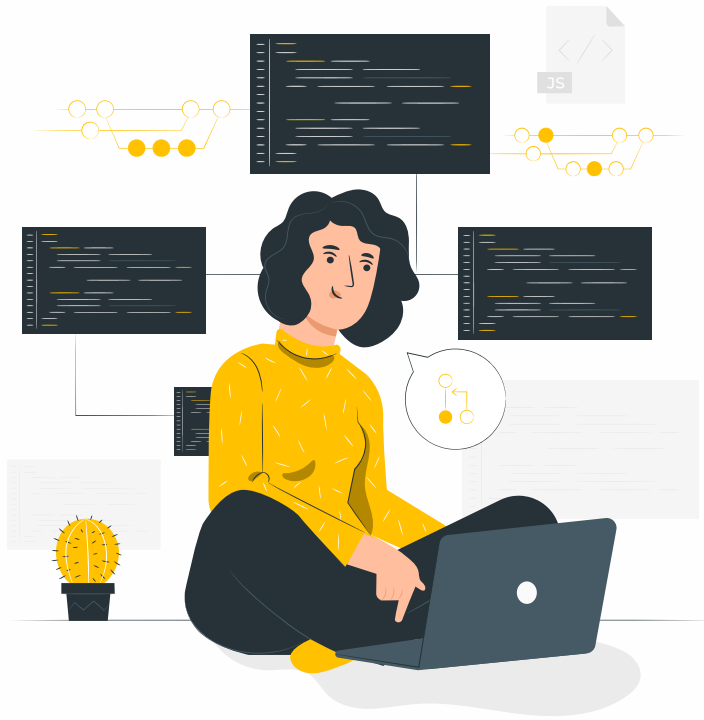
POO (La programación orientada a objetos)

Javascript es un lenguaje multiparadigma, lo que significa que soporta múltiples metodologías de programación, como la programación imperativa, la funcional o la programación orientada a objetos muy común en lenguajes como **Java** o **C#**.

La programación orientada a objetos o **POO** se basa en la creación de **Clases** que definen propiedades y comportamientos (métodos) de subclases o clases hijas que heredan estas definiciones, de hecho, los objetos funcionales vistos en la sección anterior, son una representación básica de lo que las clases pueden lograr.

***Nota:** La **sugar syntax** es una forma simplificada y más legible de escribir código, que enmascara una implementación más compleja sin cambiar su funcionalidad subyacente.

Operadores avanzados



Destructuring Operator

`{a, b} = params;`

Extrae valores de arrays o propiedades de objetos y los asigna a variables.

Arrays

Desempaqueta valores basados en su posición en el array.

Objetos

Extrae propiedades y las asigna a variables con el mismo nombre.

Spread Operator {...}

Extrae valores de arrays o propiedades de objetos y los asigna a variables.

Uso en Arrays

Combina o copia arrays sin modificar el original.

Uso en Objetos

Combina o copia objetos de manera similar a los arrays.



Próximos Pasos y Recursos

Próximos Temas

- Módulos y librerías
- Manejo de Promesas
- Servidores Web

Recursos Adicionales

- [Documentación MDN sobre Objetos](#)
- [Guía sobre funciones constructoras](#)
- [Documentación sobre Clases en JavaScript](#)

Ejercicio Práctico

Storytelling

Matías y Sabrina han preparado un nuevo reto para evaluar tus habilidades.



- ① “Imagina que estás organizando información sobre una flota de vehículos”, dice Sabrina.
“Queremos ver cómo manejas datos más estructurados”.

Ejercicio Práctico

Obligatorio

Misión 1:

1) Crea un array con 10 objetos, donde cada objeto represente un automóvil con la siguiente información:

- Marca
- Modelo
- Año
- Color

2) Usa un método de array para recorrer la lista e imprime por consola todos los datos de los automóviles cuyo **año sea mayor a 2018**.

Matías añade: "Queremos que veas esto como un ejercicio para prepararte para trabajar con datos reales en el futuro".

Ejercicio Práctico

Obligatorio

Impresionados con tu avance, Matías y Sabrina suben un poco la dificultad. Sabrina te plantea:



“Queremos saber si puedes analizar la información de forma específica. Aquí tienes tu próximo desafío”.

Misión 2:

1. Crea una función que recorra el array de automóviles.
2. Usa **destructuring** dentro de la función para obtener el color de cada automóvil.
3. La función debe aceptar un color como parámetro y devolver por consola cuántos automóviles tienen ese color.

Ejercicio Práctico

Obligatorio



“Este tipo de habilidad es esencial para manejar sistemas dinámicos”, explica Matías.

¿Estás listo para superar esta prueba y demostrar tu capacidad para manipular datos con precisión? ¡Adelante!



¡Nuevo cuestionario en Campus!



No olvides que los cuestionarios son de carácter obligatorio para poder avanzar con la cursada.