

Weak comparisons

Backward Incompatible Changes

View <https://www.php.net/manual/en/migration80.incompatible.php>

Key principles

Follow these fundamental rules when writing compatible code:

Analyze variable types: Always understand what types and values a variable can hold.

Use strict comparisons: Prefer `===` and `!==` over `==` and `!=` to avoid implicit type coercion.

Declare types explicitly: Always use type declarations for function/method parameters and return types.

Validate with strict checks: Use strict type checks in conditional statements to prevent unexpected behavior (e.g., `in_array` with `true` as third parameter).

Example: <https://www.php.net/manual/en/function.in-array.php#example-2>

Use PHP constants: Always use predefined PHP constants instead of numbers (e.g., `UPLOAD_ERR_OK` instead of `0`).

Enable strict types: Add `declare(strict_types=1);` at the top of your PHP files *when possible*.

Strings

Use `empty()` instead of comparing with empty string.

❌ DON'T

✅ DO

<pre>if (\$a == '') { // ... }</pre>	<pre>if (empty(\$a)) { // ... }</pre>
<pre>if (\$a != '') { // ... }</pre>	<pre>if (!empty(\$a)) { // ... }</pre>

Checkbox values

In PHP 8, comparing numeric strings with numbers produces different results.

❌ DON'T	✅ DO
<pre>if (\$a == 0) { // ... } elseif (\$a == 1) { // ... } else { // ... }</pre>	<pre>if (strval(\$a) === '0') { // ... } elseif (strval(\$a) === '1') { // ... } else { // ... }</pre>