

Referencia Rápida de PHP

Sintaxis, arrays, OOP, base de datos, E/S de archivos

Fundamentos

Hola Mundo

```
<?php
echo "Hello, World!\n";
// PHP code must be inside <?php ... ?> tags
```

Ejecutar PHP

```
php script.php # run a file
php -r 'echo "hi\n";' # run inline code
php -S localhost:8000 # built-in dev server
```

Comentarios

```
// single-line comment
# also single-line
/* multi-line
comment */
```

Variables y Tipos

Variables

```
$name = "PHP"; // string
$version = 8.3; // float
$count = 42; // int
$active = true; // bool
$items = null; // null
```

Verificación de Tipos

gettype(\$x)	Retorna el tipo como cadena
is_string(\$x)	Verificar si es cadena
is_int(\$x)	Verificar si es entero
is_array(\$x)	Verificar si es array
is_null(\$x)	Verificar si es null
isset(\$x)	Verificar si está definido y no es null
empty(\$x)	Verificar si está vacío (falso)

Conversión de Tipos

```
$n = (int) "42"; // 42
$s = (string) 3.14; // "3.14"
$b = (bool) ""; // false
$a = (array) $obj; // object to array
```

Constantes

```
define("MAX_SIZE", 100);
const API_VERSION = "v2";
echo MAX_SIZE; // 100
```

Cadenas

Básicos de Cadenas

```
$name = "World";
echo "Hello, $name!"; // variable interpolation
echo 'Hello, $name!'; // literal (no interpolation)
echo "Value: {$arr['key']}"; // complex expression
```

Funciones de Cadenas

strlen(\$s)	Longitud en bytes
mb_strlen(\$s)	Longitud en caracteres (multibyte)
strtolower(\$s)	Convertir a minúsculas
strtoupper(\$s)	Convertir a mayúsculas
trim(\$s)	Eliminar espacios en ambos extremos
str_replace(a, b, \$s)	Reemplazar a con b en \$s
substr(\$s, 0, 5)	Subcadena desde posición 0, longitud 5
strpos(\$s, 'find')	Posición de subcadena (false si no se encuentra)
explode(' ', \$s)	Dividir cadena en array
implode(' ', \$a)	Unir array en cadena

Heredoc y Nowdoc

```
$html = <<<HTML
<p>Hello, $name</p>
HTML;
$raw = <<<'TEXT'
No $interpolation here
TEXT;
```

Arrays

Indexados y Asociativos

```
$nums = [1, 2, 3]; // indexed
$user = ["name" => "Alice", "age" => 30]; // associative
$num = 4; // append
echo $user["name"]; // access
```

Funciones de Array

count(\$a)	Número de elementos
array_push(\$a, \$v)	Agregar al final
array_pop(\$a)	Eliminar y retornar el último
array_merge(\$a, \$b)	Combinar dos arrays
in_array(\$v, \$a)	Verificar si el valor existe
array_key_exists(\$k, \$a)	Verificar si la clave existe
array_map(\$fn, \$a)	Aplicar función a cada elemento
array_filter(\$a, \$fn)	Filtrar elementos por callback
sort(\$a)	Ordenar en sitio (reindexa)
array_keys(\$a)	Retornar todas las claves

Iteración

```
foreach ($users as $user) { echo $user; }
foreach ($map as $key => $value) {
    echo "$key: $value\n";
}
```

Funciones

Función Básica

```
function add(int $a, int $b): int {
    return $a + $b;
}
echo add(3, 5);
```

Argumentos Predeterminados y con Nombre

```
function greet(string $name, string $greeting = "Hello"): string {
    return "$greeting, $name!";
}
greet("Alice");
greet(greeting: "Hi", name: "Bob"); // named args (PHP 8+)
```

Funciones Flecha

```
$double = fn(int $x): int => $x * 2;
$num = array_map(fn($n) => $n * 10, [1, 2, 3]);
```

Closures

```
$factor = 3;
$multiply = function(int $x) use ($factor): int {
    return $x * $factor;
};
echo $multiply(5); // 15
```

Clases y Objetos

Definición de Clase

```
class User {
    public function __construct(
        private string $name,
        private int $age = 0,
    ) {}
    public function greet(): string { return "Hi, {$this->name}"; }
}
```

Herencia e Interfaces

```
interface Printable {
    public function toString(): string;
}
class Admin extends User implements Printable {
    public function toString(): string { return "Admin"; }
}
```

Visibilidad

public	Accesible desde cualquier lugar
protected	Accesible desde la clase y subclases
private	Accesible solo dentro de la clase
readonly	Solo se puede asignar una vez (PHP 8.1+)
static	Pertenece a la clase, no a instancias
abstract	Debe ser implementado por la subclase

Traits

```
trait Timestamped {
    public function createdAt(): string {
        return date('Y-m-d H:i:s');
    }
}
class Post { use Timestamped; }
```

Manejo de Errores

Try / Catch / Finally

```
try {
    $result = riskyOperation();
} catch (InvalidArgumentException $e) {
    echo "Bad input: " . $e->getMessage();
} catch (Exception $e) {
    echo "Error: " . $e->getMessage();
} finally { cleanup(); }
```

Excepciones Personalizadas

```
class ApiException extends RuntimeException {
    public function __construct(string $message, private int
    $statusCode = 500) {
        parent::__construct($message, $statusCode);
    }
}
```

Seguridad Nula (PHP 8+)

```
$len = $user?->address?->zip; // nullsafe operator
$name = $input ?? "default"; // null coalescing
$data ??= []; // null coalescing assignment
```

Referencia Rápida de PHP

E/S de Archivos

Leer y Escribir Archivos

```
$content = file_get_contents("data.txt");
file_put_contents("out.txt", $content);
$lines = file("data.txt", FILE_IGNORE_NEW_LINES);
```

Manejador de Archivo

```
$f = fopen("log.txt", "a");
fwrite($f, "entry\n");
fclose($f);
```

Funciones de Archivo

file_exists(\$path)	Verificar si el archivo existe
is_dir(\$path)	Verificar si la ruta es un directorio
mkdir(\$path, 0755, true)	Crear directorio recursivamente
unlink(\$path)	Eliminar un archivo
glob('*.*txt')	Buscar archivos que coincidan con el patrón
realpath(\$path)	Resolver ruta absoluta completa

Base de Datos

Conexión PDO

```
$pdo = new PDO(
    "mysql:host=localhost;dbname=app",
    "user", "password",
    [PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION]
);
```

Sentencias Preparadas

```
$stmt = $pdo->prepare("SELECT * FROM users WHERE id = :id");
$stmt->execute(["id" => 42]);
$user = $stmt->fetch(PDO::FETCH_ASSOC);
```

Insertar y Actualizar

```
$stmt = $pdo->prepare("INSERT INTO users (name, email) VALUES
(?, ?)");
$stmt->execute(["Alice", "alice@example.com"]);
$id = $pdo->lastInsertId();
```

Modos de Fetch PDO

fetch()	Obtener una sola fila
fetchAll()	Obtener todas las filas
FETCH_ASSOC	Retornar como array asociativo
FETCH_OBJ	Retornar como objeto anónimo
FETCH_CLASS	Retornar como instancia de clase especificada

Funciones Comunes

JSON

```
$json = json_encode(["name" => "Alice", "age" => 30]);
$data = json_decode($json, true); // true = assoc array
$data = json_decode($json); // object
```

Fecha y Hora

```
echo date("Y-m-d H:i:s"); // 2026-03-26 12:00:00
$ts = strtotime("+1 week");
$dt = new DateTime("2026-01-01");
echo $dt->format("D, M j"); // Thu, Jan 1
```

Matemáticas y Aleatorio

abs(\$n)	Valor absoluto
round(\$n, 2)	Redondear a 2 decimales
ceil(\$n) / floor(\$n)	Redondear hacia arriba / abajo
min(\$a, \$b) / max(\$a, \$b)	Mínimo / máximo
random_int(1, 100)	Entero aleatorio criptográficamente seguro
number_format(\$n, 2)	Formatear con separador de miles

Expresiones Regulares

```
preg_match('/^[a-z]+$/i', $str, $matches);
preg_match_all('/\d+/', $str, $all);
$result = preg_replace('/\s+/', ' ', $str);
```