

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

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→

Введение в классы и объекты

Сейчас мы с вами займемся изучением ООП в PHP. Давайте рассмотрим пример из жизни, а потом перенесем его на PHP.

В качестве примера возьмем автомобиль. У него есть колеса, цвет, вид кузова, объем двигателя и так далее. Кроме того, водитель может отдавать ему команды: ехать, остановится, повернуть направо, налево и тп.

Можно говорить о том, что существует некоторый *класс* автомобилей, обладающий общими свойствами (у всех есть колеса и всем им можно отдавать команды).

Конкретный автомобиль, стоящий на улице - это представитель этого класса, или, другими словами, *объект* этого класса.

У всех объектов этого класса есть *свойства* (их также называют *полями*): количество колес, цвет, вид кузова и *методы*: ехать, остановится, повернуть направо, налево.

Другими словами сам класс - это **чертеж**, по которым на заводе делаются автомобили. Объект же - это сама машина, сделанная по этим чертежам.

В PHP класс создается с помощью ключевого слова **class**, за которым следует название этого класса.

Давайте сделаем класс **Car** (то есть Автомобиль):

```
1  <?php
2      class Car
3      {
4          // тут код, то есть PHP-чертеж автомобиля
5      }
6  ?>
```

PHP код данного урока достаточно абстрактный и нужен для того, чтобы вы первично ознакомились с синтаксисом классов. Не нужно пробовать его запускать - просто изучайте его, а что-то более практичное мы начнем делать уже со следующего урока и там каждый пример кода вы можете запустить у себя.

Укажем теперь в нашем чертеже, что любой автомобиль, созданный по этому чертежу, будет иметь свойство для цвета и свойство для количества топлива.

Для этого внутри класса напомним свойство **\$color** (цвет) и свойство **\$fuel** (топливо):

```
1  <?php
2      class Car
3      {
4          // Зададим свойства (по сути переменные класса):
5          public $color; // цвет автомобиля
6          public $fuel; // количество топлива
7      }
8  ?>
```

Ключевое слово **public** регулирует область видимости наших свойств. Подробнее данное ключевое слово мы будем изучать в следующих уроках, пока просто указывайте его при объявлении свойств и методов.

Давайте теперь сделаем методы нашего класса. В PHP методы, подобно обычным функциям, объявляются с помощью ключевого слова **function**, перед которым пишется ключевое слово **public**.

Как уже упоминалось выше, наш автомобиль может ехать, может поворачивать, может останавливаться. Сделаем соответствующие методы в нашем классе:



```
1  <?php
2  class Car
3  {
4      public $color; // цвет автомобиля
5      public $fuel; // количество топлива
6
7      // Команда ехать:
8      public function go()
9      {
10         // какой-то PHP код
11     }
12
13     // Команда поворачивать:
14     public function turn()
15     {
16         // какой-то PHP код
17     }
18
19     // Команда остановиться:
20     public function stop()
21     {
22         // какой-то PHP код
23     }
24 }
25 ?>
```

Мы с вами сделали чертеж нашего автомобиля. Теперь нужно отправится на завод и сделать объект этого класса (то есть конкретный автомобиль).

В PHP это делается с помощью ключевого слова **new**, после которого пишется имя класса:

```
1  <?php
2  new Car; // командуем заводу сделать автомобиль
3  ?>
```

Однако, если просто создать объект класса - это ни к чему не приведет (это все равно, что, к примеру, объявить массив и нигде его не записать). Нам нужна переменная для хранения этого объекта.

Пусть эта переменная будет называться **\$myCar** (мойАвтомобиль) - запишем в нее созданный нами объект:

```
1  <?php
2  $myCar = new Car; // запишем созданный объект в п
   еременную $myCar
3  ?>
```

После создания автомобиля можно обращаться к его свойствам. Обращение к ним происходит через стрелочку ->, вот так: *имяОбъекта->егоСвойство*.

Давайте установим свойства нашего объекта:

```
1  <?php
2  $myCar = new Car; // командуем заводу сделать авт
   омобиль
3
4  // Устанавливаем свойства объекта:
5
6  $myCar->color = 'red'; // красим в красный цвет
7  $myCar->fuel = 50; // заливаем топливо
8  ?>
```

Все, наш автомобиль создан, покрашен и заправлен. Теперь мы можем отдавать ему команды через методы этого автомобиля.

Обращение к методам также происходит через стрелочку ->, но, в отличие от свойства, после имени метода следует писать круглые скобки, вот так: *имяОбъекта->егоМетод()*.

Давайте покомандуем нашим объектом:

```
1  <?php
2  // Задаем команды созданному автомобилю:
```

```
3  $myCar->go(); // автомобиль->едь
4  $myCar->turn(); // автомобиль->поверни
5  $myCar->stop(); // автомобиль->остановись
6  ?>
```

