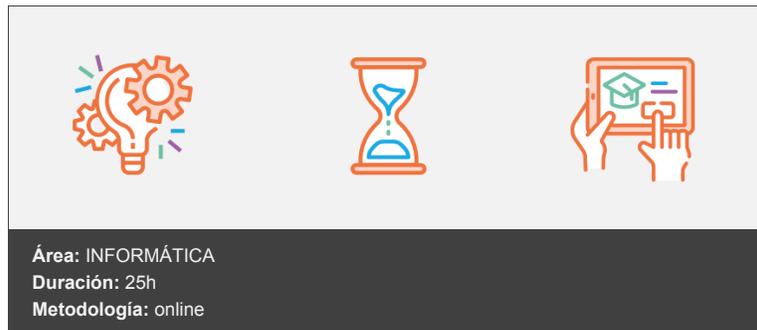


## JavaScript básico: estableciendo las bases



### Objetivos

Obtener las bases de generación de código en lenguaje JavaScript para el desarrollo e implementación de aplicaciones web, tanto en el lado del cliente como el lado del servidor.

### Contenidos y estructura del curso

Introducción

Introducción

Comprensión de los conceptos fundamentales en JavaScript

Desarrollo del código JavaScript sólido y legible

Aprendizaje de las mejores prácticas de desarrollo en JavaScript

Resumen

Conceptos básicos en JavaScript (I): Qué es JavaScript y desplegar entorno de desarrollo

Introducción

Qué es, qué no es JavaScript y qué representa en el desarrollo web actual

Despliegue del entorno de desarrollo, Aptana Studio

Pasos para desplegar el entorno de desarrollo con Aptana Studio

Características de Aptana Studio

Visual Studio Code

Instalar Visual Studio Code

Node.js

Instalar Node.js

Buenas prácticas y recomendaciones

1. Utilizar nombres descriptivos
2. Comentar el código
3. Evitar variables globales
4. Usar camelCase para nombrar variables y funciones
5. Realizar pruebas y depuración
6. Mantener el código limpio y legible
7. Actualizarse constantemente
8. Optimizar la reutilización del código
9. Seguir las recomendaciones de ECMAScript
10. Manejo adecuado de errores

11. Promover la seguridad del código

12. Refactorizar y mantener el código actualizado

Resumen

Conceptos básicos en JavaScript (II): Tipos en JavaScript

Introducción

Variables. Diferencias entre ámbito local y global. Instrucciones Let y Var

La instrucción var

La instrucción let

Diferencias entre ámbito local y global

Variables y ámbito en bucles y condicionales

Constantes

Tipos primitivos. Tipado dinámico

Tipos primitivos

Tipado dinámico

Objetos

Creación de Objetos

Prototipos y Herencia

Arrays

Creación y Manipulación de Arrays

Arrays Multidimensionales

Características Avanzadas de Arrays

Funciones

Expresiones de función

Arrow functions

Funciones como valores y callbacks

Closures en JavaScript

Parámetros predeterminados y rest

Desestructuración de Argumentos

Resumen

Operadores en JavaScript (I): Aritméticos, de asignación, de comparación y de igualdad

Introducción

Conceptos básicos sobre operadores en JavaScript

Operadores aritméticos

Suma (+)

Resta (-)

Multiplicación (\*)

División (/)

Módulo (%)

Operadores unarios

Incremento (++)

Decremento (--)

Exponenciación (\*\*)

Positivo unario (+)

Negación unaria (-)

Precedencia de operadores

Operadores de asignación

Operador de asignación de adición +=

Operador de asignación de sustracción -=

Operador de asignación de multiplicación \*=

Operador de asignación de división /=

Operador de asignación de módulo %=

Operador de asignación de exponenciación \*\*=

Operadores de comparación

Operador "Mayor que" (>)

Operador "Menor que" (<)

Operador "Mayor o igual que" (>=)

Operador "Menor o igual que" (<=)

Operadores de igualdad

Operador de Igualdad Abstracta (==)

Operador de Igualdad Estricta (===)

Operador de Desigualdad Abstracta (!=)

Operador de Desigualdad Estricta (!==)

Comparación de cadenas

Resumen

## Operadores en JavaScript (II): Ternarios, lógicos y booleanos, bitwise y jerarquización de operadores

### Introducción

#### Operadores ternarios

##### Uso básico del operador ternario

##### Ventajas sobre la estructura if-else

##### Anidación de operadores ternarios

##### Cuestiones de legibilidad

##### Buenas prácticas

#### Operadores lógicos y booleanos

##### Operador AND (&&)

##### Operador OR (| |)

##### Operador NOT (!)

##### Short-circuit evaluation

#### Operadores BitWise

##### Operador AND (&)

##### Operador OR (|)

##### Operador XOR (^)

##### Operador NOT (~)

#### Operadores de desplazamiento (Shift Operators)

##### Desplazamiento a la izquierda (Left shift)

##### Desplazamiento a la derecha con signo (Sign-propagating right shift)

##### Desplazamiento a la derecha con cero (Zero-fill right shift)

#### Jerarquización de operadores

##### Orden de Precedencia

##### Resumen

## Control de flujo de ejecución en JavaScript (I): If y switch

### Introducción

#### Sentencia if...else

##### Sentencia if

##### Sentencia else

##### Sentencia else if

#### Operadores lógicos en condiciones

##### Mejores prácticas al utilizar if, else...if y else

#### Sentencia switch...case

##### Sintaxis básica de la sentencia switch

##### Case

##### Break

##### Default

##### Grupos de casos y simplificación

##### Buenas prácticas en el uso de switch

##### Resumen

## Control de flujo de ejecución en JavaScript (II): For y while

### Introducción

#### Sentencia for

##### Usos avanzados de la sentencia for

##### Manipulación de arrays con sentencia for

##### Anidación de ciclos for

##### Optimizaciones de rendimiento en ciclos for

##### Sentencia for...in

##### Manipulación de arrays con sentencia for...in

##### Consideraciones al utilizar for...in

##### Sentencia for...of

##### Iterando sobre un array

##### Iterando sobre un string

##### Iterando sobre mapas y conjuntos

##### Uso avanzado del for...of con desestructuración

##### Iterando sobre elementos HTML

#### Sentencia while

##### Precauciones al usar bucles while

##### Manejo de múltiples condiciones

##### Sentencia do...while

##### Ejemplo básico de do...while

##### Precauciones al utilizar do...while

##### Uso en aplicaciones reales

##### Comparación con otros bucles

Variaciones en la Condición  
Anidamiento de bucles  
Refactorización de bucles infinitos  
Mejores prácticas  
Resumen

Control de flujo de ejecución en JavaScript (III): Break y continue

Introducción

Extra - Cómo evitar bucles infinitos

Validación de condiciones antes de iterar

Establecimiento de un contador de seguridad

Revisión y depuración

Uso de declaraciones condicionales dentro de bucles

Timer o timeout como mecanismo de escape

Pruebas unitarias y tests de integración

Sentencias break and continue

Sentencia break

Definición y utilidad de la sentencia break

Uso de break en bucles

Uso de break en if

Uso de break en switch

Consideraciones al usar break

Sentencia continue

Concepto de la sentencia continue

Uso de la sentencia continue en un bucle for

Uso de la sentencia continue en un bucle while

Consideraciones al usar continue

Prácticas recomendadas y casos de uso comunes

Resumen

## Metodología

En Critería creemos que para que la formación e-Learning sea realmente exitosa, tiene que estar basada en contenidos 100% multimedia (imágenes, sonidos, vídeos, etc.) diseñados con criterio pedagógico y soportados en una plataforma que ofrezca recursos de comunicación como chats, foros y conferencias...Esto se logra gracias al trabajo coordinado de nuestro equipo e-Learning integrado por profesionales en pedagogía, diseño multimedia y docentes con mucha experiencia en las diferentes áreas temáticas de nuestro catálogo.

## Perfil persona formadora

Esta acción formativa será impartida por un/a experto/a en el área homologado/a por Critería, en cumplimiento con los procedimientos de calidad, con experiencia y formación pedagógica.

***En Critería queremos estar bien cerca de ti, ayúdanos a hacerlo posible:  
¡Suscríbete a nuestro blog y síguenos en redes sociales!***

Blog de Critería

