Bachiller: Yusneidi Hidalgo

C.I: 30.210.451

Investigación

De qué manera se define una constante y en qué ocasiones es necesario realizar una declaración de una constante?

Una constante en PHP es un identificador simple (nombre) para un valor que no puede ser modificado durante la ejecución del script.

1. Maneras de Definir una Constante en PHP

Hay dos formas principales de definir una constante en PHP, dependiendo del contexto:

A. Uso de define() (Más común y flexible)

La función global define() se usa para definir constantes en tiempo de ejecución.

2. Ocasiones para Declarar una Constante

Es necesario declarar una constante cuando se tiene un valor fijo que es crucial para la lógica de la aplicación y que no debe cambiar bajo ninguna circunstancia para evitar errores o vulnerabilidades.

Casos de Uso Imprescindibles:

Configuración de la Base de Datos (Seguridad):

Para almacenar credenciales de conexión que se utilizan repetidamente en el código y deben ser inmutables.

define('DB_HOST', 'localhost');

define('DB_USER', 'root');

Valores de Configuración Central:

Variables de entorno, rutas o límites que definen el comportamiento de la aplicación.

define('RUTA_ARCHIVOS', '/var/www/uploads/');

define('TIEMPO_MAXIMO_SESION', 3600); (Tiempo en segundos).

Números Mágicos y Literales:

Evitar el uso de números o strings fijos y sin explicación directa en el código (conocidos como "números mágicos"). Usar una constante mejora la legibilidad.

En lugar de if ($estado == 1), se usa define('ESTADO_ACTIVO', 1); y luego if ($estado == ESTADO_ACTIVO).

Constantes Matemáticas o Financieras:

Tasas o valores fijos que se usan en cálculos.

define('PI', 3.14159);

const TASA_INTERES = 0.05; (Como constante de clase en un modelo financiero).

Al usar constantes, el código se vuelve más seguro (ya que el valor no se puede reasignar accidentalmente en otra parte del programa) y más fácil de mantener (ya que solo se necesita cambiar el valor en un único lugar).

qué es una variable de entorno, cómo se aplican Y de qué manera se utilizan en un entorno de desarrollo y de producción?

Una variable de entorno es un valor dinámico que afecta el comportamiento de los procesos que se ejecutan en un sistema operativo o dentro de un entorno de ejecución específico (como un servidor web o un contenedor). Estas variables son una forma estándar de pasar información de configuración al software sin modificar el código fuente.

Cómo se Aplican (Uso y Acceso)

Las variables de entorno son fundamentales para la seguridad y la portabilidad de las aplicaciones modernas, ya que separan la configuración del código.

Método de Aplicación	Descripción
Nivel Sistema Operativo	Se establecen directamente en el servidor (ej. en Linux usando export MI_VARIABLE=valor en el shell). Esto es común para configuraciones globales.
Servidor Web (Apache/Nginx)	Se configuran en el archivo de configuración del servidor (.htaccess en Apache o en el bloque server de Nginx).
Archivos .env (Práctica Estándar)	Se usa un archivo simple de texto llamado .env para almacenar pares clave-valor (ej. DB_PASSWORD=m1p4ssS3guro). El código PHP utiliza una biblioteca (como Vlucas/PHPdotenv) para cargar estas variables del archivo `.env* al entorno de PHP.

Cuáles son las variables súper globales... profundizar en:

Server

Get

Post

Files

Cookies

Session

Request

Las variables superglobales en PHP son arrays asociativos predefinidos que están siempre disponibles en cualquier alcance (función, método o archivo) sin necesidad de usar global $variable;. Almacenan información importante sobre el entorno, la solicitud del usuario, la sesión y los archivos subidos.

Variable Superglobal	Propósito Principal	Ejemplo de Uso
$_SERVER	Información sobre el servidor y el entorno de ejecución.	$_SERVER['REQUEST_METHOD']
$_GET	Variables pasadas a través de la URL (método HTTP GET).	$_GET['id_producto']
$_POST	Variables pasadas a través de formularios (método HTTP POST).	$_POST['nombre_usuario']
$_FILES	Elementos subidos al script a través del formulario (archivos).	$_FILES['imagen']['name']
$_COOKIE	Datos pasados al script vía cookies HTTP.	$_COOKIE['sesion_id']
$_SESSION	Variables registradas en la sesión actual.	$_SESSION['usuario_logueado']
$_REQUEST	Contiene el contenido de $_GET, $_POST y $_COOKIE.	$_REQUEST['busqueda']
$_ENV	Variables de entorno del script.	$_ENV['DB_HOST']
$_GLOBALS	Un array que hace referencia a todas las variables disponibles en el alcance global.	global

De qué manera se aplican esas funciones para sanitizar datos antes del envío de un formulario?

Las funciones predefinidas de PHP se aplican en el backend (servidor) inmediatamente después de que un formulario se envía, actuando como una segunda y esencial línea de defensa para sanitizar y validar los datos antes de que interactúen con la base de datos o se utilicen en la lógica de negocio.

Este proceso es crucial porque la validación del lado del cliente (JavaScript) es fácilmente eludible por un atacante.

El proceso de sanitización y validación en PHP sigue un flujo estricto:

1. **Recepción:** El servidor PHP recibe los datos en $_POST, $_GET, o $_FILES.
2. **Verificación de Existencia:** Se usa **isset()/empty()** para confirmar que el dato está presente.
3. **Sanitización:** Se usa **trim()** y **str_replace()** para limpiar el dato de suciedad o formatos indeseados.
4. **Validación:** Se usa **strlen()** o **is_numeric()** para confirmar que el dato cumple con la lógica del negocio.
5. **Procesamiento Final (Seguridad):** Si es una contraseña, se aplica **password_hash()** antes de interactuar con la base de datos.
6. **Interacción con BD:** Solo si el dato **supera todos los pasos anteriores** se utiliza en la consulta SQL.

Cómo evitar un Cross site scripting, un doss, xss. Cuáles son sus afectaciones en el sistema y base de datos?

Cross-Site Scripting (XSS)

El XSS ocurre cuando un atacante inyecta scripts maliciosos (generalmente JavaScript) en una página web vista por otros usuarios. El navegador del usuario ejecuta el script, creyendo que proviene de una fuente confiable.

Afectaciones en el Sistema y la Base de Datos

Sistema y Usuario:

Robo de Sesiones (Cookies): El script inyectado puede robar la cookie de sesión del usuario y enviarla al atacante, permitiéndole tomar el control de la cuenta (secuestro de sesión).

Defacing (Alteración Visual): Modificar el contenido visible de la página web.

Keylogging: Registrar las pulsaciones de teclado del usuario.

Base de Datos (Indirecta):

Si el XSS es persistente (almacenado), el script malicioso se guarda en la BD (ej. en el campo de un comentario o un perfil de usuario).

Cuando otro usuario legítimo ve ese dato, el script se extrae de la BD y se ejecuta en su navegador.

Denegación de Servicio (DoS / DDoS)

Un ataque de Denegación de Servicio (DoS), o su versión distribuida (DDoS), busca sobrecargar un servidor, red o aplicación con una cantidad abrumadora de tráfico o solicitudes. El objetivo es que los usuarios legítimos no puedan acceder al servicio.

Afectaciones en el Sistema y la Base de Datos

Sistema (Servidor Web): El impacto primario es la indisponibilidad. El servidor se agota en recursos (CPU, memoria, ancho de banda), causando una caída o una lentitud extrema.

Base de Datos:

Agotamiento de Conexiones: El servidor web sobrecargado intenta abrir miles de conexiones a la BD, agotando el límite de conexiones del motor de la BD.

Bloqueos de Consultas: Un ataque DoS que involucre un gran volumen de solicitudes puede generar picos de consultas lentas a la BD, causando bloqueos y degradando su rendimiento.

Qué es un array asociativo, qué es una array normal, qué es una array multidimensional, cuáles son los diferentes tipos de funciones para array que existen en php?

Los arrays en PHP son estructuras de datos extremadamente flexibles que pueden almacenar múltiples valores bajo un solo nombre de variable. PHP maneja tres tipos principales de arrays basados en cómo se asignan sus claves: numérico, asociativo y multidimensional.

Tipos de Arrays en PHP

1. Array Numérico (o Indexado)

Un array numérico utiliza números enteros (índices) como sus claves. Por defecto, PHP comienza a contar en el índice 0.

Array Asociativo

Un array asociativo utiliza cadenas de texto (strings) significativas como sus claves en lugar de números enteros.

Array Multidimensional

Un array multidimensional es un array que contiene otros arrays. Esto permite almacenar datos en estructuras tabulares o jerárquicas (filas y columnas, o árboles).