

GUÍA TEÓRICO-PRÁCTICA:

UNIDAD I

SEMANA 2: EVENTOS Y FUNCIONES

TEMA 2.1: EVENTOS BÁSICOS - onclick, onload

Teoría

Eventos: Los eventos son acciones que ocurren en el navegador que JavaScript puede detectar y responder .

Eventos principales:

onclick: Se activa cuando el usuario hace clic en un elemento.

```
html
```

```
1 <button onclick="miFuncion()">Haz click</button>
2
3 <script>
4 function miFuncion() {
5 |   alert("Button clicked!");
6 }
7 </script>
```

onload: Se activa cuando la página o un elemento ha terminado de cargar.

```
html
```

```
1 <body onload="inicializar()">
2 |   <h1>Mi Página</h1>
3 </body>
4
5 <script>
6 function inicializar() {
7 |   console.log("Página cargada completamente");
8 }
9 </script>
```

Otros eventos comunes:

- **onmouseover/onmouseout:** Cuando el mouse entra/sale de un elemento
- **onfocus/onblur:** Cuando un campo recibe/pierde el foco

- **onchange:** Cuando cambia el valor de un input
- **onkeydown/onkeyup:** Cuando se presiona/libera una tecla
- **onsubmit:** Cuando se envía un formulario

Pregunta de reflexión 5: ¿Cuál es la diferencia entre onclick en HTML y addEventListener? (Lo veremos más adelante)

Ejemplo

html

```
1 <!DOCTYPE html>
2 <html>
3 <body onload="bienvenida()">
4   <h1 id="titulo">Eventos en JavaScript</h1>
5
6   <button onclick="cambiarColor()">Cambiar Color</button>
7   <button onmouseover="agrandar()" onmouseout="normalizar()">
8     | Pasa el mouse
9   </button>
10
11 <div id="caja" style="width:100px; height:100px; background:blue;">
12 </div>
13
14 <script>
15   function bienvenida() {
16     | console.log("Bienvenido a la página");
17   }
18
19   function cambiarColor() {
20     | let colores = ["red", "green", "yellow", "purple"];
21     | let random = Math.floor(Math.random() * colores.length);
22     | document.getElementById("titulo").style.color = colores[random];
23   }
24
25   function agrandar() {
26     | document.getElementById("caja").style.width = "150px";
27     | document.getElementById("caja").style.height = "150px";
28   }
```

```

29
30     function normalizar() {
31         document.getElementById("caja").style.width = "100px";
32         document.getElementById("caja").style.height = "100px";
33     }
34 </script>
35 </body>
36 </html>

```

Ejercicio 2.1

Crea una página HTML que:

1. Muestre un alert() de bienvenida cuando cargue la página (onload)
2. Tenga un botón que al hacer clic (onclick) cambie el texto de un párrafo
3. Tenga una imagen que cambie de tamaño cuando pases el mouse sobre ella (onmouseover/onmouseout)

html

```

1 <!DOCTYPE html>
2 <html>
3 <body onload="">
4     <h1>Mis Eventos</h1>
5     <p id="miParrafo">Texto original</p>
6
7     <button onclick="">Cambiar texto</button>
8
9     
12
13     <script>
14         function bienvenida() {
15
16         }
17
18         function cambiarTexto() {
19
20         }
21
22         function cambiarTamaño() {
23
24         }
25     </script>
26 </body>
27 </html>

```

TEMA 2.2: FUNCIONES EN JS - DEFINICIÓN Y LLAMADA

Teoría

Funciones: Una función es un bloque de código reutilizable que realiza una tarea específica .

Formas de declarar funciones:

1. Declaración de función (Function Declaration):

```
javascript

1 function saludar(nombre) {
2   |   return "Hola " + nombre;
3 }
4
5 let mensaje = saludar("María");
6 console.log(mensaje); // "Hola María"
```

2. Expresión de función (Function Expression):

```
javascript

1 const sumar = function(a, b) {
2   |   return a + b;
3 };
4
5 let resultado = sumar(5, 3);
6 console.log(resultado); // 8
```

3. Arrow Function (ES6):

```
javascript

1 const multiplicar = (a, b) => {
2   |   return a * b;
3 };
4
5 // O más corto:
6 const dividir = (a, b) => a / b;
7
8 console.log(multiplicar(4, 2)); // 8
```

Partes de una función:

- **Palabra clave:** function
- **Nombre:** Identificador de la función
- **Parámetros:** Variables que recibe la función
- **Cuerpo:** Código entre llaves {}
- **Retorno:** Valor que devuelve (return)

Pregunta de reflexión 6: ¿Cuál es la diferencia entre console.log() y return en una función?

Ejemplo

```

javascript

1 // Función sin parámetros ni retorno
2 function saludar() {
3   console.log("¡Hola!");
4 }
5 saludar(); // Llamada a la función
6
7 // Función con parámetros
8 function sumar(a, b) {
9   let resultado = a + b;
10  console.log("La suma es:", resultado);
11 }
12 sumar(5, 3); // 8
13
14 // Función con retorno
15 function restar(a, b) {
16   return a - b;
17 }
18 let diferencia = restar(10, 4);
19 console.log(diferencia); // 6
20
21 // Función con parámetros por defecto
22 function despedir(nombre = "amigo") {
23   return "Adiós " + nombre;
24 }
25 console.log(despedir()); // "Adiós amigo"
26 console.log(despedir("Carlos")); // "Adiós Carlos"
27
28 // Arrow function
29 const cuadrado = (x) => x * x;
30 console.log(cuadrado(5)); // 25

```



Ejercicio 2.2

Crea las siguientes funciones:

1. saludar(nombre) - Retorne "Hola [nombre], bienvenido"
2. calcularIMC(peso, altura) - Retorne el IMC (peso / altura²)
3. esMayorDeEdad(edad) - Retorne true si edad >= 18, false si no
4. calcularAreaCirculo(radio) - Retorne el área ($\pi \times \text{radio}^2$)

```
javascript
```

```

1 // Tu código aquí:
2
3
4
5
6

```

TEMA 2.3: USO DE getElementById Y innerHTML

Teoría

getElementById(): Método que selecciona un elemento del DOM por su atributo id .

```
javascript
```

```
1 let elemento = document.getElementById("miId");
```

innerHTML: Propiedad que obtiene o establece el contenido HTML dentro de un elemento .

```
javascript
```

```

1 // Leer contenido
2 let contenido = document.getElementById("parrafo").innerHTML;
3
4 // Modificar contenido
5 document.getElementById("parrafo").innerHTML = "Nuevo texto";

```

Otros métodos de selección:

- `getElementsByName()`: Selecciona por clase

- `getElementsByName()`: Selecciona por etiqueta
- `querySelector()`: Selecciona usando selectores CSS
- `querySelectorAll()`: Selecciona todos los elementos que coincidan

Pregunta de reflexión 7: ¿Por qué es importante que los IDs sean únicos en una página?

Ejemplo

```
html

1 <!DOCTYPE html>
2 <html>
3 <body>
4   <h1 id="titulo">Título Original</h1>
5   <p id="parrafo">Párrafo inicial</p>
6   <div id="contenedor"></div>
7
8   <button onclick="modificar()">Modificar Contenido</button>
9
10  <script>
11    function modificar() {
12      // Cambiar texto del h1
13      document.getElementById("titulo").innerHTML = "Título Modificado";
14
15      // Cambiar estilo
16      document.getElementById("titulo").style.color = "red";
17
18      // Agregar HTML complejo
19      document.getElementById("contenedor").innerHTML =
20        "<p>Nuevo <strong>párrafo</strong> con <em>formato</em></p>" +
21        "<ul><li>Item 1</li><li>Item 2</li></ul>";
22
23      // Leer contenido
24      let texto = document.getElementById("parrafo").innerHTML;
25      console.log("Contenido del párrafo:", texto);
26    }
27  </script>
28 </body>
29 </html>
```

Ejercicio 2.3

Crea una página con:

1. Un `<h1>` con `id="titulo"`

2. Un `<p>` con `id="descripcion"`
3. Un `<div>` con `id="resultado"`
4. Tres botones que:
 - Botón 1: Cambie el título a "JavaScript es genial"
 - Botón 2: Cambie la descripción a "Aprendiendo DOM"
 - Botón 3: Agregue una lista de 3 elementos en el div resultado

html

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4   <h1 id="titulo">Título Inicial</h1>
5   <p id="descripcion">Descripción inicial</p>
6   <div id="resultado"></div>
7
8   <button onclick="">Cambiar título</button>
9   <button onclick="">Cambiar descripción</button>
10  <button onclick="">Mostrar lista</button>
11
12  <script>
13    function cambiarTitulo() {
14
15    }
16
17    function cambiarDescripcion() {
18
19    }
20
21    function mostrarLista() {
22
23    }
24  </script>
25 </body>
26 </html>
```

TEMA 2.4: MANIPULACIÓN DEL DOM - CAMBIAR TEXTO Y COLORES

Teoría

Manipulación del DOM: El Document Object Model (DOM) es una representación en árbol de la estructura HTML que JavaScript puede modificar.

Cambiar texto:

```
javascript
```

```
1 // Usando innerHTML (puede contener HTML)
2 elemento.innerHTML = "<strong>Texto en negrita</strong>";
3
4 // Usando textContent (solo texto, más seguro)
5 elemento.textContent = "Texto plano";
6
7 // Usando innerText (similar a textContent pero respeta CSS)
8 elemento.innerText = "Texto visible";
```

Cambiar estilos CSS:

```
javascript
```

```
1 // Cambiar una propiedad
2 elemento.style.color = "red";
3 elemento.style.fontSize = "20px";
4 elemento.style.backgroundColor = "blue";
5
6 // Múltiples propiedades
7 elemento.style.cssText = "color: red; font-size: 20px;";
8
9 // Agregar/quitar clases
10 elemento.classList.add("miClase");
11 elemento.classList.remove("otraClase");
12 elemento.classList.toggle("activa");
```

Pregunta de reflexión 8: ¿Cuál es la diferencia entre innerHTML y textContent? ¿Cuándo usar cada uno?

Ejemplo

html

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <style>
5          .destacado {
6              background-color: yellow;
7              font-weight: bold;
8              padding: 10px;
9          }
10         .oculto {
11             display: none;
12         }
13     </style>
14 </head>
15 <body>
16     <h1 id="titulo">Manipulación del DOM</h1>
17     <p id="texto">Texto original</p>
18     <div id="caja" style="width: 200px; height: 200px; background: lightblue;">
19     </div>
20
21     <button onclick="cambiarTexto()">Cambiar Texto</button>
22     <button onclick="cambiarColor()">Cambiar Color</button>
23     <button onclick="agregarClase()">Agregar Clase</button>
24     <button onclick="toggleOculto()">Ocultar/Mostrar</button>
25
26     <script>
27         function cambiarTexto() {
28             document.getElementById("texto").innerHTML =
29             "Nuevo texto con <strong>formato</strong>";
30         }
31
32         function cambiarColor() {
33             let caja = document.getElementById("caja");
34             let colores = ["red", "green", "blue", "purple", "orange"];
35             let random = Math.floor(Math.random() * colores.length);
36             caja.style.backgroundColor = colores[random];
37         }
38
39         function agregarClase() {
40             document.getElementById("titulo").classList.add("destacado");
41         }
42
43         function toggleOculto() {
44             document.getElementById("texto").classList.toggle("oculto");
45         }
46     </script>
47 </body>
48 </html>

```

Ejercicio 2.4

Crea una página que permita:

1. Cambiar el color de fondo de la página aleatoriamente
2. Cambiar el tamaño de fuente de un párrafo (pequeño, mediano, grande)
3. Poner en negrita y cambiar el color de un título
4. Mostrar/ocultar un elemento con un botón

html

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <style>
5     .grande { font-size: 24px; }
6     .pequeno { font-size: 12px; }
7     .negrita { font-weight: bold; }
8     .oculto { display: none; }
9   </style>
10 </head>
11 <body>
12   <h1 id="miTitulo">Mi Título</h1>
13   <p id="miParrafo">Este es un párrafo de ejemplo.</p>
14   <div id="miDiv">Este div se puede ocultar</div>
15
16   <button onclick="">Cambiar color fondo</button>
17   <button onclick="">Tamaño pequeño</button>
18   <button onclick="">Tamaño mediano</button>
19   <button onclick="">Tamaño grande</button>
20   <button onclick="">Estilo título</button>
21   <button onclick="">Ocultar/Mostrar div</button>
22
23 <script>
24   function cambiarColorFondo() {
25
26   }
27
28   function tamanoPequeno() {
```



```
29  
30     }  
31  
32     function tamanoMediano() {  
33  
34     }  
35  
36     function tamanoGrande() {  
37  
38     }  
39  
40     function estiloTitulo() {  
41  
42     }  
43  
44     function toggleDiv() {  
45  
46     }  
47     </script>  
48 </body>  
49 </html>
```