

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

урок 11 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: Хэллоун. Призовой фонд: 100\\$. Подробности→](#)

# Хранение объектов в массивах

Пусть у нас дан вот такой класс **User**:

```
1  <?php
2  class User
3  {
4      public $name;
5      public $age;
6
7      public function __construct($name, $age)
8      {
9          $this->name = $name;
10         $this->age = $age;
11     }
12 }
13 ?>
```

Давайте создадим 3 объекта этого класса:

```
1  <?php
2  $user1 = new User('Коля', 21);
3  $user2 = new User('Вася', 22);
4  $user3 = new User('Петя', 23);
5  ?>
```

Давайте теперь запишем созданные нами объекты в массив **\$users**:

```
1  <?php
2  $user1 = new User('Коля', 21);
3  $user2 = new User('Вася', 22);
4  $user3 = new User('Петя', 23);
5
6  $users[] = $user1;
7  $users[] = $user2;
8  $users[] = $user3;
9
10 var_dump($users);
11 ?>
```

В общем-то переменные, в которые мы сохраняем наши объекты, и не нужны. Можем сократить наш код:

```
1  <?php
2  $users[] = new User('Коля', 21);
3  $users[] = new User('Вася', 22);
4  $users[] = new User('Петя', 23);
5
6  var_dump($users);
7  ?>
```

Давайте теперь переделаем наш код в другом стиле - добавим элементы в массив сразу при его создании:

```
1  <?php
2  $user1 = new User('Коля', 21);
3  $user2 = new User('Вася', 22);
4  $user3 = new User('Петя', 23);
5
```



```
6 | $users = [$user1, $user2, $user3];
7 | var_dump($users);
8 | ?>
```

Здесь также можно избавиться от промежуточных переменных:

```
1 | <?php
2 | $users = [
3 |     new User('Коля', 21),
4 |     new User('Вася', 22),
5 |     new User('Петя', 23)
6 | ];
7 |
8 | var_dump($users);
9 | ?>
```

Неважно каким способом мы создаем наш массив с объектами - важен сам принцип: объекты можно хранить в массивах.

Затем эти объекты можно, к примеру, перебрать циклом. Давайте сделаем это:

```
1 | <?php
2 | $users = [
3 |     new User('Коля', 21),
4 |     new User('Вася', 22),
5 |     new User('Петя', 23)
6 | ];
7 |
8 | // Переберем созданный массив циклом:
9 | foreach ($users as $user) {
10 |     echo $user->name . ' ' . $user->age . '<b
    r>';
11 | }
12 | ?>
```

## Задача 11.1

Сделайте класс **City** (город), в котором будут следующие свойства: **name** (название города), **population** (количество населения).

## Задача 11.2

Создайте 5 объектов класса **City**, заполните их данными и запишите в массив.

## Задача 11.3

Переберите созданный вами массив с городами циклом и выведите города и их население на экран.

