

Манипуляции с типами данных

Приведение типа — это преобразование значения одного типа в значение другого типа.

Есть два вида приведения типов:

- явное
- неявное

Неявное приведение типа выполняется интерпретатором автоматически, без непосредственного участия программиста. Например, значение будет автоматически преобразовано, если оператор ожидает числовые операнды:

```
$sum = "3" + 4; // Строка "3" будет неявно преобразована в число 3
```

Чтобы выполнить явное приведение, нужно указать в круглых скобках имя требуемого типа непосредственно перед приводимым значением или переменной:

```
$num = (int)"50";
```

Приведение типов в PHP работает так же, как и в C: имя требуемого типа записывается в круглых скобках перед приводимой переменной. Пример:

```
<?php
$foo = 10; // $foo это целое число
$bar = (boolean) $foo; // $bar это булев тип
?>
```

Допускаются следующие приведения типов:

- (int), (integer) - приведение к целому числу
- (bool), (boolean) - приведение к булеву типу
- (float), (double), (real) - приведение к числу с плавающей точкой (float)
- (string) - приведение к строке
- (array) - приведение к массиву
- (object) - приведение к объекту

Преобразование в тип Boolean (двоичные данные)

При преобразовании в логический тип, следующие значения рассматриваются как **FALSE**:

- сам булев FALSE
- целое 0 (ноль)
- число с плавающей точкой 0.0 (ноль)
- пустая строка и строка "0"
- массив с нулевыми элементами
- объект с нулевыми переменными-членами
- специальный тип NULL (включая неустановленные переменные)

Все остальные значения рассматриваются как **TRUE** (включая любой ресурс).

Внимание: -1 считается TRUE, как и любое ненулевое (отрицательное или положительное) число!

Примеры преобразований:

```
<?php
var_dump((bool) "");           // bool(false)
var_dump((bool) 1);           // bool(true)
var_dump((bool) -2);          // bool(true)
var_dump((bool) "foo");       // bool(true)
var_dump((bool) 2.3e5);        // bool(true)
var_dump((bool) array(12));    // bool(true)
var_dump((bool) array());      // bool(false)
var_dump((bool) "false");     // bool(true)
?>
```

Преобразование в тип Integer (целые числа)

При приведении bool в integer, FALSE преобразуется в 0 (нуль), а TRUE — в 1 (единицу). При приведении float в integer, дробная часть будет округлена в сторону нуля. Значение NULL преобразуется в 0:

```
echo (int)FALSE, "<br>"; // 0
echo (int)TRUE, "<br>"; // 1
echo (int)12.3, "<br>"; // 12
echo (int)NULL; // 0
```

Строки преобразуются по следующим правилам:

- Если первый символ строки является цифрой, знаком + или -, то интерпретатор переходит к анализу второго символа, если второй символ строки является цифрой, то интерпретатор переходит к анализу третьего символа и так до тех пор, пока не будет обнаружен символ отличный от цифры, после этого интерпретатор возвращает полученное целое число.
- Если строка пустая или первый символ строки не является цифрой, знаком + или -, она преобразуется в 0.

```
echo (int)"12abc", "<br>"; // 12
echo (int)"a123bc", "<br>"; // 0
echo (int)"-12", "<br>"; // -12
echo (int)"12.3", "<br>"; // 12
echo (int)""; // 0
```

Если число с плавающей точкой превышает пределы целого (как правило, это +/- 2.15e+9 = 2³¹), результат будет неопределенным, так как целое не имеет достаточной точности, чтобы вернуть верный результат. В этом случае не будет выведено ни предупреждения, ни даже замечания.

Внимание! Никогда не приводите неизвестную дробь к целому, так как это может иногда дать неожиданные результаты, например:

```
<?php
echo (int) ( (0.1+0.7) * 10 ); // выводит 7!
?>
```

Преобразование в тип Float (числа с плавающей точкой)

Строки преобразуются по следующим правилам:

- Если первый символ строки является цифрой, точкой, знаком + или -, то интерпретатор переходит к анализу второго символа, если второй символ строки является цифрой, точкой (если перед этим была цифра), буквой e или E (если перед этим была цифра или точка), то интерпретатор переходит к анализу третьего символа и так до тех пор, пока не будет обнаружен символ, который не может быть преобразован в числовое значение, после этого интерпретатор возвращает полученное число с плавающей точкой.
- Если строка пустая или первый символ строки не является цифрой, точкой, знаком + или -, она преобразуется в 0.

```
echo (float)"12abc", "<br>"; // 12
echo (float)"a123bc", "<br>"; // 0
echo (float)"-12", "<br>"; // -12
echo (float)"12.3", "<br>"; // 12.3
echo (float)".13", "<br>"; // 0.13
echo (float)"1.2e5", "<br>"; // 120000
echo (float)"1.e5"; // 100000
```

Преобразование в тип String (строки)

- Булево (boolean) значение TRUE преобразуется в строку "1", а значение FALSE представляется как "" (пустая строка).
- Целое (integer) или число с плавающей точкой (float) преобразуется в строку, представленную числом, состоящим из его цифр (включая показатель степени для чисел с плавающей точкой).
- Массивы всегда преобразуются в строку "Array".
- NULL всегда преобразуется в пустую строку.
- Ресурсы всегда преобразуются в строки со структурой "Resource id #1", где 1 - это уникальный номер ресурса (resource), присвоенный ему PHP во время выполнения.
- Объекты всегда преобразуются в строку "Object".

Для преобразования объекта в строку, объект должен иметь метод `__toString()`

```
class Obj {  
    public function __toString() {  
        return "Object";  
    }  
}  
  
$obj = new Obj;  
echo (string)$obj;
```

Если объект не имеет метода `__toString()`, то в результате преобразования будет вызвана фатальная ошибка:

```
class Obj {}  
  
$obj = new Obj;  
echo (string)$obj; // Будет вызвана ошибка
```

Преобразование в тип Array (массивы)

Для любого из типов: integer, float, string, boolean и resource, если вы преобразуете значение в массив, вы получите массив с одним элементом (с индексом 0), являющимся скалярным значением, с которого вы начали.

Если вы преобразуете в массив объект (object), вы получите в качестве элементов массива свойства (переменные-члены) этого объекта. Ключами будут имена переменных-членов.

Если вы преобразуете в массив значение NULL, вы получите пустой массив.

Преобразование в тип Object (объекты)

Если объект преобразуется в объект, он не изменяется. Если же в объект преобразуется значение любого иного типа, создается новый экземпляр встроенного класса stdClass. Если значение было пустым, новый экземпляр также будет пустым. При любом другом значении оно будет содержаться в переменной-члене scalar:

```
<?php
$obj = (object) 'ciao';
echo $obj->scalar; // выведет 'ciao'
?>
```

Преобразование в тип Resource (ресурсы)

Поскольку тип ресурс содержит специальные указатели на открытые файлы, соединения с базой данных, область изображения и тому подобное, вы не можете преобразовать какое-либо значение в ресурс.

Преобразование в NULL

Приведение к типу NULL не удаляет переменную и её значение, а лишь возвращает значение типа NULL:

```
$num = 10;
echo gettype((unset)$num); // NULL
echo $num;                // 10
```