Comparing Objects

When using the comparison operator (`==`), object variables are compared in a simple manner, namely: Two object instances are equal if they have the same attributes and values (values are compared with `==`), and are instances of the same class.

When using the identity operator (`===`), object variables are identical if and only if they refer to the same instance of the same class.

An example will clarify these rules.

Example of object comparison

```
<?php
function bool2str($bool)
{
    if ($bool === false) {
        return 'FALSE';
    } else {
        return 'TRUE';
    }
}

function compareObjects($o1, $o2)
{
    echo 'o1 == o2 : ' . bool2str($o1 == $o2) . "\n";
    echo 'o1 != o2 : ' . bool2str($o1 != $o2) . "\n";
    echo 'o1 === o2 : ' . bool2str($o1 === $o2) . "\n";
    echo 'o1 !== o2 : ' . bool2str($o1 !== $o2) . "\n";
}

class Flag
{
    public $flag;

    function __construct($flag = true) {
        $this->flag = $flag;
```

```
    }
}

class OtherFlag
{
    public $flag;

    function __construct($flag = true) {
        $this->flag = $flag;
    }
}

$o = new Flag();
$p = new Flag();
$q = $o;
$r = new OtherFlag();

echo "Two instances of the same class\n";
compareObjects($o, $p);

echo "\nTwo references to the same instance\n";
compareObjects($o, $q);

echo "\nInstances of two different classes\n";
compareObjects($o, $r);
?>
```

ENTITY

```
Two instances of the same class
o1 == o2 : TRUE
o1 != o2 : FALSE
o1 === o2 : FALSE
o1 !== o2 : TRUE

Two references to the same instance
o1 == o2 : TRUE
o1 != o2 : FALSE
```

```
o1 === o2 : TRUE
o1 !== o2 : FALSE

Instances of two different classes
o1 == o2 : FALSE
o1 != o2 : TRUE
o1 === o2 : FALSE
o1 !== o2 : TRUE
```

> [!NOTE] Extensions can define their own rules for comparing their objects (`==`).

□□□ #2
□ □□□ □□□ 2026-10-31 17:40:31 UTC
□ □□□ □□□ 2026-10-31 17:40:32 UTC