

Запись на курсы по HTML, CSS, JavaScript, PHP, фреймворкам и CMS,
а также: помощь в поиске работы и заказов, стажировка на реальных проектах→

урок 15 из 107

[Верстка](#) [JavaScript](#) [PHP](#) [NodeJs](#) [Vue](#) [React](#) [Laravel](#) [WordPress](#) [AJAX](#) [Парсинг](#)[Бесплатные курсы по React для новичков. Начало 4-го ноября→](#) [Конкурс CSS картинок. Тема: Хэллоуин. Призовой фонд: 100\\$. Подробности→](#)

Переменные названия методов

По аналогии с названиями свойств в переменной также можно хранить и имена методов. Давайте посмотрим на примере. Пусть у нас дан вот такой класс **User** с геттерами свойств:

```
1  <?php
2  class User
3  {
4      private $name;
5      private $age;
6
7      public function __construct($name, $age)
8      {
9          $this->name = $name;
10         $this->age = $age;
11     }
12
13     public function getName()
14     {
15         return $this->name;
16     }
17
18     public function getAge()
19     {
20         return $this->age;
21     }
22 }
23 ?>
```

Пусть в переменной **\$method** хранится имя метода. Давайте вызовем метод с таким именем:

```
1  <?php
2  $user = new User('Коля', 21);
3
4  $method = 'getName';
5  echo $user->$method(); // выведет 'Коля'
6  ?>
```

Если имя метода получается из массива, то такое обращение к методу следует брать в фигурные скобки вот таким образом (круглые скобки будут снаружи фигурных):

```
1  <?php
2  $user = new User('Коля', 21);
3
4  $methods = ['getName', 'getAge'];
5  echo $user->{$methods[0]}(); // выведет 'Ко
6  ля'
```

Все остальные нюансы точно такие же, как и при работе со свойствами из переменной.

Задача 15.1

Пусть массив **\$methods** будет ассоциативным с ключами **method1** и **method2**:

```
1 | <?php
2 |     $methods = ['method1' => 'getName', 'method2' => 'getAge'];
3 | ?>
```

Выведите имя и возраст пользователя с помощью этого массива.

←

→