

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

урок 56 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→](#)

Класс Date в ООП на PHP

Начиная с данного урока мы с вами будем применять изученную теорию по ООП на практических примерах, создавая различные классы.

Давайте для разминки сделаем класс для работы с датой. Пусть этот класс параметром конструктора принимает дату в формате 'год-месяц-день' и имеет следующие методы:

```
1  <?php
2  class Date
3  {
4      public function __construct($date = null)
5      {
6          // если дата не передана - пусть берется тек
           ущая
7      }
8
9      public function getDay()
10     {
11         // возвращает день
12     }
13
14     public function getMonth($lang = null)
15     {
16         // возвращает месяц
17
18         // переменная $lang может принимать значение ru
           или en
19         // если эта не пуста - пусть месяц будет сло
           вом на заданном языке
20     }
21
22     public function getYear()
23     {
24         // возвращает год
25     }
26
27     public function getWeekDay($lang = null)
28     {
29         // возвращает день недели
30
31         // переменная $lang может принимать значение ru
           или en
32         // если эта не пуста - пусть месяц будет сло
           вом на заданном языке
33     }
34
35     public function addDay($value)
36     {
37         // добавляет значение $value к дню
38     }
39
40     public function subDay($value)
41     {
42         // отнимает значение $value от дня
43     }
44 }
```



```
45     public function addMonth($value)
46     {
47         // добавляет значение $value к месяцу
48     }
49
50     public function subMonth($value)
51     {
52         // отнимает значение $value от месяца
53     }
54
55     public function addYear($value)
56     {
57         // добавляет значение $value к году
58     }
59
60     public function subYear($value)
61     {
62         // отнимает значение $value от года
63     }
64
65     public function format($format)
66     {
67         // выведет дату в указанном формате
68         // формат пусть будет такой же, как в ф
        // ункции date
69     }
70
71     public function __toString()
72     {
73         // выведет дату в формате 'год-месяц-день'
74     }
75 }
76 ?>
```

Давайте посмотрим, как мы будем пользоваться нашим классом:

```
1  <?php
2  $date = new Date('2025-12-31');
3
4  echo $date->getYear(); // выведет '2025'
5  echo $date->getMonth(); // выведет '12'
6  echo $date->getDay(); // выведет '31'
7
8  echo $date->getWeekDay(); // выведет '3'
9  echo $date->getWeekDay('ru'); // выведет 'ср
   еда'
10 echo $date->getWeekDay('en'); // выведет 'we
   dnesday'
11 ?>

1  <?php
2  echo (new Date('2025-12-31'))->addYear(1); //
   '2026-12-31'
3  echo (new Date('2025-12-31'))->addDay(1); //
   '2026-01-01'
4
5  echo (new Date('2025-12-31'))->subDay(3)->add
   Year(1); // '2026-12-28'
6  ?>
```

Задача 56.1

Реализуйте описанный класс Date. Проверьте его работу.

