

Glosario de PHP

Palabras Clave y Conceptos

PHP: Lenguaje de programación que se ejecuta en el servidor y se emplea para desarrollar sitios web dinámicos.

HTML: Lenguaje utilizado para definir la estructura del contenido en páginas web.

CSS: Herramienta que permite aplicar estilos visuales y diseños a los elementos HTML.

JavaScript: Lenguaje de programación que se ejecuta en el navegador para añadir interactividad a las páginas.

Servidor: Equipo que procesa y entrega los contenidos solicitados por los navegadores de los usuarios.

Cliente: Navegador o dispositivo desde donde los usuarios acceden a sitios web.

Base de Datos: Sistema organizado para guardar y gestionar información digital.

Formulario: Elemento web que permite a los usuarios enviar información al servidor.

Método POST: Envío de datos al servidor sin que se muestren en la dirección URL.

Método GET: Envío de datos visibles en la URL, comúnmente usado en formularios de búsqueda.

Autoloading: Técnica para cargar clases de forma automática en PHP, sin incluir archivos manualmente.

Namespaces: Mecanismo para organizar el código en espacios únicos y evitar colisiones de nombres.

Traits: Funcionalidad que permite compartir métodos entre clases sin herencia.

Interfaces: Plantillas que definen qué métodos deben implementar las clases.

Clases Abstractas: Clases base que no se pueden instanciar directamente, pero sí extender.

PDO: Capa de abstracción para trabajar con diferentes bases de datos de forma segura.

ORM: Técnica para manipular bases de datos usando objetos en lugar de consultas SQL.

REST: Estilo de arquitectura para crear APIs utilizando las reglas del protocolo HTTP.

SOAP: Protocolo que permite el intercambio estructurado de datos entre sistemas.

Middleware: Capa intermedia que procesa solicitudes antes de que lleguen a la aplicación.

Composer: Herramienta que gestiona las dependencias y librerías en proyectos PHP.

Framework: Conjunto de herramientas y buenas prácticas que simplifican el desarrollo web.

MVC: Modelo de organización que divide la aplicación en modelo, vista y controlador.

XSS: Ataque que inserta scripts maliciosos en una página para ejecutarse en el navegador.

CSRF: Técnica que induce a un usuario a ejecutar acciones no autorizadas en una web.

SQL Injection: Tipo de ataque que manipula consultas SQL para acceder a datos no autorizados.

Session Hijacking: Robo de sesiones activas para suplantar a un usuario legítimo.

Caching: Almacenamiento temporal de datos para mejorar la velocidad de respuesta.

Load Balancing: Técnica que distribuye la carga entre múltiples servidores para optimizar el rendimiento.

API: Conjunto de funciones que permite la comunicación entre diferentes programas.

Funciones y Métodos Comunes en PHP

echo: Imprime texto en la salida del navegador.

print: Similar a echo, pero devuelve un valor.

require: Inserta un archivo externo obligatorio; si falla, detiene el programa.

require_once: Igual que require, pero asegura que el archivo solo se incluya una vez.

include: Inserta un archivo externo opcional; si falla, el programa continúa.

include_once: Inserta un archivo una sola vez para evitar duplicaciones.

isset(): Verifica si una variable está definida y no es nula.

`empty()`: Comprueba si una variable está vacía.

`trim()`: Elimina espacios al inicio y final de una cadena.

`strlen()`: Cuenta los caracteres de una cadena.

`strpos()`: Encuentra la posición de un texto dentro de otro.

`str_replace()`: Sustituye texto dentro de una cadena.

`intval()`: Convierte valores a enteros.

`floatval()`: Convierte valores a decimales.

`date()`: Devuelve la fecha y hora actual formateada.

`header()`: Envía encabezados HTTP al navegador.

`session_start()`: Inicia o continúa una sesión de usuario.

`session_destroy()`: Borra todos los datos de la sesión actual.

`mysqli_connect()`: Establece conexión con una base de datos MySQL.

`mysqli_query()`: Ejecuta una consulta sobre la base de datos.

`mysqli_fetch_assoc()`: Obtiene resultados como arreglo asociativo.

`json_encode()`: Transforma datos a formato JSON.

`json_decode()`: Convierte texto JSON en estructura de PHP.

`array_map()`: Aplica una función a cada elemento de un arreglo.

`array_filter()`: Filtra elementos de un arreglo usando una función.

`array_reduce()`: Reduce un arreglo a un solo valor mediante una función.

`array_merge()`: Une múltiples arreglos en uno.

`array_slice()`: Extrae una parte de un arreglo.

`array_splice()`: Reemplaza o elimina partes de un arreglo.

`json_last_error()`: Informa del último error con funciones JSON.

`filter_var()`: Aplica filtros a valores como validación o sanitización.

`password_hash()`: Crea un hash seguro de una contraseña.

`password_verify()`: Compara una contraseña con su hash.

Conceptos de Programación Orientada a Objetos

Clase: Plantilla para crear objetos.

Objeto: Instancia concreta de una clase.

Método: Función definida dentro de una clase.

Propiedad: Atributo o dato que pertenece a un objeto.

Constructor: Método que se ejecuta al crear una instancia.

Destructor: Método que se ejecuta al eliminar una instancia.

Herencia: Permite que una clase herede características de otra.

Polimorfismo: Capacidad de usar métodos con el mismo nombre en distintas clases.

Encapsulamiento: Ocultar los detalles internos de un objeto.

Buenas Prácticas en PHP

Validación de Datos: Asegura que la información ingresada sea correcta antes de procesarla.

Manejo de Errores: Técnica para controlar y registrar errores de forma segura.

Seguridad: Prácticas para proteger el código contra vulnerabilidades comunes.

Comentarios: Explicaciones dentro del código para facilitar su comprensión.

Inyección de Dependencias: Técnica que mejora la flexibilidad del código permitiendo inyectar objetos desde fuera de una clase.

Pruebas Unitarias: Verificaciones automatizadas para asegurar que partes del programa funcionen correctamente. PHPUnit es una herramienta útil para esto.

Documentación: Facilita la comprensión del código mediante herramientas como PHPDoc.

Control de Versiones: Registro de cambios en el código, útil para colaboración, como con Git.

Despliegue Continuo: Automatización del proceso de entrega del software, con herramientas como GitHub Actions o Jenkins.