

The background is a dark, textured surface with faint, light-colored sketches of various scientific and educational symbols. These include a globe in the upper left, a large letter 'V' in the top left, a microscope on the left side, a stack of books at the bottom left, a cross symbol, an open book with handwritten text in the bottom center, a percentage sign and an exclamation mark in the bottom right, and a less-than sign on the far right.

# Библиотеки РНР

Преподаватель – Борисова Татьяна Андреевна

## Сегодня на занятии:

- Проверка домашнего задания
- Обсуждение вопросов по пройденному материалу
- Библиотеки. Зачем они нужны?
- Чтение документации
- Домашние задание

# Библиотеки

PHP, как и любой другой язык, поставляется с набором полезных функций. Все вместе они составляют так называемую **стандартную библиотеку**. В неё обычно входят тысячи функций, которые невозможно выучить — этого и не нужно делать. Подразумевается, что любой программист знает, где искать документацию по ним и примерно представляет себе, чего он хочет достичь. А дальше — дело техники. Программировать без интернета крайне сложно.



# Библиотеки

Для новичков эта информация часто выглядит так: «Сходи туда, не знаю куда, принеси то, не знаю что». То есть непонятно, как узнавать про эти функции, когда ты ничего не знаешь вообще. Как ни странно, не существует способа раз и навсегда познать всё, что нужно познать. Любой разработчик в процессе своего профессионального взросления знакомится со всё более интересными функциями, решающими его задачи более элегантно, и таким образом пополняет свой арсенал.

# Библиотеки

Вот некоторые советы, как узнавать о новых функциях:

- Всегда чётко отслеживайте, с чем вы сейчас работаете (какой тип данных). Почти всегда вы найдете необходимую функцию в соответствующем разделе документации, например, для работы со строками нужно изучать строковые функции.
- Периодически открывайте раздел со стандартными функциями по изучаемой тематике и просто пробегайтесь по ним, изучая сигнатуры и способы использования.
- Чаще читайте чужой код, особенно код библиотек, которые вы используете. Он весь доступен на GitHub.

# Библиотеки

Ещё одна особенность функций PHP из стандартной библиотеки в том, что они *доступны глобально*. Другими словами, вы можете вызывать их в любом месте вашей программы (это не работает с функциями, написанными самостоятельно). Большинство функций стандартной библиотеки сосредоточено вокруг какой-то функциональности или какого-то типа данных. Например, есть большой блок функций по обработке строк, по работе с числами и так далее. По некоторым из них мы пройдемся в процессе обучения.

# Задание

- Функция `gettype()` позволяет определить тип передаваемого аргумента. Название типа возвращается в виде строки. Например, вызов `gettype(10)` вернёт строку `"integer"` (`integer` — целое число).
- Выведите на экран тип значения переменной `$motto`.

Решение на следующем слайде.

# Решение задачи

```
<?php  
$motto = 'Family, Duty, Honor';  
// BEGIN  
    print_r(gettype($motto));  
// END
```



# Чтение документации

- Умение читать документацию — один из важнейших навыков любого разработчика.
- *Несмотря на наличие перевода, я рекомендую учиться пользоваться англоязычной документацией. Библиотеки, которые сотнями используются в реальных проектах, почти всегда имеют только англоязычное описание.*
- Основной раздел, к которому программисты возвращаются постоянно — справочник функций языка. Во-первых, этих функций много и невозможно запомнить всё. Во-вторых, в PHP особенно остро стоит проблема несогласованности параметров и возвращаемых значений функций. Поэтому постоянно приходится вспоминать их порядок.

# Чтение документации

Для описания функций используется собственная нотация (свой собственный синтаксис). Она немного напоминает сам PHP, но всё же отличается от него, другими словами — это не код на PHP, а только лишь описание сигнатуры функции. Рассмотрим функцию `round()`, которая округляет число:

```
round ( float $val [, int $precision = 0 [, int $mode = PHP_ROUND_HALF_UP ]] ) : float
```

Задача этого формата — наглядно показать типы, количество и значения по умолчанию входящих параметров, а также тип выходного параметра. Последний всегда стоит в самой правой позиции, прямо после знака двоеточия. В данном случае тип возвращаемого значения `float`.

# Чтение документации

Перечисление параметров идет через запятую в скобках после названия функции. Перед каждым именем параметра стоит его тип. Например, тип параметра `$val` — это `float`. Значение по умолчанию задается как присваивание, например `$precision` по умолчанию равно нулю.

Квадратные скобки `[]` показывают необязательные параметры функции, а запятые отделяют их друг от друга. Для функции выше это `$precision` и `$mode`. `$mode`, в свою очередь, по умолчанию равен значению константы `PHP_ROUND_HALF_UP`. Если какой-либо параметр равен значению константы, значит в документации этой функции есть перечисление и описание всех возможных констант, которые могут быть использованы как значение этого параметра.

# Чтение документации

Основываясь на сказанном выше, мы можем вызвать функцию следующим образом:

```
<?php
```

```
round(5.3); // 5.0
```

```
round(8.333, 1); // 8.3
```

```
round(8.333, 2, PHP_ROUND_HALF_UP); // 8.33
```

```
?>
```



# Задание

Реализуйте функцию `getAge()`, которая округляет возраст в нижнюю сторону. То есть если человеку десять с половиной лет, то функция должна вернуть 10.

Используйте для этого стандартную функцию `floor()`, которая округляет число в меньшую сторону.

```
<?php
```

```
getAge(10); // 10.0
```

```
getAge(9.1); // 9.0
```

```
getAge(8.9); // 8.0
```

```
?>
```

# Решение

```
<?php
namespace HexletBasics\DefineFunctions\ReadingDocumentation;
// BEGIN
function getAge($val) {
    return floor($val);
}
// END
?>
```

# Домашние задание

Выполнить тест по пройденному материалу