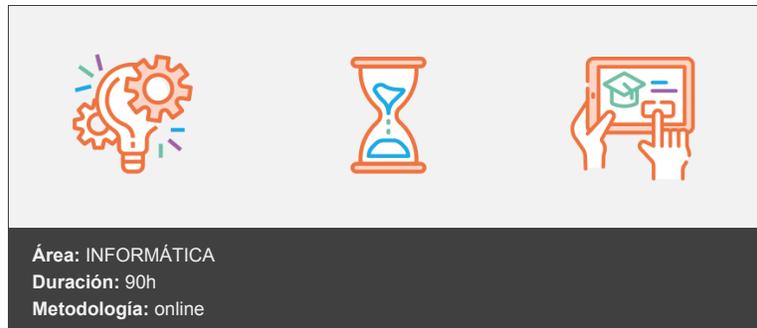


criteria

Introducción a la programación Mobile con JavaScript y PHP (IFCD56)



Objetivos

El objetivo del curso es proporcionar a los estudiantes los conocimientos y habilidades necesarios para comenzar a desarrollar aplicaciones web y móviles, utilizando JavaScript y PHP como principales lenguajes de programación.

Contenidos y estructura del curso

Tipos de Datos en JavaScript y PHP

Introducción a JavaScript

Características principales

Casos de uso

Entorno de desarrollo en JavaScript

Editor de Código

Node.js y npm (Node Package Manager)

Git y GitHub

Sistema de construcción y empaquetado

Bundler y transpilador de JavaScript

Pruebas unitarias y pruebas de integración

Frameworks y bibliotecas

Herramientas de depuración

Control de calidad del código

Entorno de desarrollo local

Instalación del entorno de NodeJS

Tipos de datos en JavaScript

Datos Booleanos

Datos numéricos

Datos de texto

Array

Objetos

Null

Introducción a PHP

Características principales

Casos de uso

Entorno de desarrollo en PHP

Consejos para elegir un entorno de desarrollo

Editor de código

Servidor web local

PHP

Gestor de dependencias

Base de datos

Control de versiones

Frameworks PHP

Instalación de Xampp en Windows

Tipos de datos en PHP

Datos Booleanos

Datos numéricos

Datos de texto

Array

Objetos

Null

Resumen

Variables

Tipos de variables en JavaScript

Declaración de variables en JavaScript

Uso de var, let y const (JavaScript)

Creación de una variable y asignación de valores

Comportamiento dinámico

Conversión de tipos

Ámbitos de variables en JavaScript

Ámbitos locales

Ámbitos globales

Ámbitos superglobales

Variables estáticas

Tipos de variables en PHP

Declaración de variables en PHP

Uso del símbolo \$ y define (PHP)

Creación de una variable y asignación de valores

Ejemplo 1: Creación de una variable y asignación de un valor entero

Ejemplo 2: Creación de una variable y asignación de una cadena de texto

Ejemplo 3: Creación de una variable y asignación de un valor booleano

Conversión de tipos en PHP

Conversión Implícita

Conversión Explícita

Ámbitos de variables en PHP

Ámbitos locales

Ámbitos globales

Ámbitos superglobales

Variables estáticas

Resumen

Constantes: Características y funcionalidades de los diferentes tipos de datos

Características de las constantes en JavaScript

Funcionalidades de las constantes en JavaScript

Ventajas de usar constantes en JavaScript

Características de las constantes en PHP

Funcionalidades de las constantes en PHP

Comparación de las constantes en JavaScript y PHP

Ejemplo de comparación de las constantes en JavaScript y PHP

Resumen

Interpolación de cadenas de texto (string) con Javascript y PHP

Interpolación de cadenas de texto con JavaScript

Utilización del formato `${variable}` y las comillas invertidas (JavaScript)

Interpolación de cadenas de texto con PHP

Utilización de las comillas dobles y simples (PHP)

Comparación de la interpolación de cadenas en JavaScript y PHP

JavaScript

PHP

Diferencias clave entre JavaScript y PHP

Resumen

Concepto de operador y expresión en programación

Concepto de operador en programación

Precedencia de operadores en programación

Tabla de precedencia de operadores

Ejemplos de precedencia de operadores

Concepto de expresión en programación

Tipos de expresiones en programación

Ejemplos de expresiones en programación con JavaScript

Ejemplos de expresiones en programación con PHP

Resumen

Clasificación de los operadores comunes en JavaScript y PHP

Introducción a operadores y expresiones en JavaScript y PHP

Clasificación: aritméticos, de comparación, lógicos, de asignación, etc.

Prioridad y asociatividad de operadores

Clasificación de los operadores comunes en JavaScript

Operadores de asignación

Ejemplo con operadores de asignación en JavaScript

Ejemplo con operadores bitwise

Operadores aritméticos en JavaScript

Operadores de incremento / decremento en JavaScript

Ejemplo de Operador de Decremento

Operadores de comparación en JavaScript

Operadores ternarios en JavaScript

Operadores lógicos en JavaScript

Clasificación de los operadores comunes en PHP

Operadores de asignación en PHP

Ejemplo completo operadores de asignación en PHP

Operadores aritméticos en PHP

Ejemplo completo operadores aritméticos en PHP

Operadores de comparación en PHP

Operadores ternarios en PHP

Ejemplo completo operadores ternarios en PHP

Operadores lógicos en PHP

Ejemplo completo operadores lógicos en PHP

Resumen

Operadores y expresiones exclusivos en PHP

Operadores exclusivos de PHP (PHP 7)

Operador de fusión de null

Operador de naves espaciales (<=>)

Operador de declaración de tipo (: tipo)

Operador de asignación por referencia (=&)

Operador de expansión de splat (...)

Operador de escalera (match)

Operador de comparación de identidad (===)

Operador de comparación de no identidad (!==)

Expresiones exclusivas de PHP (PHP 7)

Resumen

Operadores y expresiones en JavaScript (ECMAScript 6+)

Operadores en JavaScript (ECMAScript 6+)

Operadores de asignación

Operadores de comparación

Operadores lógicos

Ejemplo completo con los operadores lógicos

Operadores ternarios

Operador de propagación

Expresiones en JavaScript (ECMAScript 6+)

Uso de la función eval en JavaScript

Resumen

Introducción a la relación HTML/CSS/JavaScript

Estructura del documento HTML

Cabecera y cuerpo de un documento HTML

El papel de CSS

Ejemplo HTML + CSS

Integración de JavaScript en un documento HTML

Incrustado directamente en la página HTML

Incluido como un archivo externo

Eventos HTML y JavaScript

Resumen

Los eventos

Introducción a los eventos en JavaScript

Definición de eventos: interactividad basada en acciones del usuario

Modelo de eventos en JavaScript: captura, propagación, manejo

Cambiar el color de fondo de un elemento al hacer click

Mostrar un mensaje al hacer clic en un botón y detener la propagación

Tipos de eventos en JavaScript y su función

Evento onload

Evento onclick

Eventos del ratón: click, mouseover, mouseout, mousemove

click

mouseover

mouseout

mousemove

Eventos de teclado: keydown, keyup, keypress

Eventos de formulario: submit, change, input

PHP y su único evento

PHP vs JavaScript

Resumen

Las funciones

Declaración y llamada de funciones

Declaración de función con la palabra clave function

Expresión de función

Funciones de flecha (introducidas en ECMAScript 6)

Llamada de funciones

Parámetros y argumentos en funciones

Argumentos opcionales

Objeto arguments

Devolución de los datos generados a una función

Usando la palabra clave return

Modificando variables por referencia

Usando callbacks

Ventajas de los callbacks

Desventajas de los callbacks

Usando

promesas

Resumen

Relación de las funciones y los eventos

Relación de las funciones y los eventos en JavaScript

Ejemplo de relación entre eventos y funciones

Manejo de eventos con JavaScript

Uso de addEventListener: Asociación de funciones con eventos

Manejo de eventos de un formulario

Funciones de manejo de eventos: Cómo responder a un evento

Ejemplo con document.querySelector()

Ejemplo con document.querySelectorAll()

Relación de las funciones y los eventos en PHP

Validar los datos de un formulario

Resumen

Tipos de condicionales y sus funcionalidades

Tipos de condicionales y sus funcionalidades en JavaScript

Sentencia if

Sentencia else if y sentencia else

Sentencia switch

Operadores ternarios

Tipos de condicionales y sus funcionalidades en PHP

Sentencias if y else if

Calculando la función factorial en PHP

Sentencia switch

Operador ternario

Resumen

Utilización correcta de las condiciones, operadores y expresiones

Operadores de comparación en JavaScript

Validación de un formulario de registro con JavaScript

Operadores lógicos en JavaScript

Validación de una fecha con JavaScript

Operadores de comparación en PHP

Validación de un formulario de registro con PHP

Operadores lógicos en PHP

Validación de una fecha con PHP

Resumen

Estructura y sintaxis

Utilización correcta de los condicionales en JavaScript

Operadores de comparación

Operadores lógicos

Sentencias anidadas

Otros ejemplos de sentencias anidadas en JavaScript

Consejos para usar condicionales correctamente

Utilización correcta de los condicionales en PHP

Sentencias anidadas en PHP

Otros ejemplos de sentencias anidadas en PHP

Resumen

Los arrays, características y funcionalidades

Introducción al concepto de array

Características principales

Utilización correcta de los arrays en programación

Características de los arrays en programación

Operaciones

Ejemplo operaciones con arrays

Tipos de arrays

Ejemplo con array unidimensional

Ejemplo con array multidimensional

Ventajas y desventajas

Estructura de los arrays

Funcionalidades de los arrays en programación

Resumen

Sintaxis

Sintaxis de los arrays

Buenas prácticas

Utilización correcta de los arrays en JavaScript

Utilización correcta de los arrays en PHP

Uso de corchetes y comas (para separar valores)

Uso de corchetes y comas en JavaScript

Ejemplo de uso de corchetes y comas en JavaScript

Uso de corchetes y comas en PHP

Ejemplo de uso de corchetes y comas en PHP

Recorrer 2 arrays de forma simultánea

Selección de valores: array[índice]

Selección de valores en JavaScript

Selección de valores en PHP

Resumen

Creación de arrays

Creación de arrays en JavaScript

Constructor Array()

Escritura compacta

Escritura por valores

Operadores de spread y rest

Escritura con el método push

Tipos de arrays en PHP

Arrays Indexados (Numéricos)

Arrays Asociativos

Creación de arrays en PHP

Ejemplos adicionales

Escritura compacta

Escritura por valores

Escritura con el método push

Resumen

Propiedad y métodos de los arrays

Métodos sobre arrays en JavaScript

indexOf (búsqueda de valores en un array desde el principio)

lastIndexOf (búsqueda de valores en un array desde el final)

push (añadir valores al final)

unshift (añadir valores al principio)

slice (seleccionar múltiples valores)

splice (eliminación de valores)

join (conversión a String)

concat (concatenar diferentes arrays)

Métodos sobre arrays en PHP

array_push(): Añadir un elemento al final del array

array_unshift(): Añadir un elemento al principio del array

array_shift(): Elimina el primer elemento del array

count(): Devuelve el número de elementos en el array

current(): Devuelve el elemento actual del array

next(): Avanza al siguiente elemento del array

end(): Devuelve el último elemento del array

foreach(): Itera sobre los elementos del array

key(): Devuelve el índice del elemento especificado

array_keys(): Devuelve un array con los índices del array

array_pop(): Elimina el último elemento del array

array_rand(): Devuelve un elemento aleatorio del array

reset(): Devuelve el primer elemento del array

array_reverse(): Invierte el orden de los elementos del array

array_slice(): Devuelve una porción del array

sort(): Ordena los elementos del array

array_splice(): Elimina una porción del array y reemplaza los elementos eliminados con los valores especificados

Resumen

Funcionalidades, características y utilidades de los bucles

Introducción al concepto de bucle

Utilización correcta de los bucles en programación

Características de los bucles en programación

Estructura de los bucles

Ejemplos de bucles

Funcionalidades de los bucles en programación

Uso de los bucles para búsqueda y ordenación de datos

Operaciones sobre bucles

Resumen

Tipos de bucles y sus orientaciones

Introducción a los bucles en JavaScript

Tipos de bucles en JavaScript

For Loop

Parámetros obligatorios: (inicio, condición e incremento/decremento)

Tabla de multiplicar usando for loop

Uso de break

Caso práctico sentencia break

Uso de continue

Do-while loop

While loop

Adivinar un número utilizando bucles

Iteración sobre arrays

Foreach

For...of Loop

For...in Loop

map (Exclusivo a partir de ECMAScript 6+)

Resumen

Relación de los bucles con los arrays: creación dinámica de contenidos

Introducción al DOM

Estructura jerárquica: Nodos y relaciones entre ellos

Nodos como unidades básicas

Nodos y elementos en el DOM

Ejemplo práctico acceso a nodos y elementos del DOM

Tipos de nodos: elementos, atributos, texto

Acceso a nodos: métodos getElementById, getElementsByName, getElementsByTagName, y querySelector

Uso de getElementById

Uso de getElementsByTagName

Uso de getElementsByTagName

Uso de querySelector

Relación de los bucles con los arrays: creación dinámica de contenido

InnerHTML con el operador +=

Manipulación de contenido

Modificación de texto y atributos: innerText, innerHTML, setAttribute, getAttribute

Uso de innerText

Uso de setAttribute

Creación y eliminación de nodos: createElement, appendChild, removeChild

Uso de createElement

Uso de appendChild

Uso de removeChild

Resumen

Introducción al desarrollo Mobile con Apache Cordova

Introducción a Apache Cordova

Historia y propósito de Apache Cordova

Ventajas y desventajas de utilizar Cordova para el desarrollo móvil

Arquitectura de Apache Cordova

Componentes principales: WebView, Plugins y API Bridge

Ciclo de vida de una aplicación

Cómo Cordova permite la creación de aplicaciones multiplataforma

Funcionamiento de Cordova

Instalación de Node.js y npm

Instalación de Java para proyectos en Android

Instalación de Gradle para proyectos en Android

Instalación de Cordova CLI y dependencias

Configuración de proyectos para plataformas específicas (iOS, Android, etc.)

Paso a paso

Resumen

Desarrollo de un proyecto con Apache Cordova

Creación de un proyecto básico

Crear un nuevo proyecto Cordova

Estructura de directorios y archivos importantes

Ficheros de configuración

Desarrollo de funcionalidades básicas

Implementar lógica de aplicación con JavaScript

Uso de la cámara del dispositivo

Uso de la geolocalización del dispositivo

Mostrar mapa con la geolocalización del dispositivo

Resumen

Trabajar con JavaScript y PHP en un proyecto con Apache Cordova

JavaScript para Cordova

Uso de JavaScript en el contexto de aplicaciones móviles

APIs de Cordova para acceder a características del dispositivo

Ejemplos con las APIs

Ej

emplo de uso sobre información del dispositivo en Apache Cordova

Manejo de eventos y navegación en la interfaz de usuario

Navegación en la interfaz de usuario

Ejemplo manejar eventos y navegación básica

PHP en el lado del servidor

Ejemplos de uso de PHP en el lado del servidor

Enviar y recibir datos desde un servidor PHP

Conceptos básicos de PHP y su papel en el desarrollo web

Comunicación cliente-servidor con JavaScript

Uso de AJAX para realizar peticiones al servidor desde JavaScript

Ejemplo de solicitud GET con XMLHttpRequest en JavaScript

Manejo de respuestas JSON en el lado del cliente

Ejemplo de manejo de respuesta JSON con Fetch API

Ejemplo con Fetch API método POST

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, videos, 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

