



## Programación páginas web JavaScript y PHP

Curso de desarrollo de aplicaciones web. Para ello se estudia la programación de la parte cliente con JavaScript y la programación de la parte servidor con la tecnología PHP y el servidor de bases de datos MySQL. Al finalizar el curso, el alumno será capaz de desarrollar completas aplicaciones de comercio electrónico respaldadas por bases de datos.

**40 horas lectivas**

**40 horas prácticas**

## Contenido

### —Fundamentos de JavaScript

Estudia el lenguaje JavaScript para crear guiones o scripts que se incluyen en las páginas web y que son ejecutados por el navegador que utiliza el usuario. Tareas como la validación de los datos enviados por el usuario en un formulario o la manipulación del Modelo de Objetos del Documento (DOM) pueden realizarse mediante un lenguaje como JavaScript.

#### -Programación de páginas web

Presenta la necesidad de crear páginas web dinámicas, en las que el usuario accede a información personalizada; y explica las diferencias entre la programación del lado cliente, que ejecuta el navegador, y la programación del lado servidor, ejecutada por el servidor web. Contenido: 1. Introducción. 2. Programación del lado cliente. 3. Programación del lado servidor. 4. ¿Qué herramientas utilizar?

#### -Introducción a JavaScript

Explica la forma de incluir código JavaScript en las páginas web, para lo que se utiliza la etiqueta script, así como preparar código alternativo para aquellos navegadores que no pueden ejecutarlo o que lo tienen desactivado. También se introducen conceptos fundamentales de programación, como variables, tipos de datos, operadores y cuadros de diálogo sencillos. Contenido: 1. La etiqueta script 2. Contenido alternativo 3. Variables. 4. Tipos de datos. 5. Operadores. 6. Cuadros de diálogo.

#### -Fundamentos de programación

Estudia las estructuras de control y cómo definir funciones en JavaScript. También se explica el ámbito de las variables, aclarando la diferencia entre una variable global y una variable local. Contenido: 1. Estructuras de decisión. 2. Expresiones lógicas. 3. Estructuras de repetición. 4. Definir funciones. 5. Llamar funciones. 6. Ámbito de las variables.

#### -Objetos y Arrays en JavaScript

Estudia cómo utilizar los objetos en el lenguaje JavaScript. Aunque no se trata de un lenguaje orientado a objetos, sí que tiene objetos propios que el programador puede utilizar. Presenta la jerarquía de objetos que proporciona el navegador, así como el concepto de array, muy utilizado en programación. Contenido: 1. Introducción. 2. La jerarquía de objetos. 3. Propiedades y eventos. 4. Métodos. 5. Arrays.

#### -Los objetos location e history

Estudia los objetos location e history de JavaScript. Introduce el concepto de url. Contenido: 1. ¿Qué es una URL? 2. El objeto location. 3. Actualizando la dirección. 4. El objeto history.

#### -El objeto document

Estudia el objeto document de JavaScript, que representa el contenido de la página web: su título, conjunto de imágenes, conjunto de hipervínculos, etc. Contenido: 1. Introducción. 2. La propiedad title. 3. El método write. 4. El conjunto images.

#### -El objeto form

Estudia el objeto form de JavaScript, que permite el acceso a la información manejada en los formularios que aparecen en las páginas web. Explica cómo validar la información del formulario con JavaScript antes de enviarla al servidor web. Contenido: 1. Formularios HTML. 2. El conjunto forms. 3. La propiedad elements. 4. Validar la información. 5. ¿Cuándo realizar la validación? 6. Tipos de validación.

#### -Modelo de Objetos del Documento (DOM)

Estudia el concepto de Modelo de Objetos del Documento o DOM, indicando que los navegadores representan los documentos HTML en forma de un árbol de nodos, en el que la raíz es el objeto document. Describe los métodos esenciales de la interfaz de programación del DOM, como son los métodos para seleccionar elementos (getElementById, getElementsByTagName,

getElementById) y los que permiten obtener o establecer el valor de los atributos (getAttribute y setAttribute).  
Contenido: 1. El árbol del documento. 2. Tipos de nodos y relaciones. 3. Obtener elementos. 4. Obtener y establecer atributos.

### **-Manipulación del DOM**

Describe las propiedades y métodos del DOM que permiten manipular el contenido de una página web una vez se ha cargado, sin necesidad de volver a solicitarla al servidor web. Se estudia cómo recorrer el árbol del documento, cómo modificar el valor de los nodos de texto o cómo crear nuevo contenido. Contenido: 1. Recorrer el árbol del documento. 2. Modificar el valor de los nodos. 3. Crear, eliminar y reemplazar nodos. 4. El método innerHTML. 5. Modificar el formato dinámicamente.

### **-Buenas prácticas**

Se describen buenas prácticas a la hora de escribir código JavaScript y trabajar con el DOM de un documento: separar el código JavaScript del código HTML; escribir los manejadores de evento en el mismo código JavaScript y no como atributos de las etiquetas HTML; detectar las características con las que es compatible el navegador del usuario en lugar de intentar detectar el propio modelo de navegador, etc. Para afianzar todas estas prácticas, la lección finaliza describiendo un ejemplo completo en el que se simula una galería de imágenes. Contenido: 1. Separar estructura y comportamiento. 2. Añadir los manejadores de evento. 3. Detectar características de los navegadores. 4. Ejemplo: Una galería fotográfica. 5. Ejemplo: La estructura de la página. 6. Ejemplo: El código JavaScript.

## **—PHP**

Se estudia la tecnología de programación de aplicaciones web PHP, prestando especial atención a las características especiales que presentan este tipo de aplicaciones, a su enfoque orientado a objetos, a la seguridad en la aplicación y al acceso a bases de datos.

### **-Instalación**

Se describe la instalación del paquete XAMPP, que incluye todos los componentes necesarios para crear un entorno de desarrollo completo de aplicaciones web: el servidor web Apache, PHP, el servidor de bases de datos MySQL y otras aplicaciones auxiliares. Contenido: 1. Introducción. 2. Obtener el paquete XAMPP. 3. Instalar el paquete XAMPP. 4. Apache y MySQL como servicios. 5. Resumen de la instalación.

### **-Crear un sitio web**

Empieza explicando el funcionamiento de las páginas dinámicas, en concreto cuando se han creado con la tecnología de servidor PHP. Seguidamente configura un sitio web en Apache a través de un alias y cómo hacer lo propio para poder utilizarlo en Dreamweaver. Contenido: 1. ¿Cómo funcionan las páginas PHP? 2. Crear un alias en Apache. 3. Crear un sitio en Dreamweaver. 4. La página principal.

### **-Introducción a PHP**

Introduce conceptos fundamentales del lenguaje de programación PHP. Empieza indicando cómo debe delimitarse el código php en una página web junto al código HTML y después estudia conceptos como el de variable, tipo de datos o constante. Contenido: 1. Las etiquetas PHP. 2. Variables. 3. Tipos de datos. 4. Constantes.

### **-Arrays y estructuras de control**

Introduce el concepto de matriz o array como estructura de datos y las estructuras de control, que permiten modificar el flujo de ejecución de un programa. Contenido: 1. Arrays. 2. Estructuras de repetición. 3. Estructuras de decisión. 4. Combinar estructuras.

### **-Más sobre los arrays**

Estudia aspectos avanzados del manejo de arrays en PHP, como los arrays asociativos y el bucle foreach para recorrerlos; los arrays multidimensionales, etc. Contenido: 1. Arrays asociativos. 2. El bucle foreach. 3. Arrays multidimensionales.

### **-Funciones**

Describe las particularidades de PHP respecto de la utilización de funciones como una de las herramientas básicas para dividir la aplicación en partes más pequeñas y fáciles de manejar. Estudia los tipos de paso de parámetros y los parámetros por defecto. Contenido: 1. Introducción. 2. Crear funciones. 3. Llamar a una función. 4. Paso de parámetros. 5. Parámetros por defecto.

### **-Incluir archivos**

La lección empieza estudiando el ámbito de las variables, diferenciando entre lo que es una variable global y una variable local. Después se explica cómo incluir archivos con código en las páginas PHP, de forma que podamos dividir nuestro código en archivos que se puedan utilizar en más de una página. Contenido: 1. Ámbito de las variables. 2. Variables estáticas. 3. Uso de include y require. 4. Incluir una sola vez. 5. Seguridad de los archivos incluidos.

## **-Programación orientada a objetos (I)**

Introduce el concepto de objeto como nueva herramienta para el desarrollo de aplicaciones con PHP. Describe las diferencias entre la programación orientada a objetos y la programación procedural. Estudia el concepto de propiedad, método y constructor. Contenido: 1. Introducción. 2. Clases. 3. Propiedades. 4. Métodos.

## **-Programación orientada a objetos (II)**

Explica cómo crear objetos de una determinada clase, el concepto de visibilidad y las propiedades de solo lectura o escritura y cómo se aplican en el lenguaje de programación PHP. También menciona la posibilidad de crear destructores. Contenido: 1. Visibilidad. 2. Crear objetos. 3. Destructores.

## **-Herencia**

Explica el concepto de herencia en un lenguaje de programación orientado a objetos y cómo utilizarla en PHP. Se destacan los beneficios de utilizar esta característica en un proyecto de programación de cierta envergadura. Contenido: 1. Presentación. 2. Crear subclases. 3. Crear objetos de las subclases. 4. Sobrescribir métodos. 5. El acceso protected.

## **-Recoger datos del usuario (I)**

Se introducen los dos métodos fundamentales a la hora de recoger información proveniente del usuario, es decir, para intercambiar información entre el navegador y el servidor web. Estos métodos son el método Get y el método Post. Contenido: 1. Introducción. 2. La directiva eliminada register\_globals. 3. El array \$\_GET. 4. El array \$\_POST.

## **-Recoger datos del usuario (II)**

Estudia el envío de información a través de un formulario y con el método Post, tanto utilizando la misma página como destinataria de dicha información u otra distinta. También introduce la necesidad de validar todo lo que provenga del usuario. Contenido: 1. Recogerlos en una página distinta. 2. Recogerlos en la misma página. 3. Entradas requeridas.

## **-Validación de formularios**

Se explica la importancia de validar toda aquella información proveniente del usuario y se estudian técnicas y herramientas de PHP para evitar posibles ataques malintencionados. Contenido: 1. Expresiones regulares. 2. Limpiando la información. 3. Comprobar el formulario de origen.

## **-Trabajar con cookies**

Explica cómo utilizar cookies para mantener el seguimiento de lo que realiza el visitante de un sitio web. Contenido: 1. Introducción. 2. Crear cookies. 3. Caducidad de las cookies. 4. Dependencia del navegador. 5. Características de las cookies.

## **-Sesiones en PHP**

Estudia el concepto de sesión, utilizado para mantener información personalizada en el servidor sobre cada uno de los usuarios que acceden al sitio web. Contenido: 1. ¿Qué es una sesión? 2. El array \$\_SESSION. 3. Finalizar la sesión. 4. El identificador de las sesiones. 5. ¿Dónde se almacena la información de sesión?

## **-Acceso a archivos**

Estudia las funciones de PHP que permiten utilizar archivos para almacenar información. Se indica cómo escribir en los archivos y cómo leer su contenido y qué precauciones hay que tomar al respecto cuando la información proviene del usuario o de una fuente poco fiable. Contenido: 1. Introducción. 2. Crear el archivo. 3. Escribir en el archivo. 4. Leer de un archivo.

## **-Tratamiento de errores**

Estudia los distintos tipos de errores nativos del lenguaje PHP y cómo manejarlos en nuestra aplicación. Proporciona consejos en cuanto a cómo y cuándo mostrar esos errores. Contenido: 1. Introducción. 2. Nivel de registro de errores. 3. Mensajes de error personalizados. 4. Lanzar errores. 5. Guardar los errores en un registro.

## **-Excepciones**

Presenta la forma de tratar los errores a través de excepciones. Estudia el concepto de excepción y cómo debe utilizarse en PHP, ya que todavía se encuentra en un estado de desarrollo incompleto. Contenido: 1. Introducción. 2. Manejar excepciones. 3. Jerarquía de excepciones. 4. ¿Dónde manejar las excepciones? 5. Manejador por defecto. 5. Tratar los errores como excepciones.

## **-Bases de datos**

Se presenta la base de datos MySQL, que será la que utilizaremos para respaldar nuestra aplicación de comercio electrónico creada con PHP. Se indican algunos ajustes a realizar en la configuración inicial de MySQL; se elige la extensión mysqli para utilizarla en el código PHP y la herramienta visual PHPMyAdmin para administrar el servidor de bases de datos. Contenido: 1. MySQL. 2. Contraseña para el root. 3. Extensión mysqli. 4. PHPMyAdmin. 5. Administración de usuarios.

## **-La base de datos de ejemplo**

Describe la base de datos de ejemplo que se utilizará para respaldar la aplicación de la tienda virtual de comercio electrónico que se está construyendo. Se crean las tablas, relaciones y restricciones de integridad referencial utilizando tablas InnoDB. Contenido: 1. Tipos de tablas en MySQL. 2. Crear tablas. 3. Relaciones 1 a muchos. 4. Relaciones muchos a muchos.

## **-El lenguaje SQL**

Repasa la sintaxis del lenguaje de interrogación de bases de datos relacionales SQL. Contenido: 1. Introducción. 2. La sentencia SELECT. 3. Selecciones complejas. 4. Funciones agregadas. 5. La sentencia INSERT. 6. La sentencia UPDATE. 7. La sentencia DELETE.

## **-Acceder a la base de datos**

Presenta la secuencia de acciones a la hora de acceder a la información almacenada en una base de datos MySQL desde una aplicación PHP. Aplica el enfoque orientado a objetos disponible en la extensión mysqli. Contenido: 1. Introducción. 2. Establecer la conexión. 3. Mostrar los datos en una tabla. 4. Cerrar la conexión. 5. Mejorar el aspecto.

## **-Buscando más funcionalidad**

Explica cómo conseguir ordenar y dividir en distintas páginas el resultado obtenido a partir de la base de datos. También estudia el concepto de consulta preparada, que persigue conseguir un mayor rendimiento en ciertas aplicaciones. Contenido: 1. Ordenar el resultado. 2. Dividir el resultado en páginas. 3. Consultas preparadas.

## **-Autenticación de los usuarios**

Se explica cómo preparar una página de login donde los usuarios introduzcan sus credenciales para poder acceder a páginas privadas o protegidas. Se utilizan variables de sesión para diferenciar entre un usuario autenticado y un usuario que navega anónimamente por el sitio web. Contenido: 1. La página de login. 2. La página de registro. 3. Asegurar la confidencialidad.

## **-Sitios web seguros**

Se estudian los pasos necesarios para configurar un sitio web seguro mediante el par de protocolos SSL / HTTPS: desde la obtención de un certificado digital de servidor hasta su aplicación en el sitio web. Contenido: 1. Petición de certificado. 2. Obtener el certificado. 3. Configurar Apache con el certificado. 4. Utilizar SSL.

## **-El proceso de compra (I)**

Se explica cómo implementar el proceso de compra en la tienda virtual, centrando el estudio en la forma de identificar el carrito de la compra del cliente. Contenido: 1. Introducción. 2. Modificar listaproductos.php. 3. La página comprar.php. 4. Identificar al cliente.

## **-El proceso de compra (II)**

Se completa todo lo necesario para implementar el proceso de compra en la tienda virtual, viendo cómo se crean las órdenes en firme correspondientes a los carritos de la compra de los clientes. Contenido: 1. La página carritocompra.php. 2. Confirmar el pedido. 3. La página de desconexión. 4. Migrar el carrito de la compra. 5. Probar la compra.