

GUÍA TEÓRICO-PRÁCTICA:

UNIDAD I

SEMANA 5: OBJETOS Y CSS

TEMA 5.1: OBJETOS BÁSICOS EN JS

Teoría

Objetos literales:

```
javascript
```

```
1 let persona = {
2   nombre: "Ana",
3   edad: 25,
4   saludar: function() {
5     return "Hola, soy " + this.nombre;
6   }
7 };
8
9 console.log(persona.saludar()); // "Hola, soy Ana"
```

Constructor de objetos:

```
javascript
```

```
1 function Persona(nombre, edad) {
2   this.nombre = nombre;
3   this.edad = edad;
4   this.saludar = function() {
5     return "Hola, soy " + this.nombre;
6   };
7 }
8
9 let juan = new Persona("Juan", 30);
10 console.log(juan.saludar());
```

Pregunta de reflexión 17: ¿Qué es this en JavaScript?

Ejemplo

javascript

```

1 // Objeto literal
2 let libro = {
3     titulo: "JavaScript Avanzado",
4     autor: "Carlos Pérez",
5     anio: 2024,
6     disponible: true,
7     prestar: function() {
8         if (this.disponible) {
9             this.disponible = false;
10            return "Libro prestado";
11        }
12        return "Libro no disponible";
13    }
14 };
15
16 console.log(libro.prestar());
17 console.log(libro.disponible); // false
18
19 // Objeto con métodos
20 let calculadora = {
21     valor: 0,
22     sumar: function(n) { this.valor += n; return this; },
23     restar: function(n) { this.valor -= n; return this; },
24     multiplicar: function(n) { this.valor *= n; return this; },
25     resultado: function() { return this.valor; }
26 };
27
28 console.log(calculadora.sumar(5).multiplicar(2).resultado()); // 20

```

Ejercicio 5.1

Crea un objeto cuentaBancaria con:

1. Propiedades: titular, saldo
2. Métodos: depositar(cantidad), retirar(cantidad), consultarSaldo()
3. Valida que no se pueda retirar más del saldo disponible
4. Crea 2 cuentas diferentes y realiza operaciones

```
javascript
```

```
1 // Tu código aquí:  
2  
3  
4  
5
```

TEMA 5.2: PROPIEDADES Y MÉTODOS

Teoría

Propiedades: Atributos o características de un objeto.

Métodos: Funciones que pertenecen a un objeto.

Acceso dinámico:

```
javascript
```

```
1 let obj = { clave: "valor" };  
2 let propiedad = "clave";  
3 console.log(obj[propiedad]); // "valor"
```

Pregunta de reflexión 18: ¿Cómo iteras sobre las propiedades de un objeto?

Ejemplo

```
javascript
```

```
1  let producto = {
2      nombre: "Laptop",
3      precio: 1000,
4      stock: 10,
5      descuento: 0.1,
6
7      precioConDescuento: function() {
8          return this.precio * (1 - this.descuento);
9      },
10
11     agregarStock: function(cantidad) {
12         this.stock += cantidad;
13     },
14
15     vender: function(cantidad) {
16         if (cantidad <= this.stock) {
17             this.stock -= cantidad;
18             return true;
19         }
20         return false;
21     },
22
23     getInfo: function() {
24         return `${this.nombre} - ${this.precioConDescuento()}
25         (Stock: ${this.stock})`;
26     }
27 };
28
29 // Iterar propiedades
30 for (let clave in producto) {
31     if (typeof producto[clave] !== 'function') {
32         console.log(clave + ":", producto[clave]);
33     }
34 }
35
36 console.log(producto.getInfo());
```

Ejercicio 5.2

Crea un objeto estudiante con:

1. Propiedades: nombre, calificaciones (array), materia
2. Métodos:
 - agregarCalificacion(nota)
 - calcularPromedio()
 - obtenerEstado() (retorna "Aprobado" si promedio \geq 60)
 - mostrarInformacion()
3. Crea 3 estudiantes y muestra sus promedios

```
javascript
```

```

1 // Tu código aquí:
2
3
4
5

```

TEMA 5.3: INTERACTIVIDAD CON CSS

Teoría

Manipular CSS con JavaScript:

```
javascript
```

```

1 // style.property
2 elemento.style.color = "red";
3 elemento.style.fontSize = "20px";
4
5 // classList
6 elemento.classList.add("activo");
7 elemento.classList.remove("inactivo");
8 elemento.classList.toggle("visible");
9
10 // getComputedStyle
11 let estilo = window.getComputedStyle(elemento);
12 let color = estilo.color;


```

Pregunta de reflexión 19: ¿Cuál es la ventaja de usar clases CSS en lugar de modificar estilos directamente?

Ejemplo

html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <style>
5     .caja {
6       width: 100px;
7       height: 100px;
8       background: blue;
9       transition: all 0.3s;
10    }
11    .caja.grande {
12      width: 200px;
13      height: 200px;
14    }
15    .caja.rojo {
16      background: red;
17    }
18    .caja.oculto {
19      opacity: 0;
20      transform: scale(0);
21    }
22    .animar {
23      animation: mover 2s infinite;
24    }
25    @keyframes mover {
26      from { transform: translateX(0); }
27      to { transform: translateX(100px); }
28    }
29  </style>
```



```
30 </head>
31 <body>
32   <div id="miCaja" class="caja"></div>
33   <button onclick="agrandar()">Agrandar</button>
34   <button onclick="cambiarColor()">Color</button>
35   <button onclick="toggleOculto()">Ocultar</button>
36   <button onclick="animar()">Animar</button>
37
38   <script>
39     let caja = document.getElementById("miCaja");
40
41     function agrandar() {
42       caja.classList.toggle("grande");
43     }
44
45     function cambiarColor() {
46       caja.classList.toggle("rojo");
47     }
48
49     function toggleOculto() {
50       caja.classList.toggle("oculto");
51     }
52
53     function animar() {
54       caja.classList.toggle("animar");
55     }
56   </script>
57 </body>
58 </html>
```

Ejercicio 5.3

Crea una página con:

1. Un elemento que cambie de color al pasar el mouse
2. Un botón que active/desactive un modo oscuro (cambia fondo y texto)
3. Un slider que controle el tamaño de un elemento
4. Una animación que se active al hacer clic

```
javascript
```

```
1 // Tu código aquí:  
2  
3  
4  
5
```

TEMA 5.4: DISEÑO VISUAL DE APLICACIÓN

Teoría

Principios de diseño:

- Consistencia visual
- Feedback al usuario
- Jerarquía visual
- Espaciado y alineación

Pregunta de reflexión 20: ¿Qué hace que una interfaz sea "usable"?

Ejemplo

Aplicación completa de notas con diseño profesional.

Ejercicio 5.4

Diseña una aplicación de:

- Lista de contactos
- Calculadora visual
- Reloj digital Elige una y aplícale:
 1. Diseño responsive
 2. Animaciones CSS
 3. Validaciones visuales
 4. Feedback al usuario

javascript

```
1 // Tu código aquí:  
2  
3  
4  
5
```