**Конспект «Числа и строки в PHP»**

**Функция intval**

Функция intval принимает любое значение и всегда возвращает число. Если передать ей обычное число, она вернёт его же. Но если передать строку, функция попробует извлечь из неё число. Получить из строки число можно, если оно идёт в самом начале. Если числа в строке нет или если оно идёт не первым, функция вернёт 0:

keks\_log(intval('38 попугаев')); // Выведет: "38"

keks\_log(intval('Пятница 13')); // Выведет: "0"

**Тернарный оператор**

Тернарный оператор удобно использовать, если требуется выбрать одно из двух значений:

условие ? значение-1 : значение-2;

Если условие истинно, то PHP использует значение-1, а если ложно — значение-2. После условия ставится вопросительный знак, а значения разделяются двоеточием.

<p class="<?= get\_product\_is\_new($id) ? 'new' : 'old' ?>">...</p>

У тернарного оператора есть краткая запись:

значение-1 ?: значение-2

Краткая запись работает так: если значение-1 истинно, то используется оно. А если ложно, то используется значение-2.

$page = intval($\_GET['page']) ?: 1;

Ложными в PHP считаются число 0, строки '' и '0', булево значение false и массивы без элементов. Остальные числа, строки и массивы — истинные.

**Цикл for**

Цикл for похож на while, но синтаксис у него компактнее и при его использовании очень сложно забыть про счётчик.

for (перед циклом; условие; после итерации) {

тело цикла

}

Код в круглых скобках состоит из трёх частей, разделённых точкой с запятой:

* код, который выполнится **один раз** до начала работы цикла. Обычно здесь объявляют переменную-счётчик.
* условие, которое будет проверяться **перед каждой итерацией** цикла. Если условие истинно, код внутри тела выполнится, если ложно — цикл завершит работу.
* код, который будет выполняться **после каждой итерации**. Здесь обычно увеличивают переменную-счётчик.

Как и в других циклах, тело цикла for пишут в фигурных скобках.

for ($i = 1; $i <= 9; $i = $i + 1) {

keks\_log($i);

}

Чтобы внедрить цикл for в шаблон, используем такой синтаксис:

* PHP-теги пишем полностью — <?php ?>;
* вместо фигурных скобок ставим двоеточие — for ():;
* конец цикла обозначаем командой endfor, после которой идёт точка с запятой.

<ol>

<?php for ($i = 1; $i <= 9; $i = $i + 1): ?> // Начало цикла

<li>...</li> // Тело цикла

<?php endfor; ?> // Конец цикла

</ol>

**Функция ceil**

Функция ceil принимает число и, если оно дробное, округляет его вверх до ближайшего целого:

keks\_log(ceil(3.5)); // Выведет: "4"

keks\_log(ceil(3.99999)); // Выведет: "4"

keks\_log(ceil(3.00001)); // Выведет: "4"

В PHP также есть две другие функции для округления: floor и round. Первая округляет вниз, а вторая — в обе стороны до ближайшего целого.

keks\_log(floor(3.9)); // Выведет: "3"

keks\_log(round(3.4)); // Выведет: "3"

keks\_log(round(3.6)); // Выведет: "4"

**Функция date**

Функция date принимает строку-шаблон и возвращает дату в указанном формате.

$now = date('H:i:s d.m.Y');

Формат даты задаётся в строке-шаблоне с помощью управляющих символов. Вот некоторые из них:

| Символ | Значение |
| --- | --- |
| d | День месяца, 2 цифры с ведущим нулём |
| j | День месяца без ведущего нуля |
| F | Полное наименование месяца (на английском) |
| m | Порядковый номер месяца с ведущим нулём |
| n | Порядковый номер месяца без ведущего нуля |
| Y | Порядковый номер года, 4 цифры |
| y | Номер года, 2 цифры |
| H | Часы в 24-часовом формате с ведущим нулём |
| i | Минуты с ведущим нулём |
| s | Секунды с ведущим нулём |

Полный список символов можно посмотреть в [документации](https://www.php.net/manual/ru/function.date.php).

Символы, которые не являются управляющими, функция возвращает без изменений. Поэтому мы можем использовать в строке-шаблоне пробелы, точки и так далее.

По умолчанию функция date возвращает время той временной зоны, которая указана в настройках PHP на сервере. Временную зону можно изменить из сценария с помощью функции date\_default\_timezone\_set. Подробности в [документации](https://www.php.net/manual/ru/function.date-default-timezone-set.php).

**Одинарные и двойные кавычки**

В PHP для строк используют как одинарные ', так и двойные " кавычки. В большинстве случаев они работают одинаково, но не всегда. Например, если использовать переменную внутри одинарных кавычек, то выведется её имя. А если внутри двойных, то значение:

$name = 'Думпо';

keks\_log('Слона зовут $name'); // Выведет: "Слона зовут $name"

keks\_log("Слона зовут $name"); // Выведет: "Слона зовут Думпо"

Двойные кавычки помогают избежать запутанной конкатенации. О том, какие ещё есть различия между кавычками, вы можете узнать в [документации](https://www.php.net/manual/ru/language.types.string.php).