# PHP Web Development

```
$a = $c ? $d : $e;
```

Модуль 2. Основы PHP. Операторы

Автор: Владислав Бабак
linkedin.com/in/vladyslavbabak/

# Оператор присваивания

= - обозначается как “равно”. Оператор присваивания означает, что левый операнд получает значение правого выражения (правая ассоциативность - выполнение справа налево). Для изменения приоритета выполнения используются круглые скобки. Присвоение может осуществляться по значению (скаляры и массивы) и по ссылке (объекты). Для явного присвоения по ссылке используется амперсанд &.

Оператор присваивания может использоваться в сочетании с другими операторами = += -= *= **= /= .= %= &= |= ^= <<= >>=

```
<?php
$a = 10;
$b = $a;
$a = 15;
print $b;
```

# Оператор присваивания

```php
<?php

// передача по ссылке
$a = 10;
$b = &$a;
$a = 15;

var_dump($a, $b); // int(15) int(15)
```

# Арифметические операторы

+$a  - Идентичность (конвертация $a в int или float).
-$a - Отрицание (смена знака).
$a + $b - Сложение.
$a - $b - Вычитание.
$a * $b - Умножение.
$a / $b - Деление.
$a % $b - Деление по модулю (целочисленный остаток от деления).
$a ** $b - Возведение $a в степень $b.

# Арифметические операторы

```php
<?php

/**
 * Identity
 */
$a = '0.1';
$a = +$a;
var_dump($a);


/**
 * Negation
 */
$a = '5';
$a = -$a;
var_dump($a);
```

# Арифметические операторы

```php
<?php

/**
 * Modulo
 */
var_dump(5 % 2); // 5/2 = 2.5; 2*2 = 4; 5-4 = 1;
var_dump(13 % 3); // 13/3 = 4.3(3); 4*3 = 12; 13-12 = 1;

/**
 * Exponentiation
 */
var_dump(2 ** 3);
```

# Побитовые операторы

$a & $b - побитовое “и”. Устанавливаются только те биты, которые установлены и в $a, и в $b.

$a | $b - побитовое “или”. Устанавливаются те биты, которые установлены в $a или в $b.

$a ^ $b - “исключающее или”. Устанавливаются только те биты, которые установлены либо только в $a, либо только в $b, но не в обоих одновременно.

~ $a - побитовое отрицание. Устанавливаются те биты, которые не установлены в $a, и наоборот.

$a << $b - сдвиг влево. Все биты переменной $a сдвигаются на $b позиций влево (каждая позиция подразумевает "умножение на 2").

$a >> $b - сдвиг вправо. Все биты переменной $a сдвигаются на $b позиций вправо (каждая позиция подразумевает "деление на 2").

# Побитовые операторы

```php
<?php

/**
 * And
 */
$a = 0b0001; // 1
$b = 0b0011; // 3
$c = $a & $b; // 0b0001
var_dump(decbin($c), $c);

/**
 * Or
 */
$a = 0b0101; // 5
$b = 0b0011; // 3
$c = $a | $b; // 0b0111
var_dump(decbin($c), $c);
```

# Побитовые операторы

```php
<?php

/**
 * Xor
 */
$a = 0b0101; // 5
$b = 0b0011; // 3
$c = $a ^ $b; // 0b0110
var_dump(decbin($c), $c);

/**
 * Shift right
 */
$a = 0b0101; // 5
$c = $a >> 1; // 0b0010
var_dump(decbin($c), $c);
```

# Операторы сравнения

В случае, если вы сравниваете число со строкой или две строки, содержащие числа, каждая строка будет преобразована в число.

$a == $b равно true если $a равно $b после преобразования типов.
$a === $b - тождественно равно, строгое сравнение с учетом типа.
$a != $b - $a не равно $b после преобразования типов.
$a <> $b - $a не равно $b после преобразования типов.
$a !== $b - Тождественно не равно.
$a < $b - Меньше, true если $a строго меньше $b.
$a > $b - Больше.
$a <= $b Меньше или равно.
$a >= $b Больше или равно.
$a <=> $b Спейсшип. Этот оператор предназначен для сравнения двух выражений. Он возвращает -1, 0 или 1 если $a, соответственно, меньше, равно или больше чем $b.

# Операторы сравнения

```php
<?php

var_dump('1' < 2);

var_dump('2' >= 2);

var_dump('3' == 3);

var_dump('3' === 3);

var_dump(1 <=> 2); // -1, 1 < 2

var_dump(2 <=> 2); // 0, 2 == 2

var_dump(2 <=> '2'); // 0, 2 == ‘2’
```

# Оператор управления ошибками

@ - игнорирование сообщений об ошибках. Работает только с выражениями. Рекомендуется добавлять в код соответствующие проверки и не использовать оператор подавления ошибок.

# Оператор исполнения

`` - обратные кавычки, выполняет внешнюю программу аналогично shell_exec();

print `ls -la`; -  выводит листинг директории

# Операторы инкремента и декремента

++$a Префиксный инкремент. Увеличивает $a на единицу, затем возвращает значение $a.

$a++ Постфиксный инкремент. Возвращает значение $a, затем увеличивает $a на единицу.

--$a Префиксный декремент. Уменьшает $a на единицу, затем возвращает значение $a.

$a-- Постфиксный декремент. Возвращает значение $a, затем уменьшает $a на единицу.

# Операторы инкремента и декремента

```php
<?php

/**
 * post-increment
 */
$a = 10;
echo $a++;
echo $a;

/**
 * pre-increment
 */
$a = 10;
echo ++$a;
echo $a;
```

# Логические операторы

$a and $b И - TRUE если и $a, и $b TRUE.
$a or $b Или - TRUE если или $a, или $b TRUE.
$a xor $b Исключающее или - TRUE если $a, или $b TRUE, но не оба.
! $a Отрицание - TRUE если $a не TRUE.
$a && $b И - TRUE если и $a, и $b TRUE.
$a || $b Или - TRUE если или $a, или $b TRUE.

# Строковые операторы

. - конкатенация строки (объединение)
.= - присваивание с конкатенацией

```
$s = ‘hi, ’ . ‘Jack’;
$s .= ‘ How are you?’; // $s = $s . ‘’;
```

# Операторы для работы с массивом

$a + $b - Объединение массива $a и массива $b.
$a == $b - Равно TRUE в случае, если $a и $b содержат одни и те же пары ключ/значение.
$a === $b - Тождественно равно - TRUE в случае, если $a и $b содержат одни и те же пары ключ/значение в том же самом порядке и того же типа.
$a != $b - Не равно - TRUE, если массив $a не равен массиву $b.
$a <> $b - Не равно - TRUE, если массив $a не равен массиву $b.
$a !== $b - Тождественно не равно - TRUE, если массив $a не равен тождественно массиву $b.

# Оператор проверки типа

instanceof - используется для определения того, является ли текущий объект экземпляром указанного класса.

Оператор instanceof также может быть использован для определения, наследует ли определенный объект какой-либо класс или реализует ли класс заданный интерфейс.

# Тернарный оператор

```
$first_name = isset($_POST[‘first_name’]) ? $_POST[‘first_name’]  : ‘’;
$login = $name ? : ‘anonymous’;
```

Тернарный оператор "?:" является условным оператором.

Выражение (expr1) ? (expr2) : (expr3) интерпретируется как expr2, если expr1 имеет значение TRUE, или как expr3 если expr1 имеет значение FALSE.

Существует сокращенная форма. Выражение expr1 ? : expr3 возвращает expr1 если expr1 имеет значение TRUE, и expr3 в другом случае.

# Null coalescing

$name = $_POST[‘name’] ?? ‘’;  - оператор объединения с null (условный оператор). Он возвращает первый операнд, если он задан и не равен NULL, а в обратном случае возвращает второй операнд.

# Спасибо за внимание!

- остались вопросы?
- подведение итогов
- переход к практике
- домашнее задание


IN: linkedin.com/in/vladyslavbabak
Skype: marmalade.vlad
E-Mail: marmalade.vlad@gmail.com