Php Cheatsheet

Basics

Hello World

The `echo` statement is used to output strings or variables:

```php
<?php
echo "Hello World!";
print "Hello Again!"; // 'print' also works
?>
```

Note: `echo` can take multiple parameters (though rarely used), while `print` always returns `1` and takes a single argument.

Comments

Single-line comments

```php
// This is a comment
# This is also a comment
```

Multi-line comments

```php
/*
   This is a
   multi-line comment
*/
```

Debugging Functions

- `var_dump($var)` → Displays structured info (type + value).
- `print_r($var)` → Prints human-readable info (good for arrays).
- `var_export($var)` → Returns parsable string representation.

```php
<?php
$names = ["Harry", "Rohan"];
var_dump($names);
print_r($names);
?>
```

Variables

- Variables start with `$`.
- Case-sensitive.
- Must start with a letter or underscore.

```php
<?php
$title = "PHP Cheat Sheet";
$_count = 10;
?>
```

Constants:

```php
define("PI", 3.14);
echo PI;
```

Data Types

- **String**
- **Integer**
- **Float (double)**
- **Boolean**
- **Array**
- **Object**
- **NULL**
- **Resource** (special variable holding reference to external resources like DB connections)

Escape Characters

Valid in **double-quoted strings** and **heredoc**:

- `\n` → newline
- `\r` → carriage return
- `\t` → tab
- `\\` → backslash
- `\"` → double quote
- `\$` → dollar sign

⚠ `\e`, `\v`, `\f` exist but are rarely useful in PHP.

Operators

Arithmetic

`+ - * / % **`

Assignment

`=, +=, -=, *=, /=, %=, .=`

Comparison

`==, ===, !=, <>, !==, >, <, >=, <=, <=>`

(`<>` is the same as `!=`) `<=>` (spaceship operator) returns `-1, 0, 1` .

Increment / Decrement

`++$x, $x++, --$x, $x--`

Logical

`&&, ||, and, or, xor, !`

String

`.` → concatenation `.=` → concatenation assignment

Array

`+` (union), `==, ===, !=, !==`

Conditional Operators

Ternary

```php
$result = ($age >= 18) ? "Adult" : "Minor";
```

Null Coalescing

```php
$username = $_GET['user'] ?? "Guest";
```

Control Structures

If / Else / Elseif

```php
if ($x > 10) {
    echo "Greater";
} elseif ($x == 10) {
    echo "Equal";
} else {
    echo "Smaller";
}
```

Switch

```php
switch ($color) {
    case "red":
        echo "Stop";
        break;
    case "green":
        echo "Go";
        break;
    default:
        echo "Wait";
}
```

Loops

```php
for ($i=0; $i<5; $i++) { echo $i; }

foreach ($arr as $value) { echo $value; }
foreach ($arr as $key => $value) { echo "$key => $value"; }

while ($x < 5) { $x++; }
```

```
do { $x++; } while ($x < 5);
```

Break / Continue are supported.

Functions

```
function greet($name = "Guest") {
    return "Hello, $name";
}
echo greet("Harry");
```

- Arguments can have default values.
- Functions can return values.
- PHP supports type declarations:

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

Superglobals

- `$GLOBALS`
- `$_SERVER`
- `$_GET`
- `$_POST`
- `$_REQUEST`
- `$_FILES` (for uploads)
- `$_COOKIE`
- `$_SESSION`

- `$_ENV`

Arrays

```php
$indexed = ["Harry", "Rohan"];
$assoc = ["name" => "Harry", "age" => 25];
$multi = [
  ["Volvo", 100],
  ["BMW", 200]
];
```

String Functions

```php
strlen("Harry");           // length
str_word_count("Hi all");  // word count
strrev("Harry");           // reverse
strpos("Hello world","world"); // position
str_replace("world","PHP","Hello world"); // replace
```

File Handling

```php
$file = fopen("test.txt","r");
$content = fread($file, filesize("test.txt"));
fclose($file);

file_put_contents("test.txt","New content");
echo file_get_contents("test.txt");
```

Error Handling

```php
try {
    throw new Exception("Error!");
} catch (Exception $e) {
    echo $e->getMessage();
}
```

OOP in PHP

```php
class Bike {
    public $color;
    public function __construct($c) { $this->color = $c; }
    public function getColor() { return $this->color; }
}

$myBike = new Bike("red");
echo $myBike->getColor();
```

Access Modifiers: `public, private, protected` **Other OOP Features:** inheritance, interfaces, traits, abstract classes, namespaces.

Useful Functions

- `isset($var)` → checks if variable is set
- `empty($var)` → checks if variable is empty
- `unset($var)` → destroys variable
- `is_array(), is_string(), is_int(), is_null()`
