

GUÍA TEÓRICO-PRÁCTICA:

UNIDAD I

SEMANA 3: ESTRUCTURAS DE CONTROL

TEMA 3.1: CONDICIONALES - if, else, switch

Teoría

Estructuras condicionales: Permiten ejecutar diferentes bloques de código según se cumplan o no ciertas condiciones.

if:

```
javascript
```

```
1 if (condicion) {  
2 |   // Código si la condición es verdadera  
3 }
```

if-else:

```
javascript
```

```
1 if (condicion) {  
2 |   // Código si es verdadero  
3 } else {  
4 |   // Código si es falso  
5 }
```

if-else if-else:

```
javascript
```

```
1 if (condicion1) {  
2 |   // Código si condicion1 es verdadera  
3 } else if (condicion2) {  
4 |   // Código si condicion2 es verdadera  
5 } else {  
6 |   // Código si ninguna es verdadera  
7 }
```

switch:

```
javascript
```

```

1  switch(expresion) {
2      case valor1:
3          // Código si expresion === valor1
4          break;
5      case valor2:
6          // Código si expresion === valor2
7          break;
8      default:
9          // Código si no coincide ningún caso
10 }

```

Pregunta de reflexión 9: ¿Cuándo es mejor usar switch en lugar de múltiples if-else?

Ejemplo

```
javascript
```

```

1  // Ejemplo if-else
2  let edad = 18;
3
4  if (edad >= 18) {
5      console.log("Eres mayor de edad");
6  } else {
7      console.log("Eres menor de edad");
8  }
9
10 // Ejemplo if-else if
11 let nota = 85;
12 let calificacion;
13
14 if (nota >= 90) {
15     calificacion = "A";
16 } else if (nota >= 80) {
17     calificacion = "B";
18 } else if (nota >= 70) {
19     calificacion = "C";
20 } else if (nota >= 60) {
21     calificacion = "D";
22 } else {
23     calificacion = "F";
24 }
25 console.log("Calificación:", calificacion);

```

```

26
27 // Ejemplo switch
28 let dia = 3;
29 let nombreDia;
30
31 switch(dia) {
32     case 1:
33         nombreDia = "Lunes";
34         break;
35     case 2:
36         nombreDia = "Martes";
37         break;
38     case 3:
39         nombreDia = "Miércoles";
40         break;

```

```

41 |     case 4:
42 |         nombreDia = "Jueves";
43 |         break;
44 |     case 5:
45 |         nombreDia = "Viernes";
46 |         break;
47 |     default:
48 |         nombreDia = "Fin de semana";
49 | }
50 | console.log("Día:", nombreDia);
51 |
52 | // Switch con strings
53 | let fruta = "manzana";
54 | switch(fruta) {
55 |     case "manzana":
56 |         console.log("La manzana es roja");
57 |         break;
58 |     case "banana":
59 |         console.log("La banana es amarilla");
60 |         break;
61 |     default:
62 |         console.log("Fruta desconocida");
63 | }

```

Ejercicio 3.1

Crea un programa que:

1. Pida al usuario un número del 1 al 7 y use switch para mostrar el día de la semana
2. Pida la edad y use if-else para determinar si es niño (0-12), adolescente (13-17), adulto (18-64) o adulto mayor (65+)
3. Pida una calificación (0-100) y determine si está aprobado (≥ 60) o reprobado

```
javascript
```

```

1 // Tu código aquí:
2
3
4
5
6

```

TEMA 3.2: OPERADORES LÓGICOS (&&, ||)

Teoría

Operadores lógicos: Permiten combinar múltiples condiciones.

AND (&&):

Ambas condiciones deben ser verdaderas.

javascript

```
1 if (edad >= 18 && tieneLicencia) {  
2 |   console.log("Puede conducir");  
3 }
```

OR (||):

Al menos una condición debe ser verdadera.

javascript

```
1 if (esMiembro || tieneCupon) {  
2 |   console.log("Tiene descuento");  
3 }
```

NOT (!):

Invierte el valor booleano.

javascript

```
1 if (!estaConectado) {  
2 |   console.log("No hay conexión");  
3 }
```

Combinaciones:

javascript

```
1 if ((edad >= 18 && tieneLicencia) || esConductorProfesional) {  
2 |   console.log("Puede trabajar como conductor");  
3 }
```

Pregunta de reflexión 10: ¿Qué resultado da true && false || true? Explica el orden de evaluación.

Ejemplo

javascript

```
1 // Ejemplos AND
2 let edad = 25;
3 let tieneLicencia = true;
4
5 if (edad >= 18 && tieneLicencia) {
6   console.log("Puede conducir legalmente");
7 }
8
9 // Ejemplos OR
10 let dia = "sábado";
11
12 if (dia === "sábado" || dia === "domingo") {
13   console.log("¡Es fin de semana!");
14 }
15
16 // Ejemplos NOT
17 let estaApagado = false;
18
19 if (!estaApagado) {
20   console.log("El sistema está encendido");
21 }
22
23 // Combinación compleja
24 let usuario = "admin";
25 let password = "12345";
26 let intentos = 2;
27
28 if (usuario === "admin" && password === "12345" && intentos < 3) {
29   console.log("Acceso concedido");
30 } else {
31   console.log("Acceso denegado");
32 }
33
34 // Tabla de verdad
35 console.log("true && true:", true && true); // true
36 console.log("true && false:", true && false); // false
37 console.log("true || false:", true || false); // true
38 console.log("false || false:", false || false); // false
39 console.log("!true:", !true); // false
40 console.log("!false:", !false); // true
```

Ejercicio 3.2

Crea un sistema de validación que:

1. Verifique si un usuario puede entrar a una película (edad ≥ 18 O si viene con adulto)
2. Valide un formulario: el email NO debe estar vacío Y debe contener "@"
3. Determine si hace buen día: temperatura entre 20-30°C Y no llueve O es fin de semana

```
javascript
```

```
1 // Tu código aquí:
2
3
4
5
6
```

TEMA 3.3: RECORRER ARRAYS CON BUCLES

Teoría

Arrays (Arreglos):

Colección ordenada de valores .

```
javascript
```

```
1 let frutas = ["manzana", "banana", "naranja"];
2 let numeros = [1, 2, 3, 4, 5];
3 let mixto = ["texto", 42, true, null];
```

Acceso a elementos:

```
javascript
```

```
1 let primera = frutas[0]; // "manzana"
2 let ultima = frutas[frutas.length - 1]; // "naranja"
```

Recorrer arrays con for:

```
javascript
```

```
1 for (let i = 0; i < frutas.length; i++) {
2 |   console.log(frutas[i]);
3 }
```

Recorrer arrays con forEach:

javascript

```

1 frutas.forEach(function(fruta) {
2 |   console.log(fruta);
3 | });
4
5 // O con arrow function
6 frutas.forEach(fruta => console.log(fruta));

```

Pregunta de reflexión 11: ¿Cuál es la diferencia entre for y forEach?
¿Cuándo usar cada uno?

Ejemplo

javascript

```

1 // Crear array
2 let colores = ["rojo", "verde", "azul", "amarillo"];
3
4 // Acceder a elementos
5 console.log(colores[0]); // "rojo"
6 console.log(colores.length); // 4
7
8 // Modificar elementos
9 colores[1] = "naranja";
10 console.log(colores); // ["rojo", "naranja", "azul", "amarillo"]
11
12 // Agregar elementos
13 colores.push("morado"); // Agrega al final
14 colores.unshift("rosa"); // Agrega al inicio
15 console.log(colores);
16
17 // Recorrer con for
18 console.log("=== Usando for ===");
19 for (let i = 0; i < colores.length; i++) {
20 |   console.log(i + ": " + colores[i]);
21 | }
22

```

```

23 // Recorrer con forEach
24 console.log("=== Usando forEach ===");
25 colores.forEach((color, indice) => {
26 |   console.log(indice + ": " + color);
27 | });
28
29 // Operaciones con arrays
30 let numeros = [1, 2, 3, 4, 5];
31
32 // Sumar todos los elementos
33 let suma = 0;
34 numeros.forEach(num => suma += num);
35 console.log("Suma:", suma); // 15
36
37 // Encontrar el mayor
38 let mayor = numeros[0];
39 for (let num of numeros) {
40 |   if (num > mayor) {
41 |     mayor = num;
42 |   }
43 | }
44 console.log("Mayor:", mayor); // 5

```

Ejercicio 3.3

Crea un programa que:

1. Declare un array con 5 nombres
2. Recorra el array con for y muestre cada nombre
3. Recorra el array con forEach y muestre los nombres en mayúsculas
4. Encuentre y muestre el nombre más largo
5. Cuente cuántos nombres tienen más de 5 letras

```
javascript
```

```


1 // Tu código aquí:
2
3
4
5
6

```



```
26     let texto = input.value.trim();
27
28     if (texto === "") {
29         alert("Ingresa una tarea");
30         return;
31     }
32
33     let tarea = {
34         id: Date.now(),
35         texto: texto,
36         completada: false
37     };
38
39     tareas.push(tarea);
40     input.value = "";
41     renderizarTareas();
42 }
43
44 function toggleTarea(id) {
45     let tarea = tareas.find(t => t.id === id);
46     tarea.completada = !tarea.completada;
47     renderizarTareas();
48 }
49
50 function eliminarTarea(id) {
51     tareas = tareas.filter(t => t.id !== id);
52     renderizarTareas();
53 }
54
55 function renderizarTareas() {
56     let contenedor = document.getElementById("listaTareas");
57     contenedor.innerHTML = "";
58
59     tareas.forEach(tarea => {
60         let div = document.createElement("div");
61         div.className = "tarea";
62         div.innerHTML = `
63             <span class="${tarea.completada ? 'completada' : ''}">
64                 ${tarea.texto}
65             </span>
66             <div>
67                 <button onclick="toggleTarea(${tarea.id})">
68                     ${tarea.completada ? '↩' : '✓'}
69                 </button>
```

```
70 |         <button onclick="eliminarTarea(${tarea.id})">X</button>
71 |     </div>
72 |     `;
73 |     contenedor.appendChild(div);
74 | });
75 |
76 |     document.getElementById("contador").textContent = tareas.length;
77 | }
78 | </script>
79 | </body>
80 | </html>
```



Ejercicio 3.4

Mejora la lista de tareas agregando:

1. Un campo para priorizar tareas (alta, media, baja)
2. Un filtro para mostrar solo tareas pendientes o completadas
3. La fecha de creación de cada tarea
4. Un botón para eliminar todas las tareas completadas

javascript

```
1 // Tu código aquí:
2
3
4
5
6
```