

PHP - Boolean

In PHP, "bool" is one of the built-in scalar data types. It is used to express the truth value, and it can be either True or False. A Boolean literal uses the PHP constants True or False. These constants are case-insensitive, in the sense, true, TRUE or True are synonymous.

You can declare a variable of bool type as follows –

```
$a = true;
```

Example

Logical operators (<, >, ==, !=, etc.) return Boolean values.

```
<?php
    $gender="Male";
    var_dump ($gender=="Male");
?>
```

It will produce the following **output** –

```
bool(true)
```

Boolean Values in Control Statements

Boolean values are used in the construction of control statements such as **if**, **while**, **for** and **foreach**. The behaviour of these statements depends on the true/false value returned by the Boolean operators.

The following conditional statement uses the Bool value returned by the expression in the parenthesis in front of the **if** keyword –

```
$mark=60;

if ($mark>50)
    echo "pass";
```

```
else
    echo "fail";
```

Converting a Value to Boolean

Use the (bool) casting operator to convert a value to bool. When a value is used in a logical context it will be automatically interpreted as a value of type bool.

A non-zero number is considered as true, only 0 (+0.0 or -0.0) is false. Non-empty string represents true, empty string "" is equivalent to false. Similarly, an empty array returns false.

Example

```
<?php
    $a = 10;
    echo "$a: ";
    var_dump((bool)$a);

    $a = 0;
    echo "$a: ";
    var_dump((bool)$a);

    $a = "Hello";
    echo "$a: ";
    var_dump((bool)$a);

    $a = "";
    echo "$a: ";
    var_dump((bool)$a);

    $a = array();
    echo "$a: ";
    var_dump((bool)$a);
?>
```

It will produce the following **output** –

```
10: bool(true)
0: bool(false)
Hello: bool(true)
```

: bool(false)

Array: bool(false)