

PHP: ВВЕДЕНИЕ В ПРОФЕССИЮ

УРОК 2. ФУНКЦИИ. ОПЕРАТОРЫ КОНТРОЛЯ. ВКЛЮЧЕНИЕ ФАЙЛОВ.

boolean (bool) – тип, имеющий всего два значения:

- **true** – истина
- **false** – ложь

Например:

```
$a = true;  
$b = false;  
  
$x = (2 == 2);  
$y = ($x != $a);
```

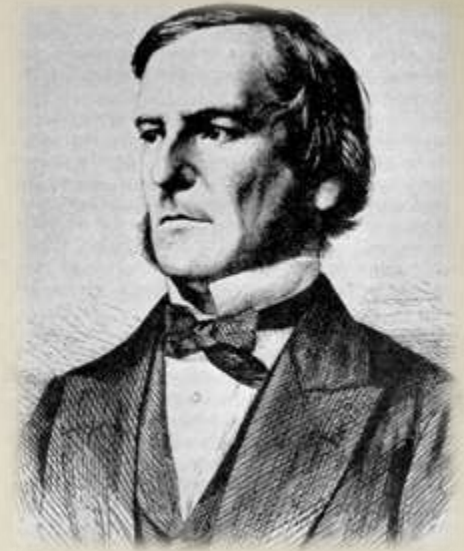
Булевы значения играют огромную роль в PHP. На них построены различные условия, многие функции принимают и возвращают булевы значения.

Приведение к boolean:

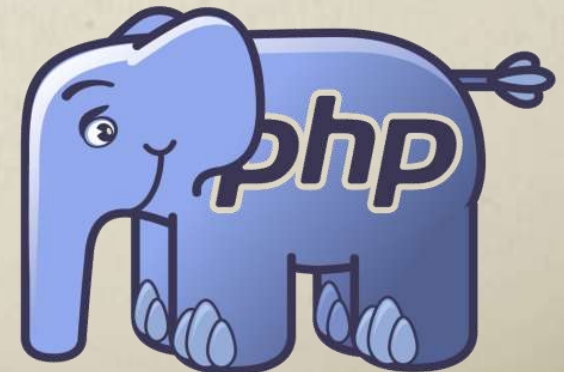
- `(bool)0 == false; (bool)'' == false`
- Ненулевое число, и непустая строка – это **true**
 - Исключение: `(bool)'0' == false`

Обратно:

- `(int>false == 0; (string>false == '';`
- `(int>true == 1; (string>true == '1';`



ТИП BOOLEAN



Булевы операторы – операторы, которые работают с булевыми значениями:

- **&&** – логическое «И»

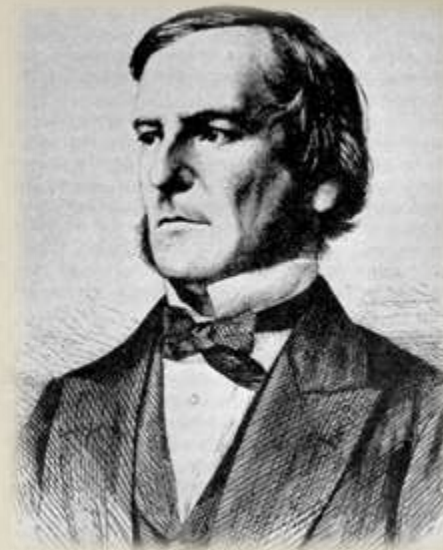
&&	0	1
0	0	0
1	0	1

- **||** – логическое «ИЛИ»

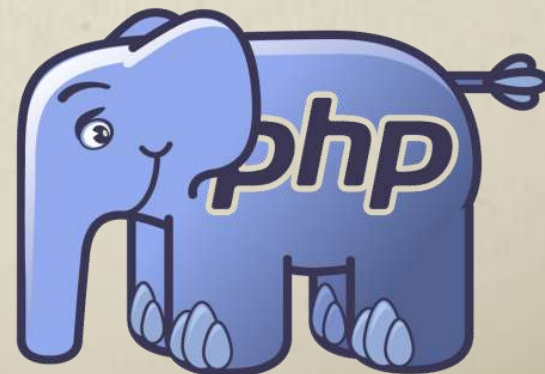
 	0	1
0	0	1
1	1	1

- **XOR** – логическое «ИСКЛЮЧАЮЩЕЕ ИЛИ»

xor	0	1
0	0	1
1	1	0



ТИП BOOLEAN



УСЛОВИЕ – это оператор PHP, определяющий, должен ли выполняться тот или иной код в зависимости от чего-либо

```
if ($x > $y) {  
    $m = $x;  
} else {  
    $m = $y;  
}
```

Сокращенная форма:

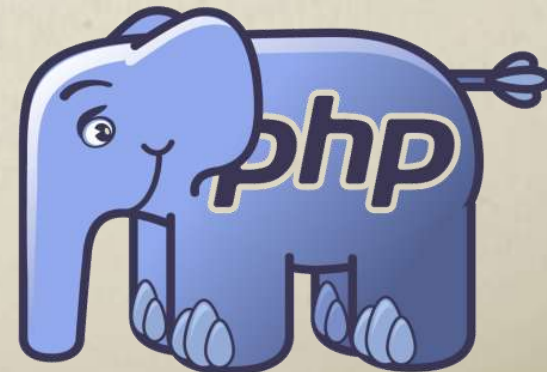
```
if ($stm) {  
    ...  
}
```

Форма с перебором условий:

```
if ($stm1) {  
    ...  
} elseif ($stm2) {  
    ...  
} else {  
    ...  
}
```

Важно! Всегда пишите фигурные скобки, даже если вам кажется, что это необязательно!

УСЛОВИЯ

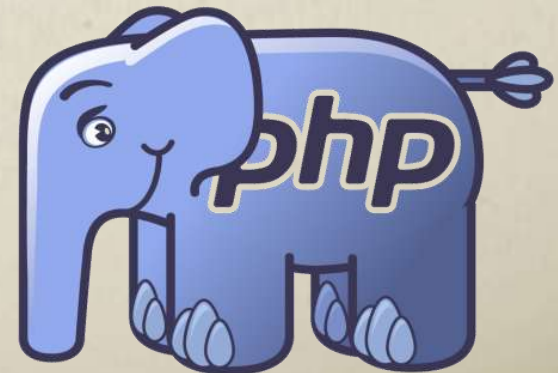


УСЛОВИЕ – это оператор PHP, определяющий, должен ли выполняться тот или иной код в зависимости от чего-либо

```
switch ($stm) {  
    case 1:  
        run1();  
        break;  
    case 2:  
        run2();  
        break;  
    default:  
        run();  
        break;  
}
```

- **break** прерывает перебор условий, не забывайте писать этот оператор!
- Секция **default** выполнится, если не совпало ни одно сравнение, ее может и не быть
- Обратите внимание, что **\$stm** и значения в **case** могут быть любыми – проверяется просто их равенство (нестрогое)
- Несколько **case** могут быть записаны один за другим и отвечать за один и тот же случай

УСЛОВИЯ



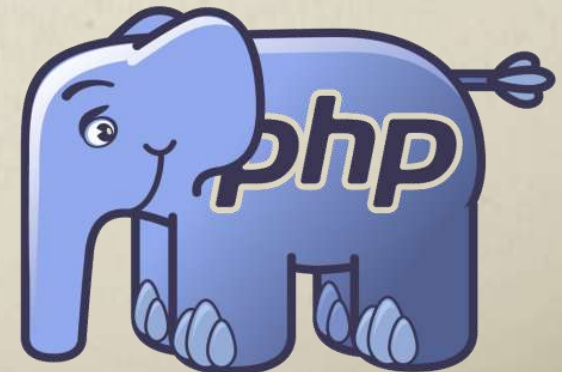
ФУНКЦИЯ – это подпрограмма, имеющая имя, набор входных значений (аргументов) и возвращаемое значение (необязательно)

```
function maxnum($a, $b) {  
    if ($a > $b) {  
        return $a;  
    } else {  
        return $b;  
    }  
}
```

```
$m = maxnum(3, 5);  
echo maxnum(-1, 1);
```

- Имя функции должно быть уникальным и не совпадать с именем библиотечной функции
- **return** – это оператор возврата значения из функции. Их может быть несколько или не быть вообще. Оператор прерывает выполнение функции!
- Все переменные, объявленные в функции, будут локальными и перестанут существовать после выхода из нее
- Глобальные переменные недоступны в функции

ФУНКЦИИ



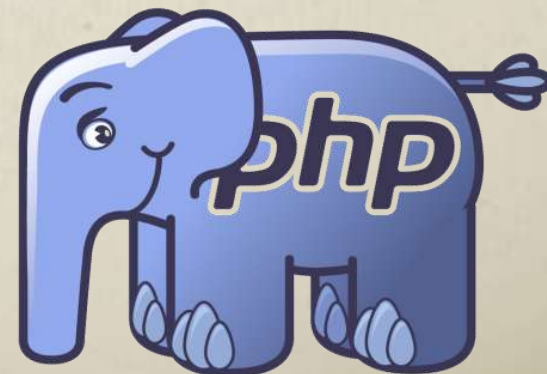
АРХИТЕКТУРНО ВЕРНО разделять свою программу на отдельные файлы.

В качестве первого шага к хорошей архитектуре можно собрать в один файл функции, а в другие файлы тот код, который их использует.

```
include __DIR__ . '/functions.php';
```

- **__DIR__** – это «магическая» константа. Она всегда содержит в себе полный путь в ФС до папки с текущим файлом
- **include** подключает файл, в случае его недоступности – выведет предупреждение, но продолжит работу программы
- **require** подключает файл, но в случае его недоступности вызовет фатальную ошибку и завершит программу
- **include_once** и **require_once** не будут подключать указанный файл, если он уже ранее был подключен

ВКЛЮЧЕНИЕ ФАЙЛОВ



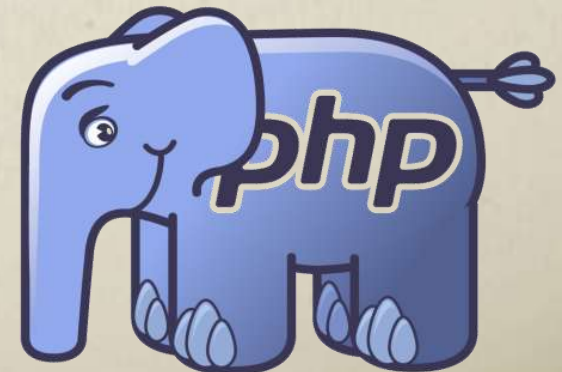
МОДУЛЬНЫЙ ТЕСТ – это код, содержащий утверждения о том, как должны работать отдельные модули (части) вашей программы

В PHP принято использовать модульные тесты для того, чтобы доказать, что ваша программа работает верно, и чтобы отслеживать ошибки при изменении кода.

```
function maxnumber($a, $b) {  
    ...  
}  
  
assert( 1 == maxnumber(1,1) );  
assert( 1 == maxnumber(-1,1) );  
assert( 1 == maxnumber(1,-1) );
```

- Старайтесь писать тесты до написания кода
- Используйте конструкцию `assert`, она встроена в язык
- Установите настройки:
 - `display_errors On`
 - `error_reporting E_ALL`
 - `zend.assertions = 1`

ТЕСТЫ



ДО ВСТРЕЧИ НА СЛЕДУЮЩЕМ УРОКЕ!

**ВИДЕОЗАПИСЬ, СЛАЙДЫ, ПРЕЗЕНТАЦИЯ
И ДОМАШНЕЕ ЗАДАНИЕ
БУДУТ ВЫЛОЖЕНЫ ДО 10 УТРА СЛЕДУЮЩЕГО ДНЯ**

