

CURSO DE JAVASCRIPT

**Corporación Ecuatoriana para el Desarrollo de la
Investigación y la Academia**

Ing. María Fernanda Granda J., PhD

11-11-2025

**“Conectando ideas,
transformando
sociedades”**

Módulo 3: Programación Orientada a Objetos

- Introducción
- ¿Qué es un objeto?
- Tipos de Objetos
- Creación de Objetos/Clase
- Instanciación de una clase
- Métodos de Objetos
- Acceder a propiedades y métodos
- Objetos anidados
- Recolección de basura en JavaScript
- Eliminación de propiedades de los objetos

Introducción

- Javascript es un lenguaje basado en objetos.
- Todo es un objeto (excepto los constructos del lenguaje, palabras claves y operadores)
- Los objetos juegan muchos roles desde representar datos hasta la manipulación de documentos HTML vía el modelo de objeto de documento (DOM), para las interfaces con el navegador y más...

¿Qué es un objeto?

- Es una colección desordenada de datos, incluyendo datos primitivos, funciones y otros objetos.
- Es una colección de propiedades, cada una de las cuales tiene un nombre y un valor.
- Los objetos de Javascript son dinámicos –las propiedades pueden ser añadidas y eliminadas.

Tipos de Objetos

- Objetos definidos por el usuario
- Objetos Nativos
- Objetos Host
- Objetos Documento

Objetos definidos por el usuario

- Objetos personalizados creados por el programador
- Pueden ser incluidos dentro de otros objetos y esto permite crear una estructura de datos compleja (compuesta) consistiendo de datos y métodos.

Objetos Nativos

- Incluyen objetos asociados con tipos de datos tales como String, Número y boolean, así como también que permitan la creación de objetos definidos por el usuario y tipos compuestos.
- Incluyen funciones de Javascript, Date, Math y manipulación RegExp.
- Manejo de excepciones y errores son también objetos nativos.
- Son regidos por el estándar del lenguaje ECMAScript.

Objetos Host

- Son soportados por ambientes host típicamente como un navegador.
- Objetos host basados en el navegador incluyen window, navigator, XMLHttpRequest, location, History, etc.

Objetos Documento

- Son parte del Document Object Model (DOM), definido por la W3C.

Creando Objetos

- Pueden ser creados de dos maneras:
 - Objeto Literal
 - Función Constructor

Objeto Literal

- Creando el objeto al definir propiedades y valores:

Sintaxis:

```
let nombreObjeto={}; //Objeto Simple
```

- Ejemplo:

```
let alumno={nombre:"Ana",curso:"JS",nota:10};
```

Métodos de los Objetos

- Son acciones que pueden ser ejecutadas por los objetos.
- Un método de JavaScript es una propiedad conteniendo una definición de función.

Acceso a Métodos de Objetos

- Creando métodos del objeto

```
nombreMetodo: function ()  
{  
    ... código  
}
```

- Accediendo a métodos del objeto

```
nombreObjeto.nombreMetodo();
```

Función Constructor

- Es una forma de crear objetos
- Reglas:
 - Nombre de la función constructora debe ser igual al del objeto.
 - El operador `this` es usado en la función constructora para referirse al objeto.
 - No debe haber un `return` en el función constructora.

Ejemplo de Función Constructor

```
function Estudiante(pnombre, pcurso, pnota){  
    this.nombre=pnombre;  
    this.curso=pcurso;  
    this.nota=pnota;  
}  
estudiante= new Estudiante ("Ana", "JavaScript", 20);
```

Ejemplo de Objeto

- Con funciones incluidas.

```
let alumno={  
    nombre:"Sandra",  
    apellido:"Bullock",  
    curso:"JavaScript",  
    nota:95,  
    nombreCompleto:function(){  
        return this.nombre+" "+this.apellido;  
    }  
};
```

Accediendo a las propiedades del Objeto

- Se puede hacer de dos formas:
 1. **Notación Punto:** es más concisa y se utiliza para propiedades con nombres de variable válidos.
 2. **Notación de corchetes:** es más flexible y se usa para propiedades con nombres no válidos o cuando el nombre de la propiedad se almacena en una variable.

Accediendo a las propiedades del Objeto

Sintaxis:

nombreObjeto.nombrePropiedad;

o

nombreObjeto["nombrePropiedad"];

Ejemplos:

```
alert(estudiante.nombre+ " "+estudiante.curso+ " "+estudiante.nota);
```

```
alert(estudiante["nombre"]+ " "+estudiante["curso"]+ " "+estudiante["nota"]);
```

Accediendo a las propiedades del Objeto

- Ejemplos de notación corchetes para propiedades con nombres no válidos o cuando el nombre de la propiedad se almacena en una variable.

```
const persona = {  
    "nombre completo": "Juan Pérez", // Nombre con espacio  
    "fecha de nacimiento": "1993-01-15" // Nombre con guion  
};  
  
// Acceder a La propiedad "nombre completo" usando La notación de corchetes  
console.log(persona["nombre completo"]); // Output: Juan Pérez  
  
// Acceder a La propiedad "fecha de nacimiento" usando La notación de corchetes  
console.log(persona["fecha de nacimiento"]); // Output: 1993-01-15  
  
// También puedes usar una variable para almacenar el nombre de La propiedad  
const propiedad = "edad";  
persona[propiedad] = 32;  
console.log(persona[propiedad]); // output: 32
```

Objetos Anidados

- Un objeto contiene a otro objeto.

```
const persona = {  
    nombre: "Juan",  
    direccion: {  
        calle: "Avenida Principal",  
        numero: 123  
    }  
};
```

// Acceder a la propiedad "calle" usando la notación de punto
`console.log(persona.direccion.calle); // Output: Avenida Principal`

// Acceder a la propiedad "numero" usando la notación de corchetes
`console.log(persona["direccion"]["numero"]); // Output: 123`

Recolección de basura de Objetos

- Javascript utiliza una recolección de basura para automáticamente eliminar objetos cuando ellos no son referenciados.

```
let estudiante= new Estudiante ("Ana", "JavaScript", 20);
estudiante=null;
```

Eliminando Propiedades de los Objetos

- El operador delete puede ser usado para eliminar completamente propiedades de un objeto.
- Estableciendo la propiedad a undefined o null sólo cambia el valor de una propiedad pero no la elimina.

Sintaxis:

```
delete nombreObjeto.nombrePropiedad;
```

Tarea 2

- Crear un programa JavaScript para crear un objeto producto, agregue la propiedad nombreProducto, descripción del producto, cantidad, unidad y precio.
- Acceda a todas las propiedades y muéstrelas. Utilice las dos formas de acceso con punto y con corchetes.
- Eliminar la propiedad unidad.

GRACIAS