PHP 빠른 참조

문법, 배열, OOP, 데이터베이스, 파일 I/O 핵심

기본

Hello World

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

PHP 실행

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

주석

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

변수 및 타입

변수

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

타입 확인

gettype($x)	타입을 문자열로 반환
is_string($x)	문자열 여부 확인
is_int($x)	정수 여부 확인
is_array($x)	배열 여부 확인
is_null($x)	null 여부 확인
isset($x)	설정되고 null이 아닌지 확인
empty($x)	빈 값(falsy) 여부 확인

타입 캐스팅

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

상수

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

문자열

문자열 기본

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

문자열 함수

strlen($s)	문자열 길이 (바이트)
mb_strlen($s)	문자열 길이 (문자, 멀티바이트 안전)
strtolower($s)	소문자로 변환
strtoupper($s)	대문자로 변환
trim($s)	양쪽 끝 공백 제거
str_replace(a, b, $s)	**$s**에서 **a**를 **b**로 치환
substr($s, 0, 5)	위치 0부터 길이 5의 부분 문자열
strpos($s, 'find')	부분 문자열 위치 (없으면 false)
explode(',', $s)	문자열을 배열로 분리
implode(',', $a)	배열을 문자열로 합치기

Heredoc & Nowdoc

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

배열

인덱스 및 연관 배열

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

배열 함수

count($a)	원소 수
array_push($a, $v)	끝에 추가
array_pop($a)	마지막 원소 제거 후 반환
array_merge($a, $b)	두 배열 병합
in_array($v, $a)	값 존재 여부 확인
array_key_exists($k, $a)	키 존재 여부 확인
array_map($fn, $a)	각 원소에 함수 적용
array_filter($a, $fn)	콜백으로 원소 필터링
sort($a)	제자리 정렬 (재인덱스)
array_keys($a)	모든 키 반환

반복

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

함수

기본 함수

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

기본값 및 이름 있는 인수

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

화살표 함수

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

클로저

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

클래스 및 객체

클래스 정의

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

상속 및 인터페이스

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

접근 제한자

public	어디서나 접근 가능
protected	클래스 및 서브클래스에서 접근 가능
private	클래스 내부에서만 접근 가능
readonly	한 번만 할당 가능 (PHP 8.1+)
static	인스턴스가 아닌 클래스에 속함
abstract	서브클래스에서 구현해야 함

트레잇

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

오류 처리

Try / Catch / Finally

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

사용자 정의 예외

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

null 안전 (PHP 8+)

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

PHP 빠른 참조

파일 I/O

파일 읽기 및 쓰기

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

파일 핸들

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

파일 함수

file_exists($path)	파일 존재 여부 확인
is_dir($path)	경로가 디렉터리인지 확인
mkdir($path, 0755, true)	재귀적으로 디렉터리 생성
unlink($path)	파일 삭제
glob('*.txt')	패턴과 일치하는 파일 찾기
realpath($path)	절대 경로 해석

데이터베이스

PDO 연결

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

준비된 구문

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

삽입 및 업데이트

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

PDO 페치 모드

fetch()	단일 행 가져오기
fetchAll()	모든 행 가져오기
FETCH_ASSOC	연관 배열로 반환
FETCH_OBJ	익명 객체로 반환
FETCH_CLASS	지정 클래스 인스턴스로 반환

일반 함수

JSON

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

날짜 및 시간

```
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
```

수학 및 난수

abs($n)	절댓값
round($n, 2)	소수점 2자리 반올림
ceil($n) / floor($n)	올림 / 내림
min($a, $b) / max($a, $b)	최솟값 / 최댓값
random_int(1, 100)	암호학적으로 안전한 난수 정수
number_format($n, 2)	천 단위 구분자로 형식화

정규 표현식

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