

Curso AulaScript

Sección 3: La página web

El DHTML (Dynamic HTML)

DHTML es una técnica que combina HTML, CSS y JavaScript para crear páginas web interactivas y dinámicas. Con DHTML, puedes cambiar el contenido, el estilo o la estructura de una página en respuesta a las interacciones del usuario sin tener que recargar toda la página.

Algunos ejemplos de cómo DHTML es utilizado:

- Mostrar y ocultar contenido.
- Cambiar el tamaño, la posición o el color de un elemento cuando el usuario mueve el ratón sobre él.
- Validar formularios sin necesidad de enviar datos al servidor.

Ejemplo de uso de DHTML:

```
html
Copiar código
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Ejemplo de DHTML</title>
  <style>
    #caja {
      width: 200px;
      height: 200px;
      background-color: blue;
    }
  </style>
</head>
<body>
  <div id="caja"></div>
  <button onclick="cambiarColor()">Cambiar color</button>

  <script>
    function cambiarColor() {
      document.getElementById("caja").style.backgroundColor = "red";
    }
  </script>
</body>
</html>
```

En este ejemplo, se cambia el color de la caja al hacer clic en el botón, sin recargar la página.

Window: BOM (Browser Object Model)

El **BOM** es un conjunto de objetos que permite interactuar con el navegador web más allá del contenido de la página. El objeto principal es `window`, que actúa como un contenedor para los demás objetos del BOM como `navigator`, `screen`, `location`, entre otros.

- **window**: Es el objeto principal del BOM y representa la ventana del navegador. Proporciona métodos como `alert()`, `prompt()`, y `confirm()` para interactuar con el usuario.
- **navigator**: Proporciona información sobre el navegador que está utilizando el usuario, como el nombre, la versión y el idioma. También permite saber si la conexión es segura (HTTPS).
- **screen**: Contiene información sobre la pantalla del dispositivo, como la resolución y la profundidad de color.
- **location**: Permite acceder a la URL actual de la página y realizar redirecciones a otras páginas.

Ejemplo del uso del BOM:

```
// Muestra una alerta con el tamaño de la pantalla
```

```
alert("Tamaño de la pantalla: " + screen.width + "x" + screen.height);
```

```
// Redirige a una nueva página
```

```
window.location.href = "https://www.ejemplo.com";
```

Document: DOM (Document Object Model)

El **DOM** es una representación en forma de árbol de todos los elementos de una página web. Cada nodo en el árbol del DOM representa una parte del documento HTML, como un párrafo, un enlace, una imagen o un div. Con JavaScript, podemos acceder a estos nodos y manipularlos para alterar la página en tiempo real.

Algunas de las cosas que se pueden hacer con el DOM incluyen:

- Cambiar el texto o el contenido HTML de un elemento.
- Agregar o eliminar elementos de la página.
- Cambiar los estilos CSS de un elemento.

- Manipular los atributos de los elementos, como cambiar el `src` de una imagen.

Métodos comunes para interactuar con el DOM:

- `document.getElementById()`: Obtiene un elemento por su ID.
- `document.querySelector()`: Obtiene el primer elemento que coincide con un selector CSS.
- `element.textContent`: Cambia el texto de un elemento.
- `element.innerHTML`: Cambia el contenido HTML de un elemento.
- `element.style`: Permite modificar los estilos CSS de un elemento.

Ejemplo de manipulación del DOM:

```
<!DOCTYPE html>
```

```
<html lang="es">
```

```
<head>
```

```
<title>Ejemplo DOM</title>
```

```
</head>
```

```
<body>
```

```
<h1 id="titulo">Título original</h1>
```

```
<button onclick="cambiarTexto()">Cambiar Título</button>
```

```
<script>
```

```
function cambiarTexto() {
```

```
    document.getElementById("titulo").textContent = "Título modificado";
```

```
}
```

```
</script>
```

```
</body>
```

</html>

En este ejemplo, al hacer clic en el botón, el texto del título se modifica dinámicamente usando el DOM.

Los eventos

Los **eventos** son acciones o sucesos que ocurren en el navegador y que pueden ser capturados y gestionados por JavaScript. Ejemplos de eventos incluyen el clic del ratón, la carga de una página, el envío de un formulario, el desplazamiento de la página, entre otros.

Tipos comunes de eventos:

- **Eventos de ratón:** click, dblclick, mousedown, mouseup, mouseover, mouseout.
- **Eventos de teclado:** keydown, keyup, keypress.
- **Eventos de formulario:** submit, reset, focus, blur.
- **Eventos de ventana:** load, resize, scroll.

Para manejar eventos en JavaScript, se utiliza el método `addEventListener()`, que permite asignar una función que se ejecutará cuando ocurra un evento específico en un elemento.

Ejemplo de evento click:

```
<!DOCTYPE html>

<html lang="es">

<head>

  <title>Ejemplo de evento click</title>

</head>

<body>

  <button id="miBoton">Haz clic aquí</button>

  <script>
```

```
document.getElementById("miBoton").addEventListener("click", function() {  
    alert("¡Botón clicado!");  
});  
</script>  
</body>  
</html>
```

En este caso, se muestra una alerta cuando el usuario hace clic en el botón.

Objetos y eventos DHTML

En el contexto del DHTML, los eventos juegan un papel crucial en la creación de páginas web interactivas. DHTML permite cambiar la apariencia o el contenido de una página en respuesta a eventos, como un clic del ratón o el paso del cursor sobre un elemento.

Ejemplo de interacción con eventos en DHTML:

```
<!DOCTYPE html>  
  
<html lang="es">  
  
<head>  
  
<style>  
  
#caja {  
  
    width: 100px;  
  
    height: 100px;  
  
    background-color: green;  
  
    }  
  
</style>  
  
</head>  
  
<body>
```

```
<div id="caja"></div>
```

```
<script>
```

```
let caja = document.getElementById("caja");
```

```
// Cambiar color al pasar el ratón por encima
```

```
caja.addEventListener("mouseover", function() {
```

```
    caja.style.backgroundColor = "blue";
```

```
});
```

```
// Restaurar color al quitar el ratón
```

```
caja.addEventListener("mouseout", function() {
```

```
    caja.style.backgroundColor = "green";
```

```
});
```

```
</script>
```

```
</body>
```

```
</html>
```

En este ejemplo, el color de la caja cambia de verde a azul cuando el usuario pasa el ratón por encima, y vuelve a ser verde cuando el ratón sale de la caja.