



Introducción a la programación web JavaScript

Aprende los fundamentos de la programación web con JavaScript

Empezar

Descripción general

Este curso te proporcionará una introducción completa a la programación web utilizando JavaScript. Aprenderás los conceptos básicos de JavaScript, cómo interactuar con el DOM y cómo manipular datos y eventos para crear aplicaciones web dinámicas.

01 Introducción



Introducción a JavaScript

01 | Introducción a JavaScript

¿Qué es JavaScript?

JavaScript es un lenguaje de programación ampliamente utilizado en el desarrollo web. Es un lenguaje de programación interpretado, lo que significa que el código JavaScript se ejecuta directamente en el navegador web sin necesidad de compilación previa.

JavaScript permite a los desarrolladores crear páginas web interactivas y dinámicas, agregar funcionalidades como formularios, animaciones y efectos visuales, y brindar una experiencia de usuario más rica.

¿Por qué aprender JavaScript?

Aprender JavaScript es fundamental para cualquier persona interesada en el desarrollo web. Al dominar este lenguaje de programación, podrás:

- **Mejorar la interacción con los usuarios:** JavaScript te permite agregar interactividad a tus sitios web, como validación de formularios, eventos de clic y acciones en respuesta a la interacción del usuario.
- **Crear aplicaciones web completas:** Con JavaScript, puedes desarrollar aplicaciones web complejas y dinámicas, lo que te permite construir desde simples juegos en línea hasta aplicaciones empresariales.
- **Ser más competitivo en el mercado laboral:** El conocimiento de JavaScript es muy valorado por los empleadores, ya que es uno de los lenguajes más populares y demandados en la industria del desarrollo web.

Características de JavaScript

JavaScript tiene varias características clave que lo diferencian de otros lenguajes de programación:

- **Lenguaje orientado a objetos:** JavaScript admite programación orientada a objetos, lo que significa que los objetos son piezas fundamentales en la construcción de aplicaciones.
- **Interpretado:** JavaScript se ejecuta directamente en un navegador web sin necesidad de compilación previa, lo que facilita el desarrollo y la depuración.
- **Lenguaje de alto nivel:** JavaScript es un lenguaje muy legible y fácil de aprender en comparación con otros lenguajes de programación de bajo nivel.

- **Dinámico:** JavaScript es un lenguaje de programación dinámico, lo que significa que puedes agregar, modificar y eliminar propiedades y métodos de los objetos en tiempo de ejecución.

Sintaxis básica de JavaScript

JavaScript tiene una sintaxis clara y fácil de entender. Aquí hay algunos conceptos básicos:

- **Variables:** Se utilizan para almacenar valores. Puedes declarar variables utilizando la palabra clave `var`, `let` o `const`, seguida del nombre de la variable y opcionalmente un valor inicial.
- **Funciones:** Permiten agrupar instrucciones en bloques reutilizables de código. Se definen utilizando la palabra clave `function`, seguida del nombre de la función, los parámetros dentro de paréntesis y las instrucciones dentro de llaves `{}`.
- **Tipos de datos:** JavaScript tiene varios tipos de datos, incluyendo números, strings, booleanos, objetos y arreglos. Cada tipo de dato tiene sus propias propiedades y métodos.

Herramientas para trabajar con JavaScript

Para programar en JavaScript, necesitarás un editor de texto y un navegador web. Aquí hay algunas herramientas populares para trabajar con JavaScript:

- **Visual Studio Code:** Es un editor de texto muy popular que ofrece soporte para JavaScript y una amplia gama de extensiones que facilitan el desarrollo.
- **Google Chrome Developer Tools:** Es una herramienta integrada en el navegador Google Chrome que te permite depurar y analizar tu código JavaScript en tiempo real.
- **Node.js:** Es un entorno de ejecución de JavaScript basado en el motor V8 de Google Chrome. Te permite ejecutar código JavaScript fuera del navegador y construir aplicaciones de servidor.

Conclusiones

En resumen, JavaScript es un lenguaje de programación esencial para el desarrollo web. Aprendiendo JavaScript, podrás crear sitios web interactivos y dinámicos, agregar funcionalidades avanzadas y mejorar la experiencia del usuario. Su sintaxis clara y su popularidad en la industria hacen de JavaScript una herramienta indispensable en el mercado laboral.

Conclusión - Introducción a JavaScript

En resumen, la introducción a JavaScript es fundamental para comprender los conceptos básicos de la programación web. A través del estudio de los principios fundamentales de JavaScript, los estudiantes adquirirán las habilidades necesarias para crear aplicaciones web interactivas y dinámicas.

Interacción con el DOM

El Document Object Model (DOM) es una representación estructurada de un documento HTML que nos permite interactuar con los elementos de la página web utilizando scripts. La interacción con el DOM es esencial para la programación web con JavaScript, ya que nos permite manipular y actualizar dinámicamente el contenido de una página web.

Acceso a los elementos del DOM

Para interactuar con los elementos del DOM, primero debemos acceder a ellos. Existen diversas formas de acceder a los elementos del DOM:

Acceso por etiqueta

Podemos acceder a los elementos del DOM utilizando el nombre de la etiqueta HTML correspondiente. Por ejemplo, si queremos acceder a todos los elementos `p` de la página, podemos utilizar el método `getElementsByTagName`:

```
var parrafos = document.getElementsByTagName('p');
```

Acceso por ID

Si un elemento tiene un atributo `id` único, podemos acceder a él utilizando el método `getElementById`:

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

Acceso por clase

Si varios elementos comparten la misma clase, podemos acceder a ellos utilizando el método `getElementsByClassName`:

```
var elementos = document.getElementsByClassName('miClase');
```

Acceso por selector

Podemos acceder a los elementos del DOM utilizando selectores CSS utilizando el método `querySelector` o `querySelectorAll`. Por ejemplo, si queremos acceder a todos los elementos con la clase `miClase`, podemos utilizar el siguiente código:

```
var elementos = document.querySelectorAll('.miClase');
```

Manipulación de los elementos del DOM

Una vez que hemos accedido a los elementos del DOM, podemos manipularlos de varias formas:

Cambio de contenido

Podemos cambiar el contenido de un elemento utilizando la propiedad `innerHTML`. Por ejemplo, si queremos cambiar el contenido de un elemento `p` con `id` "miElemento", podemos hacer lo siguiente:

```
var elemento = document.getElementById('miElemento');
elemento.innerHTML = 'Nuevo contenido';
```

Cambio de estilos

Podemos cambiar los estilos de un elemento utilizando la propiedad `style`. Por ejemplo, si queremos cambiar el color de fondo de un elemento `div` con `id` "miElemento", podemos hacer lo siguiente:

```
var elemento = document.getElementById('miElemento');
elemento.style.backgroundColor = 'blue';
elemento.style.color = 'white';
```

Añadir o eliminar clases

Podemos añadir o eliminar clases a un elemento utilizando las propiedades `classList.add` y `classList.remove`, respectivamente. Por ejemplo, si queremos añadir una clase "resaltado" a un elemento `p` con `id` "miElemento", podemos hacer lo siguiente:

```
var elemento = document.getElementById('miElemento');
elemento.classList.add('resaltado');
```

Manipulación de eventos

La interacción con el usuario en una página web se basa en eventos, como hacer clic en un botón o mover el mouse sobre un elemento. Podemos utilizar JavaScript para manipular estos eventos y realizar acciones específicas en respuesta a ellos.

Escucha de eventos

Podemos utilizar el método `addEventListener` para escuchar eventos en un elemento. Por ejemplo, si queremos ejecutar una función cuando se hace clic en un elemento `button` con `id` "miBoton", podemos hacer lo siguiente:

```
var boton = document.getElementById('miBoton');  
boton.addEventListener('click', miFuncion);
```

Eventos comunes

Existen muchos tipos de eventos que podemos escuchar, incluyendo eventos de clic (`click`), eventos de teclado (`keydown`, `keyup`), eventos de ratón (`mouseover`, `mouseout`), entre otros. Podemos utilizar estos eventos para realizar diferentes acciones en respuesta a las interacciones del usuario.

Conclusion

La interacción con el DOM es fundamental en la programación web con JavaScript, ya que nos permite manipular y actualizar dinámicamente los elementos de una página web. Al acceder y manipular los elementos del DOM, podemos cambiar el contenido, los estilos y las clases de los elementos, así como también escuchar eventos y responder a las acciones del usuario.

En conclusión, la interacción con el DOM es esencial para manipular y modificar la estructura y los estilos de una página web. Mediante el uso de JavaScript, los estudiantes aprenderán a acceder a los elementos del DOM y a cambiar su contenido y apariencia de forma dinámica, lo que les permitirá crear experiencias interactivas para los usuarios.

Manipulación de datos y eventos

03 | Manipulación de datos y eventos

La manipulación de datos y eventos es una parte fundamental en la programación web con Java Script. Permite interactuar con el usuario y modificar el contenido de una página web de manera dinámica. En este tema, exploraremos cómo manipular datos y controlar eventos en Java Script.

Manipulación de datos

La manipulación de datos en Java Script se refiere a la capacidad de leer y modificar los valores de variables. Esto es especialmente útil cuando se trabaja con formularios, donde es necesario obtener los datos ingresados por el usuario para realizar diferentes operaciones.

Existen diferentes formas de manipular datos en Java Script. Algunas de las funciones más comunes incluyen:

- `parseInt()`: Convierte una cadena de texto a un valor entero.
- `parseFloat()`: Convierte una cadena de texto a un valor decimal.
- `toString()`: Convierte un valor numérico a una cadena de texto.
- `toFixed()`: Redondea un número a una cantidad específica de decimales.
- `toUpperCase()`: Convierte una cadena de texto a mayúsculas.
- `toLowerCase()`: Convierte una cadena de texto a minúsculas.

Estas funciones nos permiten realizar operaciones matemáticas, mostrar mensajes al usuario de forma adecuada y realizar transformaciones en los datos según lo necesitemos.

Control de eventos

El control de eventos en Java Script permite detectar las acciones realizadas por el usuario en la página web, como hacer clic en un botón, escribir en un campo de texto o desplazarse sobre un elemento.

Para controlar eventos, Java Script utiliza un modelo conocido como "Event-Driven Programming" (Programación basada en eventos). Este modelo se basa en la interacción entre eventos y "event handlers" (manipuladores de eventos).

Algunos de los eventos más comunes incluyen:

- `click`: Se activa cuando se realiza un clic sobre un elemento.
- `submit`: Se activa cuando se envía un formulario.
- `keydown`: Se activa cuando se presiona una tecla.
- `mouseover`: Se activa cuando el cursor se desplaza sobre un elemento.

Para controlar un evento, se debe asociar una función (event handler) al elemento HTML correspondiente. Esta función se ejecutará automáticamente cuando se produzca el evento especificado.

Ejemplo:

```
const boton = document.querySelector('#miBoton');

boton.addEventListener('click', function() {
  console.log('Se hizo clic en el botón');
});
```

En este ejemplo, cuando se hace clic en el botón con el id "miBoton", se muestra un mensaje en la consola.

Conclusiones

La manipulación de datos y eventos es esencial en la programación web con JavaScript. Permite interactuar con el usuario y modificar el contenido de una página web de manera dinámica. Con el conocimiento de las diferentes funciones de manipulación de datos y el control de eventos, se pueden crear aplicaciones web interactivas y funcionales.

En el próximo tema, exploraremos cómo trabajar con elementos del DOM (Document Object Model) y cómo modificar su contenido utilizando JavaScript.

Conclusión - Manipulación de datos y eventos

En resumen, la manipulación de datos y eventos es un aspecto clave de la programación web con JavaScript. A través del uso de manipulación de datos, los estudiantes aprenderán a almacenar, modificar y mostrar información de manera dinámica en una página web. Además, el manejo de eventos les permitirá crear interacciones y respuestas en tiempo real, mejorando la usabilidad y la experiencia del usuario.

Pongamos en práctica tus conocimientos

04 | Ejercicios Practicos

En esta lección, pondremos la teoría en práctica a través de actividades prácticas. Haga clic en los elementos a continuación para verificar cada ejercicio y desarrollar habilidades prácticas que lo ayudarán a tener éxito en el tema.

Ejercicio 1: Calculadora básica



Crea una calculadora básica en JavaScript que permita realizar operaciones de suma, resta, multiplicación y división. La calculadora debe tener dos campos de entrada para ingresar los números y mostrar el resultado de la operación en pantalla.

Ejercicio 2: Cambio de estilo



Crea una página web que contenga un botón. Al hacer clic en el botón, el fondo de la página debe cambiar de color. Utiliza JavaScript para cambiar el

estilo del elemento.

Ejercicio 3: Lista de tareas



Crea una lista de tareas en JavaScript. La lista debe permitir agregar nuevas tareas, marcar tareas como completadas y eliminar tareas. Utiliza eventos para interactuar con los elementos de la lista.



- ✓ En resumen, la introducción a JavaScript es fundamental para comprender los conceptos básicos de la programación web. A través del estudio de los

principios fundamentales de JavaScript, los estudiantes adquirirán las habilidades necesarias para crear aplicaciones web interactivas y dinámicas.

- ✓ En conclusión, la interacción con el DOM es esencial para manipular y modificar la estructura y los estilos de una página web. Mediante el uso de JavaScript, los estudiantes aprenderán a acceder a los elementos del DOM y a cambiar su contenido y apariencia de forma dinámica, lo que les permitirá crear experiencias interactivas para los usuarios.
- ✓ En resumen, la manipulación de datos y eventos es un aspecto clave de la programación web con JavaScript. A través del uso de manipulación de datos, los estudiantes aprenderán a almacenar, modificar y mostrar información de manera dinámica en una página web. Además, el manejo de eventos les permitirá crear interacciones y respuestas en tiempo real, mejorando la usabilidad y la experiencia del usuario.

