

ESTUDIO DE CLASES Y FUNCIONES PHP

Fecha última actualización 13/12/2024

Índice

Clases.....	2
DateTime.....	2
PDO.....	5
PDOStatement.....	7
PDOException.....	9
SimpleXMLElement.....	10
Funciones.....	12
header().....	12
setcookie().....	12
hash().....	12

Clases

DateTime

La clase **DateTime** es una de las clases más utilizadas en PHP para trabajar con fechas y horas. Permite manipular fechas de manera eficiente y realizar diversas operaciones como cálculos de diferencias, formateos y más.

Métodos principales:

__construct(): Permite crear un objeto de la clase DateTime. Si no se proporciona un parámetro, la fecha actual será la predeterminada.

```
$date = new DateTime();
```

```
$date = new DateTime('2024-12-13');
```

format(): Permite formatear la fecha en un formato específico.

```
echo $date->format('Y-m-d H:i:s');
```

add() y sub(): Permiten añadir o restar intervalos de tiempo.

```
$date->add(new DateInterval('P1D')); // Sumar 1 día
```

diff(): Permite calcular la diferencia entre dos fechas.

```
$date1 = new DateTime('2024-12-10');
```

```
$date2 = new DateTime('2024-12-13');
```

```
$diff = $date1->diff($date2);
```

setDate(\$year, \$month, \$day): Establece una nueva fecha.

```
$date->setDate(2025, 1, 1);
```

setTime(\$hour, \$minute, \$second): Establece la hora.

```
$date->setTime(12, 30, 0);
```

getTimestamp(): Obtiene la marca de tiempo Unix (número de segundos desde 1970).

```
$timestamp = $date->getTimestamp();
```

setTimezone(\$timezone): Cambia la zona horaria.

`$date->setTimezone(new DateTimeZone('Europe/Madrid'));`

PDO

PDO (PHP Data Objects) es una interfaz en PHP que permite acceder a bases de datos de manera segura. Ofrece una manera de interactuar con bases de datos que es independiente del sistema gestor (MySQL, PostgreSQL, etc.).

Métodos principales:

__construct(): Crea una instancia de PDO para conectarse a una base de datos.

```
$pdo = new PDO('mysql:host=localhost;dbname=test', 'root', '');
```

prepare(): Prepara una consulta SQL para su ejecución. Ideal para consultas con parámetros.

```
$stmt = $pdo->prepare('SELECT * FROM users WHERE id = :id');
```

```
$stmt->bindParam(':id', $id);
```

```
$stmt->execute();
```

query(): Ejecuta una consulta SQL directamente, sin necesidad de preparar la consulta.

```
$stmt = $pdo->query('SELECT * FROM users');
```

beginTransaction() **commit()** y **rollBack()**: Permiten manejar transacciones.

```
$pdo->beginTransaction();
```

```
$pdo→commit();
```

```
$pdo→rollBack();
```

lastInsertId(): Obtiene el ID de la última fila insertada.

```
$lastId = $pdo→lastInsertId();
```

quote(): Escapa una cadena para que pueda ser usada de manera segura en una consulta SQL.

```
$quoted = $pdo→quote($str);
```

setAttribute(): Establece atributos de PDO, como el modo de manejo de errores.

```
$pdo->setAttribute(PDO::ATTR_ERRMODE,  
PDO::ERRMODE_EXCEPTION);
```

PDOStatement

PDOStatement es la clase que representa una declaración preparada. Los objetos de esta clase son creados cuando se llama a `prepare()` sobre un objeto PDO.

Métodos principales:

execute(): Ejecuta una declaración preparada. Puede recibir parámetros para ser vinculados en tiempo de ejecución.

```
$stmt→execute();
```

fetch(): Recupera una fila de resultados.

```
$row = $stmt→fetch(PDO::FETCH_ASSOC);
```

fetchAll(): Recupera todas las filas de resultados.

```
$rows = $stmt→fetchAll(PDO::FETCH_ASSOC);
```

bindValue(): Vincula un valor a un parámetro en una declaración preparada.

```
$stmt->bindValue(':id', $id, PDO::PARAM_INT);
```

bindParam(): Vincula una variable a un parámetro en una declaración preparada (con referencia).

```
$stmt->bindParam(':email', $email);
```

columnCount(): Devuelve el número de columnas en el conjunto de resultados.

```
$columns = $stmt->columnCount();
```

rowCount(): Devuelve el número de filas afectadas por la consulta.

```
$rows = $stmt->rowCount();
```


PDOException

PDOException es la clase que se lanza cuando ocurre un error durante el uso de PDO. Es una clase heredada de Exception.

Métodos principales:

getMessage(): Devuelve el mensaje de error generado por la excepción.

```
try {  
    // Código que puede lanzar excepción  
} catch (PDOException $e) {  
    echo "Error: " . $e->getMessage();  
}
```

getCode(): Devuelve el código de error de la excepción.

```
$errorCode = $e->getCode();
```

getTraceAsString(): Obtiene la traza del error como una cadena.

```
$trace = $e->getTraceAsString();
```

SimpleXMLElement

SimpleXMLElement es una clase de PHP utilizada para trabajar con XML de forma simple. Permite leer y modificar XML de manera sencilla.

Métodos principales:

addChild(): Añade un hijo al objeto SimpleXMLElement.

```
$xml = new SimpleXMLElement('<root/>');  
$xml->addChild('child', 'value');
```

asXML(): Devuelve el XML como una cadena.

```
echo $xml->asXML();
```

xpath(): Ejecuta una consulta XPath sobre el XML.

```
$results = $xml->xpath('//child');
```

addAttribute(): Añade un atributo a un elemento XML.

```
$xml->child->addAttribute('type', 'example');
```

children(): Devuelve un objeto SimpleXMLElement para los elementos hijos.

```
$children = $xml->children();
```

getNamespaces(): Devuelve los espacios de nombres asociados al objeto XML.

```
$namespaces = $xml->getNamespaces();
```

count(): Cuenta el número de hijos de un nodo.

```
$count = count($xml->children());
```

Funciones

Como muchas de las utilizadas la he utilizado anteriormente ahora explico las siguientes.

header()

La función header() envía cabeceras HTTP al cliente. Es útil para redireccionamientos, modificaciones de tipos de contenido, etc.

```
header("Location: /new-page.php"); // Redirige a otra página
```

setcookie()

La función setcookie() se utiliza para enviar una cookie al navegador del cliente. Las cookies se pueden utilizar para mantener el estado de sesión o almacenar datos.

```
setcookie('user', 'John', time() + 3600); // Caduca en 1 hora
```

hash()

La función hash() se utiliza para generar un valor hash de una cadena utilizando el algoritmo especificado (por ejemplo, sha256, md5, etc.).

```
$hashed = hash('sha256', 'password123');  
echo $hashed;
```