TIC

TALENTO TECH

Sesión 10

Programación Nivel Básico

UNIVERSIDAD LIBRE

etraining

Sesión 10:

Introducción a Javascript

Sintaxis básica y tipos de datos, variables y operadores

Objetivos de la sesión

Al finalizar esta sesión estarás en capacidad de:

Desarrollo de Habilidades

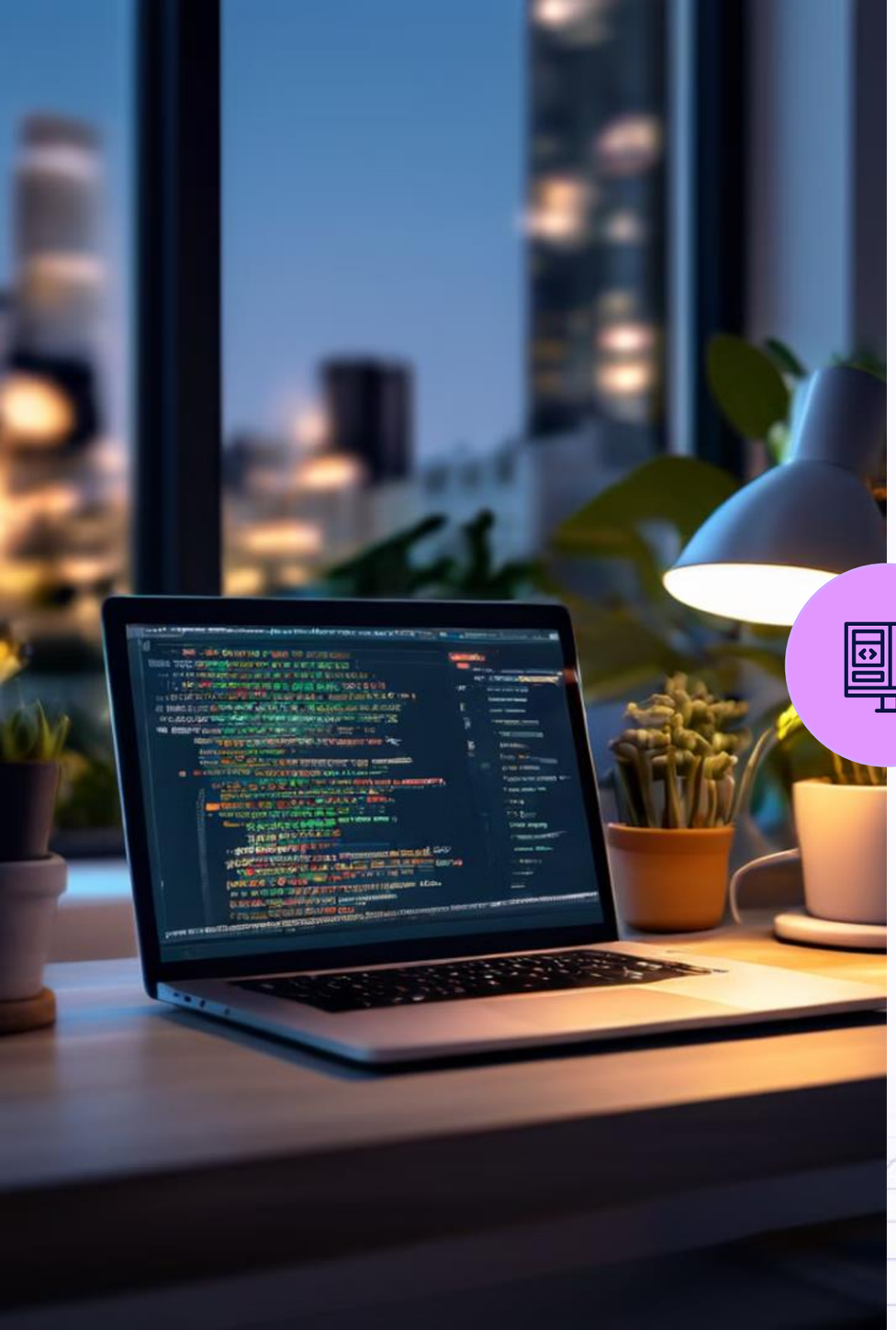
El curso busca desarrollar habilidades en JavaScript. Los estudiantes aprenderán a escribir código eficaz. Esto les permitirá crear aplicaciones web interactivas.

Comprensión de Conceptos Clave

Los participantes comprenderán conceptos fundamentales. Se cubrirán variables, operadores y estructuras de control. Estos son esenciales para la programación en JavaScript.

Aplicación Práctica

El curso incluye ejercicios prácticos. Los estudiantes resolverán problemas del mundo real. Esto ayudará a afianzar el conocimiento adquirido durante las lecciones.



Introducción a JavaScript

JavaScript es un lenguaje de programación esencial para el desarrollo web. Permite crear interactividad y dinamismo en las páginas. Con sus características versátiles, es ampliamente utilizado en aplicaciones modernas.

En esta sección, exploraremos los fundamentos de JavaScript. Abordaremos la sintaxis básica, los tipos de datos y cómo funcionan las variables y operadores. Esto te proporcionará una base sólida para avanzar en el aprendizaje de este poderoso lenguaje.

Conceptos básicos de JavaScript

Qué es JavaScript

JavaScript es un lenguaje de programación ampliamente utilizado para el desarrollo web. Permite hacer páginas interactivas y dinámicas. Con JavaScript, puedes manipular el contenido HTML y CSS de una página.

Estructura de JavaScript

JavaScript tiene una sintaxis simple y basada en objetos. Los bloques de código se agrupan en funciones y objetos. Este enfoque facilita la organización y reutilización del código.

Ejecutar JavaScript

Javascript se ejecuta en el navegador del cliente. Los scripts pueden añadirse directamente en HTML o referenciarse desde archivos externos. Esto permite una integración fluida en las páginas web.

Definición de variables

1

¿Qué es una variable?

Una variable es un contenedor para almacenar datos.

2

Propósito de las variables

Permiten manipular y acceder a información en el código.

3

Uso de variables en JavaScript

Se utilizan para representar valores dinámicos.

Las variables son fundamentales en JavaScript. Proporcionan flexibilidad al almacenar datos temporales. Una comprensión clara de las variables es esencial para el desarrollo eficaz de software.

Tipos de datos en JavaScript

Números

En JavaScript, los números son utilizados para realizar cálculos. Pueden ser enteros o decimales. JavaScript no distingue entre diferentes tipos de números, dándoles un único tipo numérico.

Cadenas de texto

Las cadenas se utilizan para representar texto en JavaScript. Se pueden definir usando comillas simples o dobles. Pueden contener letras, números y símbolos.

booleanos

Los booleanos representan un valor verdadero o falso. Son especialmente útiles en condiciones y decisiones. Permiten controlar el flujo del programa mediante estructuras de control.

Null y undefined

El valor null indica la ausencia intencional de un valor. Por otro lado, undefined significa que una variable no ha sido inicializada. Ambos son importantes para el manejo de datos en JavaScript.

Declaración de Variables

1

Uso de la palabra clave var

La declaración de variables en JavaScript puede iniciarse con la palabra clave **var**. Esta opción es ideal para declarar variables que tendrán un alcance global o de función. Sin embargo, es importante recordar que **var** permite redefinir la variable si es necesario.

2

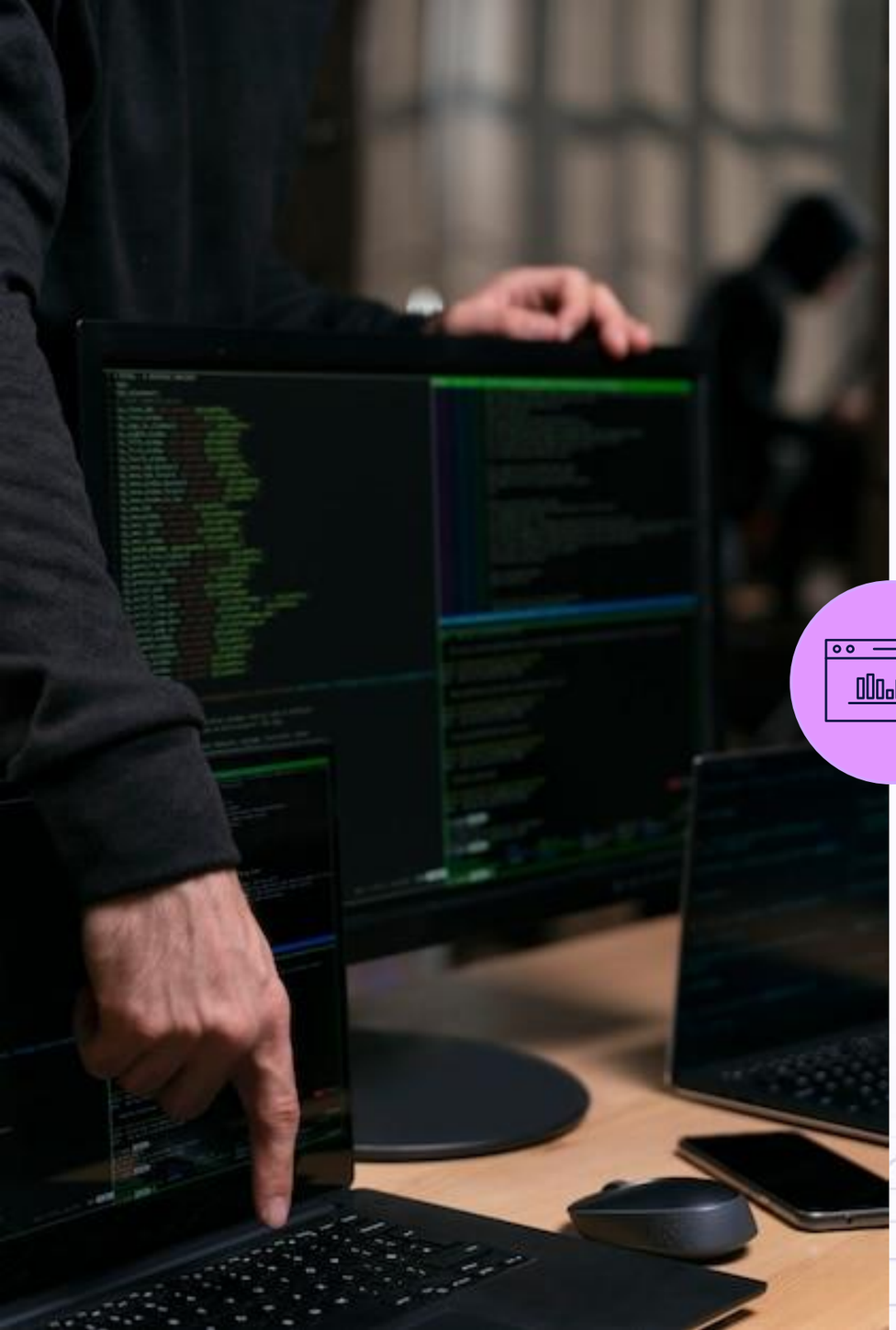
Utilizando let para un mejor control

La palabra clave **let** proporciona un control más estricto sobre el alcance de la variable. Se utiliza para definir variables de ámbito de bloque. En circunstancias donde una variable no debe ser redefinida, **let** es la mejor opción.

3

Constantes con const

Para declarar variables que no cambiarán, se utiliza **const**. Esto protege el valor y asegura que permanece constante durante la ejecución del programa. Utilizando **const**, se promueve la escritura de código más claro y seguro.



Asignación de valores a variables

1

Significado de la asignación

La asignación de valores a variables es un concepto esencial en JavaScript. Consiste en dar un valor a una variable definida previamente. Esto permite almacenar y manipular datos en un programa.

2

Uso de operadores de asignación

JavaScript ofrece varios operadores de asignación. El operador básico es el signo igual (=). También se pueden emplear operadores compuestos para asignar y modificar valores.

3

Ejemplo práctico

Por ejemplo, se puede asignar un número a una variable con: `let x = 10;`. Esta línea de código crea la variable `x` y le asigna el valor 10.

Operadores Aritméticos

Operaciones Básicas

Los operadores aritméticos son fundamentales en JavaScript. Se utilizan para realizar cálculos simples como sumar o restar.

Uso en Programación

En programación, estos operadores permiten manipular datos numéricos. Son esenciales para lograr la funcionalidad deseada.

Ejemplos Prácticos

Ejemplos prácticos ilustran cómo usar estos operadores. Ayudan a entender su aplicación en situaciones cotidianas.

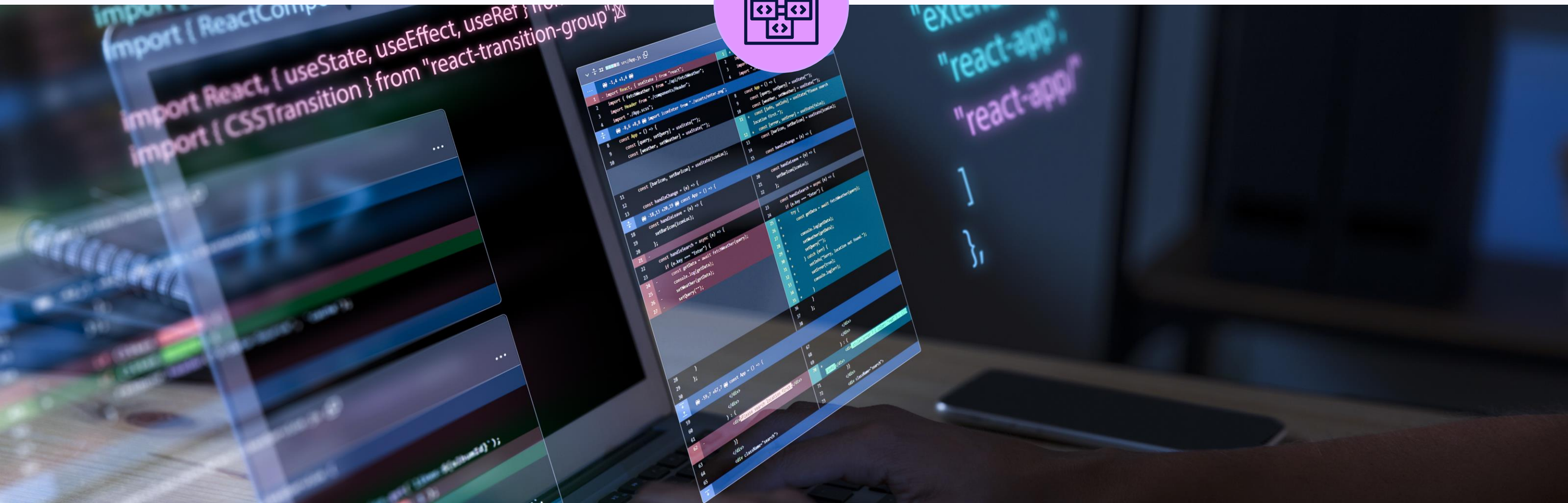
Operadores de Asignación

Asignación Simple

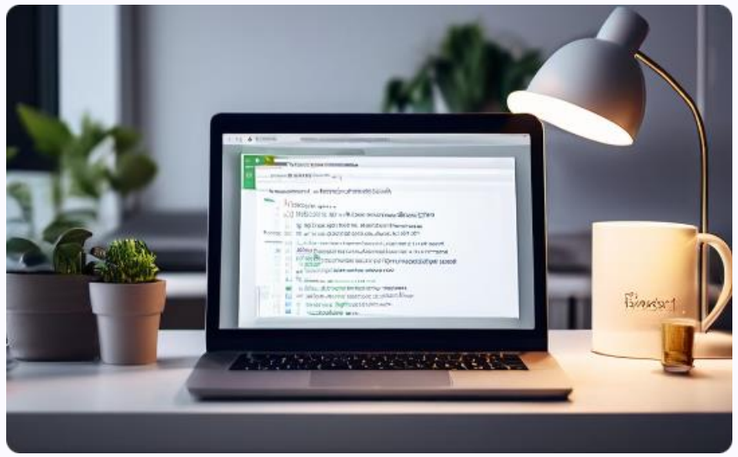
El operador de asignación simple es el igual (=). Este operador asigna el valor de la derecha a la variable de la izquierda. Por ejemplo, *let x = 5* asigna 5 a la variable x.

Operadores de Asignación Compuestos

Estos incluyen +=, -=, *=, y /=. Estos operadores permiten modificar el valor de una variable a la vez. Por ejemplo, *x += 2* suma 2 al valor actual de x.



Operadores de Comparación



Comparaciones Numéricas

Los operadores de comparación numérica permiten comparar valores. Por ejemplo, el operador '>' verifica si un número es mayor que otro.

Comparaciones de Igualdad

Los operadores de igualdad evalúan si los valores son iguales. '===' comprueba tanto el valor como el tipo de datos.

Operadores Lógicos

Los operadores lógicos combinan condiciones. Por ejemplo, el '||' verifica si al menos una condición es verdadera.

Operadores Lógicos

Definición

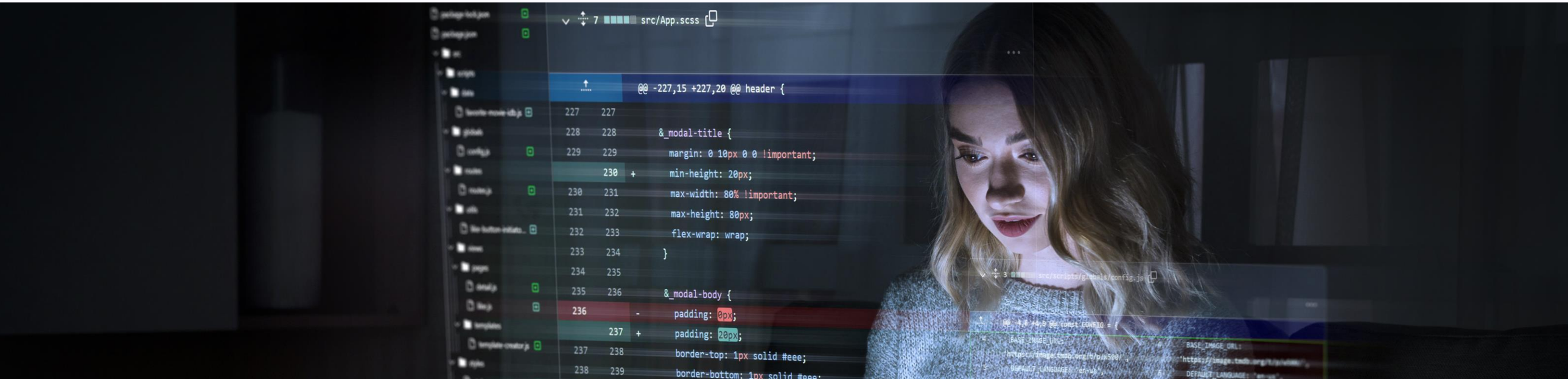
Los operadores lógicos en JavaScript permiten combinar expresiones booleanas. Las expresiones retornan valores de verdadero o falso, facilitando la toma de decisiones en el código.

Principales Operadores

Los operadores lógicos más comunes son AND (&&), OR (||) y NOT (!). Cada uno tiene funciones específicas que ayudan a manipular condiciones en el flujo del programa.

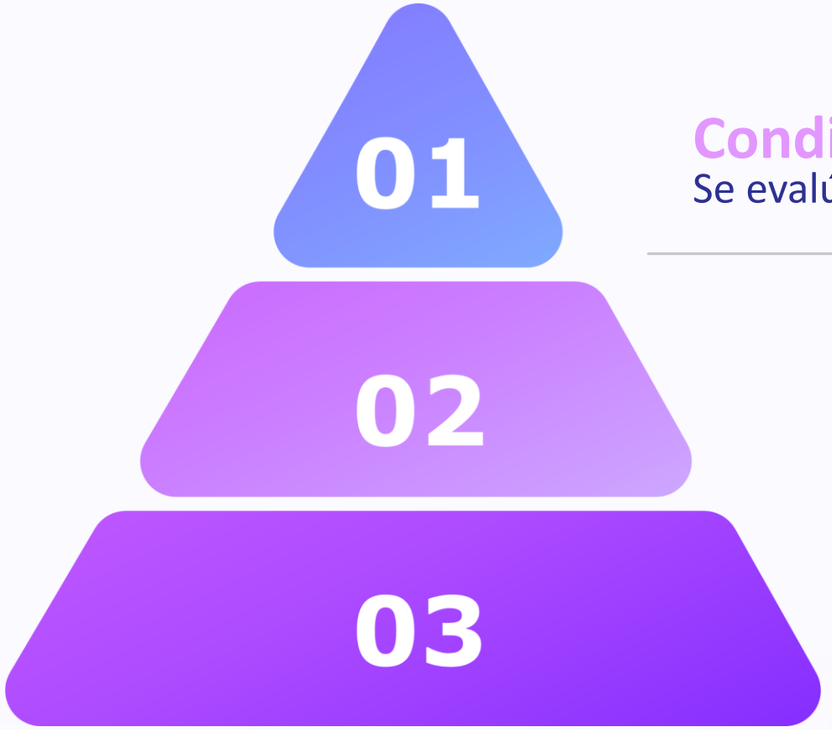
Ejemplo Práctico

Un ejemplo de uso sería: if (a && b) { ... }. Esto verifica si ambas condiciones son verdaderas antes de ejecutar el bloque de código correspondiente.



Sentencia if-else

La sentencia if-else permite tomar decisiones en el código. Es esencial para controlar el flujo de ejecución basado en condiciones. Con ella, puedes ejecutar diferentes bloques de código según si una condición es verdadera o falsa.



Condición

Se evalúa una condición específica.

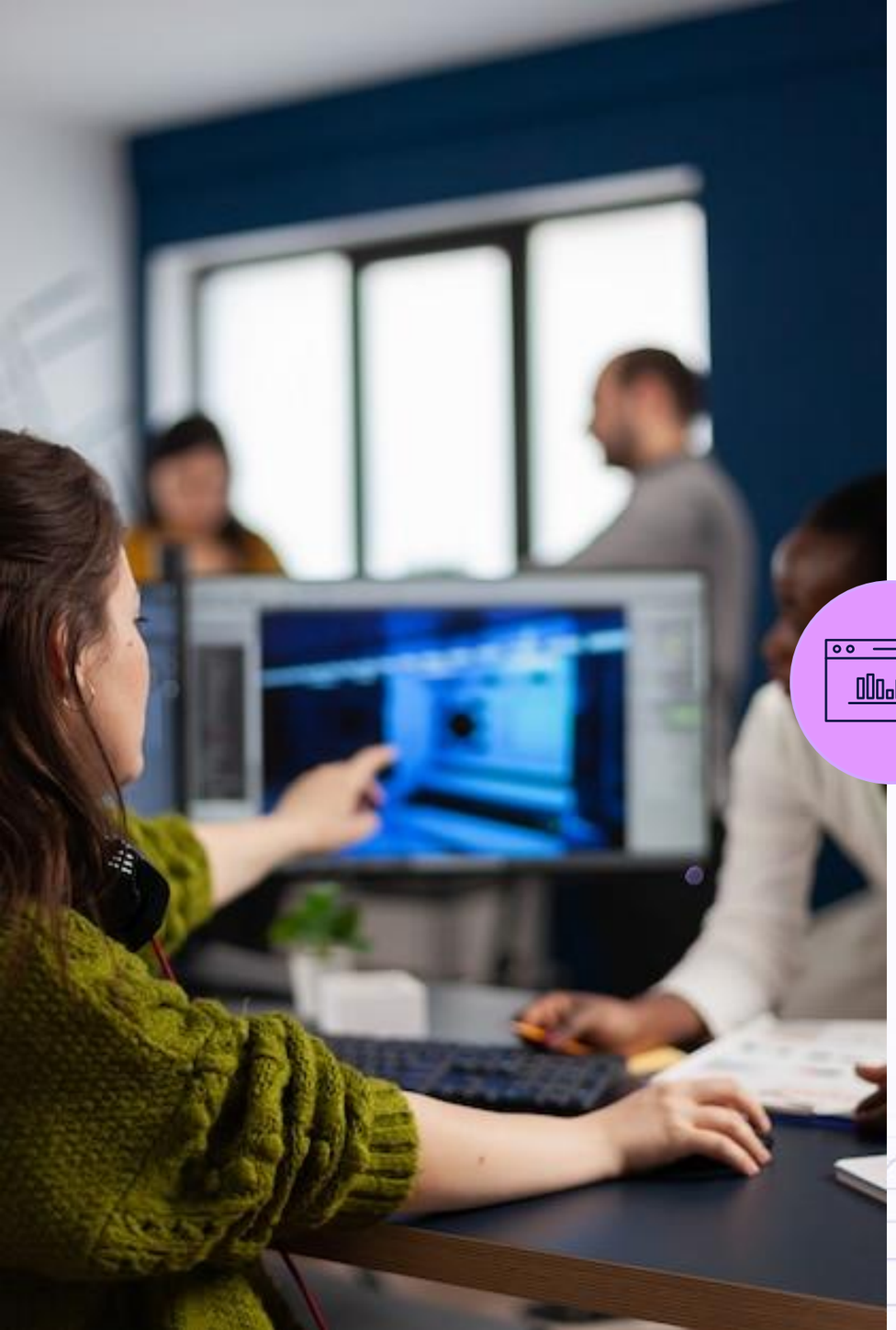
Código verdadero

Se ejecuta si la condición es verdadera.

Código falso

Se ejecuta si la condición es falsa.

El uso de if-else proporciona una estructura lógica en la programación. Es fundamental para desarrollar aplicaciones interactivas y dinámicas.



Sentencia Switch

1

Estructura de la Sentencia

La sentencia switch permite evaluar una expresión y ejecutar bloques de código en función del valor de esa expresión. Es útil cuando se necesita evaluar múltiples condiciones. Su estructura básica incluye la palabra clave switch y múltiples casos para manejar diferentes valores.

2

Casos y Default

Dentro de un switch, se definen varios casos que representan distintos valores. También se puede añadir una cláusula default para manejar situaciones sin coincidencias. Esto hace que el código sea más limpio y fácil de leer.

3

Uso Práctico

Esta sentencia es ideal para reemplazar múltiples if-else. Es común en situaciones donde se necesita determinar acciones basadas en una variable. Facilita la mantenibilidad y organización del código.

Bucle While

El bucle while es una estructura de control que permite repetir un bloque de código mientras una condición sea verdadera. Esta herramienta es fundamental en JavaScript para ejecutar acciones repetitivas e iterativas de manera eficaz. Comprender su uso es crucial para el desarrollo de scripts dinámicos.

Condición de inicio

Define el criterio para que el bucle comience.

Bloque de código

El código que se repite durante el bucle.

Condición de salida

Define cuándo el bucle debe detenerse.

Para utilizar un bucle while, asegúrate de establecer bien las condiciones iniciales y finales. Este enfoque ayuda a evitar bucles infinitos y gestionar correctamente el flujo del programa.

Bucle For

1

Introducción al Bucle For

El bucle for se utiliza para repetir un bloque de código un número específico de veces. Es útil para iterar sobre arrays o realizar cálculos repetitivos.

2

Sintaxis del Bucle For

La sintaxis básica incluye tres partes: inicialización, condición y actualización. Estas partes controlan el inicio, la duración y el final del bucle.

3

Ejecución del Bucle For

Cuando el bucle se ejecuta, evalúa la condición antes de cada iteración. Si la condición es verdadera, se ejecuta el bloque de código interno.

Ejercicio resuelto 1

1

Paso 1: Declarar una variable

Empezamos creando una variable para almacenar un valor.

2

Paso 2: Asignar un valor

Asignamos un valor a la variable previamente declarada.

3

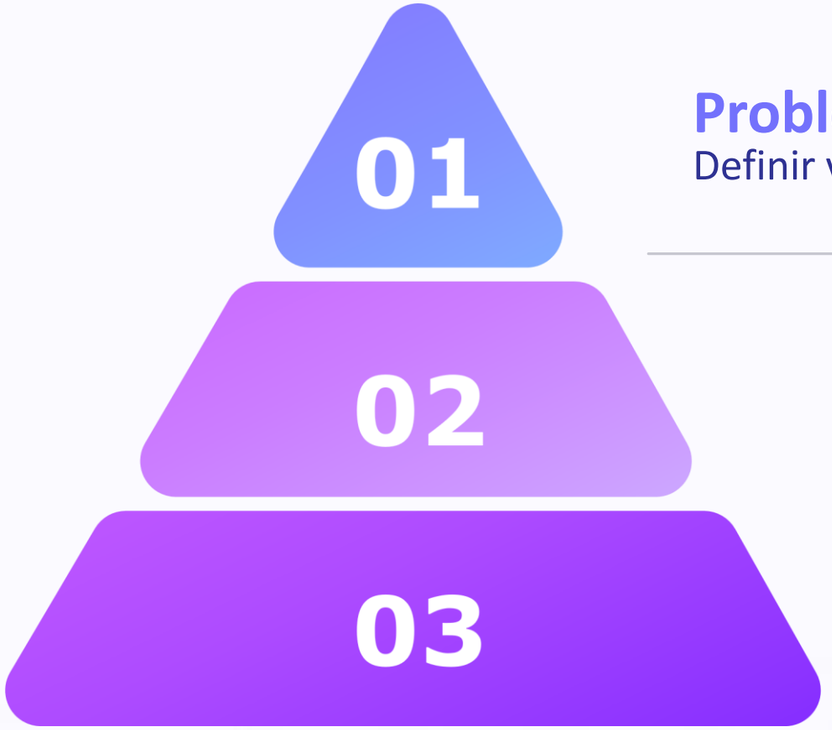
Paso 3: Mostrar el valor

Finalmente, mostramos el valor en la consola.

Este ejercicio ayuda a entender la declaración y asignación de variables en JavaScript. Cada paso es crucial para trabajar correctamente con datos en el código. Practicar estos pasos refuerza los conceptos básicos y promueve la comprensión.

Ejercicio resuelto 2

En este ejercicio resolveremos un problema aplicado utilizando variables y operadores. A continuación, se presentan los pasos necesarios para llevar a cabo la solución.



Problema planteado

Definir variables y calcular el área de un rectángulo.

Código inicial

Iniciaremos declarando las variables necesarias.

Solución final

Calcularemos y mostraremos el resultado en la consola.

Este ejercicio nos permitirá afianzar el conocimiento de la sintaxis básica en JavaScript. Practicar con ejemplos resueltos es fundamental para dominar esta tecnología.

Ejercicio Resuelto 3

1

Descripción del Problema

En este ejercicio, resolveremos un problema práctico utilizando JavaScript. El objetivo es manipular un arreglo y realizar operaciones sobre sus elementos. Aprenderemos a utilizar bucles para recorrerlo y aplicar funciones a sus valores.

2

Código de Ejemplo

A continuación, se presenta un ejemplo de código que resuelve el problema. Utilizaremos un bucle for para sumar los elementos de un arreglo. Esto nos permitirá practicar la declaración de variables y la asignación de valores.

3

Resultado y Análisis

Después de ejecutar el código, se mostrará la suma de los elementos. Este ejercicio refuerza conceptos clave de JavaScript, como la manipulación de estructuras de datos y el uso de funciones. Es un paso importante hacia la programación efectiva en JavaScript.

Ejemplo de JavaScript

```javascript
// Declaración de variables
let nombre = "Juan"; // String
let edad = 25; // Number
let esEstudiante = true; // Boolean

// Tipos de datos
let numero = 10; // Number
let texto = "Hola, mundo!"; // String
let booleano = false; // Boolean
let indefinido; // undefined
let nulo = null; // null

// Operadores aritméticos
let suma = numero + 5;
let resta = numero - 3;
let multiplicacion = numero * 2;
let division = numero / 2;
let modulo = numero % 3;

// Operadores de asignación
numero += 5; // equivalente a numero = numero + 5
numero -= 2; // equivalente a numero = numero - 2

// Operadores de comparación
let esMayor = edad > 18; // true
let esIgual = nombre === "Juan"; // true

// Operadores lógicos
let resultado = esEstudiante && edad < 30; // true
let otroResultado = esEstudiante || edad > 30; // true
```

```javascript
// Imprimir resultados
console.log("Nombre:", nombre);
console.log("Edad:", edad);
console.log("Es estudiante:", esEstudiante);
console.log("Número:", numero);
console.log("Texto:", texto);
console.log("Booleano:", booleano);
console.log("Indefinido:", indefinido);
console.log("Nulo:", nulo);
console.log("Suma:", suma);
console.log("Resta:", resta);
console.log("Multiplicación:", multiplicacion);
console.log("División:", division);
console.log("Módulo:", modulo);
console.log("Es mayor de edad:", esMayor);
console.log("Es igual a 'Juan':", esIgual);
console.log("Resultado lógico (AND):", resultado);
console.log("Resultado lógico (OR):", otroResultado);
```

Casos de uso de JavaScript

Desarrollo Web Interactivo

JavaScript es fundamental para crear experiencias interactivas. Se utiliza para validar formularios y crear elementos dinámicos. Esto mejora la usabilidad y la interacción del usuario con el sitio.

Aplicaciones de una sola página (SPA)

Las SPA dependen de JavaScript para actualizar contenido sin recargar la página. Esto permite una navegación más fluida. Frameworks como React y Angular son ejemplos populares en este contexto.

Automatización de Tareas

JavaScript permite la automatización de tareas repetitivas en el navegador. Los scripts pueden recopilar datos y manipular contenido de manera eficiente. Esto ahorra tiempo y mejora la productividad.

Desarrollo de Juegos en Línea

JavaScript es ampliamente utilizado para crear juegos en línea interactivos. Gracias a bibliotecas como Phaser, los desarrolladores pueden implementar gráficos y físicas en sus juegos. Esto ha llevado al crecimiento del desarrollo de juegos basado en navegadores.

Recursos adicionales

Libros Recomendados

Existen varios libros que pueden profundizar tu conocimiento en JavaScript. Estos textos cubren desde lo básico hasta técnicas avanzadas. Leer libros de autores reconocidos te brindará una gran ventaja.

Cursos en Línea

Numerosos cursos en línea están disponibles para aprender JavaScript. Plataformas como Udemy y Coursera ofrecen cursos accesibles. Estos cursos suelen incluir ejercicios prácticos que facilitan el aprendizaje.

Comunidades en Línea

Unirse a comunidades en línea puede enriquecer tu aprendizaje. Foros como Stack Overflow y grupos en redes sociales son útiles. Puedes hacer preguntas y compartir tus problemas con otros entusiastas.

Conclusión

Al finalizar este curso, hemos explorado los fundamentos de JavaScript. Desde variables hasta estructuras de control, cada elemento es esencial para la programación.

Al comprender estos conceptos, estarás preparado para desarrollar aplicaciones dinámicas. La práctica constante es clave para dominar esta poderosa herramienta.

Ejercicios de Práctica

¡Gracias

Por ser parte de esta Experiencia de aprendizaje!