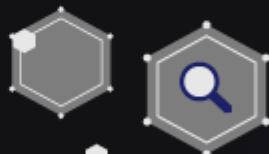


LENGUAJES DE PROGRAMACIÓN

JAVA SCRIPT





Eventos en Javascript

Los eventos juegan un papel fundamental en la programación web interactiva. En el contexto de Javascript, los eventos son acciones o sucesos que ocurren en un elemento HTML, como hacer clic en un botón, mover el mouse sobre un elemento o presionar una tecla en el teclado. Los eventos nos permiten detectar estas acciones y ejecutar código en respuesta a ellas. Es fundamental comprender los conceptos básicos de los eventos para crear sitios web dinámicos e interactivos.

En esta lección, exploraremos los eventos en Javascript. Comenzaremos por definir qué es un evento y su importancia en la programación web. Luego, nos sumergiremos en la sintaxis de los eventos, cómo detectarlos y cómo asociar funciones para manejarlos.





Eventos en Javascript

Definición de un Evento

Un evento en Javascript es una acción o suceso que ocurre en un elemento HTML y que puede ser detectado y manejado por el código Javascript. Los eventos pueden ser desencadenados por la interacción del usuario con la página, como hacer clic en un botón, mover el mouse, ingresar texto en un campo de entrada, entre otros.

Sintaxis de una Función

Los eventos en Javascript se asocian a elementos HTML utilizando el atributo on, seguido del nombre del evento y el código a ejecutar entre comillas. Por ejemplo:

```
elementoHTML.onnombreEvento = function() {  
    // Código a ejecutar en respuesta al evento  
};
```

En este ejemplo, elementoHTML es el elemento HTML al que se le va a asociar el evento, y nombreEvento es el nombre del evento que se desea detectar, como click, mouseover, keydown, entre otros.

LENGUAJES DE PROGRAMACIÓN

JAVA SCRIPT

Ejemplo de Uso de Eventos en Javascript

Veamos un ejemplo de uso de eventos en Javascript para cambiar el color de fondo de un elemento HTML al hacer clic en un botón:

```
<!DOCTYPE html>
<html>
<head>
    <title>Ejemplo de Eventos en
Javascript</title>
    <style>
        #caja {
            width: 200px;
            height: 200px;
            background-color: blue;
        }
    </style>
</head>
<body>
    <div id="caja"></div>
    <button id="boton">Cambiar color</button>
```

LENGUAJES DE PROGRAMACIÓN

JAVA SCRIPT

```
<script>
var caja = document.getElementById("caja");
var boton = document.getElementById("boton");

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

En este ejemplo, al hacer clic en el botón "Cambiar color", el evento onclick se dispara y se ejecuta la función que cambia el color de fondo de la caja a rojo.





Demostraciones y Ejemplos Adicionales

Veamos algunas demostraciones y ejemplos adicionales relacionados con los eventos en Javascript:

Ejemplo de evento mouseover:

```
var elemento = document.getElementById("miElemento");

elemento.onmouseover = function() {
    elemento.style.backgroundColor = "yellow";
};

elemento.onmouseout = function() {
    elemento.style.backgroundColor = "white";
};
```

En este ejemplo, al mover el mouse sobre el elemento con el id "miElemento", se cambia su color de fondo a amarillo (mouseover), y al salir del elemento, vuelve a su color de fondo original (mouseout).



Ejemplo de evento keydown:

```
document.onkeydown = function(evento) {  
    if (evento.key === "Enter") {  
        console.log("Se presionó la tecla  
Enter");  
    }  
};
```

En este ejemplo, al presionar la tecla "Enter" en cualquier parte de la página, se muestra un mensaje en la consola.

En esta lección, hemos explorado los eventos en Javascript. Hemos aprendido que los eventos son acciones o sucesos que ocurren en los elementos HTML y que podemos detectar y manejar mediante el código Javascript. Hemos explorado la sintaxis de los eventos y cómo asociar funciones para manejarlos.

Es esencial comprender estos conceptos básicos de los eventos en Javascript para crear sitios web dinámicos e interactivos. A medida que avancemos en nuestro aprendizaje, podremos utilizar diferentes eventos y asociar funciones para responder a las acciones del usuario de manera eficiente y efectiva.

Recuerda practicar con ejemplos y experimentar con diferentes eventos para afianzar tus conocimientos en el manejo de eventos en Javascript.