

## Basic Structure and Syntax

PHP is a popular general-purpose scripting language especially suited for web development.

- Open Tag: `<?php ... ?>`.
- Print: `echo "Hello"; print("World");`.
- Variables: Start with `$. $x = 10;`.
- Comments: `//, #, /* ... */`.

## Data Types

- Scalar Types: `string, int, float, bool`.
- Compound Types: `array, object, callable, iterable`.
- Special Types: `resource, null`.
- Checking types: `gettype(), is_string(), is_int()`, etc.

## Operators

- Arithmetic: `+, -, *, /, %, **`.
- Comparison: `==, ===` (strict), `!=, >, ===`.
- Logical: `&&, ||, !, and, or, xor`.
- Null Coalescing: `??`.
- Spaceship operator (PHP 7+): `<=>`.

## Control Flow

- **if-elseif-else:**

```
if ($x > 0) { ... } elseif ($x < 0)
{ ... } else { ... }
```
- **switch:**

```
switch ($color) {
    case "red": ... break;
    default: ...
}
```
- **match** (PHP 8+):

```
$result = match ($status) {
    200 => "OK",
    404 => "Not Found",
    default => "Unknown",
};
```
- **Loops:**
  - `for ($i = 0; $i < 10; $i++) { ... }`
  - `foreach ($arr as $key => $val) { ... }`
  - `while ($cond) { ... }`

## Functions

- Definition:

```
function add(int $a, int $b = 0): int
{
    return $a + b;
}
```
- Arrow Functions (PHP 7.4+):

```
$factor = 10;
$fn = fn($x) => $x * $factor;
```

## Arrays

PHP arrays are actually ordered maps.

```
$arr = [1, 2, 3]; // Indexed
$map = ["name" => "Alice", "age" =>
25]; // Associative
$map["city"] = "Seoul"; // Add/Edit
count($arr); // Length
```

## Classes and Objects (OOP)

```
class User {
    public string $name; // Property
    public function __construct(string
$name) {
        $this->name = $name;
    }
    public function getGreet(): string {
        return "Hello, {$this->name}";
    }
}
$user = new User("Alice");
```

## Superglobals

- `$_GET, $_POST, $_REQUEST`: Form data.
- `$_SESSION, $_COOKIE`: Session and cookie data.
- `$_SERVER`: Server and execution environment.
- `$_FILES`: Uploaded files.

## Google Style Guide (PSR compatible)

### Basic Rules

- Use `<?php` only. Do not use short tags `<?>`.
- All files must end with a single blank line.
- Use `UTF-8` encoding without BOM.

## Naming Conventions

- Classes: `PascalCase`.
- Methods: `camelCase`.
- Properties: `camelCase` or `snake_case` (be consistent).
- Constants: `UPPER_SNAKE_CASE`.
- Files: Usually matches class name (`User.php`).

## Formatting (PSR-12)

- Indentation: 4 spaces.
- Line Length: Soft limit 120, hard limit 80 characters recommended.
- Braces: Opening brace for classes and methods on a new line; for control structures on the same line.
- Visibility: Always declare visibility (`public, protected, private`) for properties and methods.

## Programming Practices

- Type Hinting: Always use type hints for arguments and return values.
- String Interpolation: Use `{ $var }` inside double quotes.
- Comparison: Prefer strict comparison `===`.
- Declarations: Use `declare(strict_types=1);` at the top of the file for better type safety.