

Table of Contents



- [1. PHP Syntax and Structure](#)
 - [Hello World \(First Program\)](#)
 - [Variables and Data Types](#)
 - [Comments](#)
- [2. PHP Output](#)
- [3. PHP Data Structures](#)
 - [Arrays](#)
 - [Associative Arrays \(Key-Value Pairs\)](#)
- [4. Conditional Statements](#)
 - [Multiple Conditions](#)
- [5. Loops](#)
 - [For Loop](#)
 - [While Loop](#)
- [6. Functions](#)
 - [Default Parameters](#)
- [7. Classes and Objects](#)
- [8. File Handling](#)
 - [Read File](#)
 - [Write to File](#)
- [9. PHP Forms \(GET and POST\)](#)
 - [HTML Form Example](#)
 - [Processing Form Data \(POST Method\)](#)
- [10. Superglobals](#)
 - [Accessing GET/POST Data](#)
 - [Working with Sessions](#)
- [11. Include and Require](#)
- [12. PHP and MySQL \(Database Connection\)](#)
- [13. Error Handling](#)
- [14. Useful String Functions](#)
- [15. Array Methods](#)

1. PHP Syntax and Structure

Hello World (First Program)

```
<?php
echo "Hello, World!";
?>
```

Variables and Data Types

```
<?php
$name = "Alice";           // String
$age = 25;                 // Integer
$height = 5.8;             // Float
$isStudent = true;        // Boolean
?>
```

Comments

```
<?php
// This is a single-line comment

/*
This is a
multi-line comment
*/
?>
```

2. PHP Output

```
<?php
echo "Welcome to PHP!";    // Print text
print "Learning PHP is fun!"; // Alternative to echo
```

?>

3. PHP Data Structures

Arrays

```
<?php
$fruits = ["apple", "banana", "cherry"];
echo $fruits[0];    // Access first item

$fruits[] = "orange"; // Add item
unset($fruits[1]);    // Remove item
?>
```

Associative Arrays (Key-Value Pairs)

```
<?php
$person = [
    "name" => "Bob",
    "age" => 30
];
echo $person["name"]; // Access value by key
$person["age"] = 31;  // Update value
?>
```

4. Conditional Statements

```
<?php
$age = 18;

if ($age >= 18) {
    echo "Adult";
}
```

```
} else {  
    echo "Minor";  
}  
?>
```

Multiple Conditions

```
<?php  
$score = 85;  
  
if ($score >= 90) {  
    echo "Grade A";  
} elseif ($score >= 75) {  
    echo "Grade B";  
} else {  
    echo "Grade C";  
}  
?>
```

5. Loops

For Loop

```
<?php  
for ($i = 1; $i <= 5; $i++) {  
    echo $i;  
}  
?>
```

While Loop

```
<?php  
$count = 0;
```

```
while ($count < 5) {  
    echo $count;  
    $count++;  
}  
?>
```

6. Functions

```
<?php  
function greet($name) {  
    return "Hello, " . $name . "!";  
}  
echo greet("Alice");  
?>
```

Default Parameters

```
<?php  
function power($base, $exponent = 2) {  
    return $base ** $exponent;  
}  
echo power(3);      // 9  
echo power(3, 3);   // 27  
?>
```

7. Classes and Objects

```
<?php  
class Dog {  
    public $name;  
  
    public function __construct($name) {
```

```
        $this->name = $name;
    }

    public function bark() {
        return "Woof!";
    }
}

$dog1 = new Dog("Rex");
echo $dog1->bark();
?>
```

8. File Handling

Read File

```
<?php
$content = file_get_contents("example.txt");
echo $content;
?>
```

Write to File

```
<?php
file_put_contents("example.txt", "Hello, World!");
?>
```

9. PHP Forms (GET and POST)

HTML Form Example

```
<form method="POST" action="process.php">
    Name: <input type="text" name="name">
```

```
        <button type="submit">Submit</button>
    </form>
```

Processing Form Data (POST Method)

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = $_POST['name'];
    echo "Hello, " . htmlspecialchars($name) . "!";
}
?>
```

10. Superglobals

Accessing GET/POST Data

```
<?php
echo $_GET['user']; // Access URL parameter
echo $_POST['email']; // Access form input
?>
```

Working with Sessions

```
<?php
session_start();
$_SESSION['username'] = 'JohnDoe';
echo $_SESSION['username'];
?>
```

11. Include and Require

```
<?php
include 'header.php';
require 'config.php';
?>
```

12. PHP and MySQL (Database Connection)

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "my_database";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

13. Error Handling

```
<?php
try {
    echo 10 / 0;
} catch (Exception $e) {
    echo 'Error: ' . $e->getMessage();
}
?>
```

14. Useful String Functions

```
<?php
$text = "Hello World!";
echo strlen($text);           // String length
echo strtoupper($text);       // Uppercase
echo strtolower($text);       // Lowercase
echo str_replace("World", "PHP", $text); // Replace text
?>
```

15. Array Methods

```
<?php
$numbers = [1, 2, 3, 4, 5];
$doubled = array_map(fn($x) => $x * 2, $numbers);
print_r($doubled); // [2, 4, 6, 8, 10]

$filtered = array_filter($numbers, fn($x) => $x > 3);
print_r($filtered); // [4, 5]
?>
```

This cheat sheet provides a quick overview of PHP's essential features.