

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

урок 60 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→](#)

# Атрибуты тегов в классе Tag в ООП на PHP

Пусть теперь мы хотим сделать так, чтобы в создаваемых тегах можно было указывать атрибуты и их значения. Давайте будем передавать атрибуты для тега в виде ассоциативного массива в конструктор тега.

Вот пример (пока не рабочий, это наша цель):

```
1 | <?php
2 | $tag = new Tag('input', ['id' => 'test', '
   | class' => 'eee bbb']);
3 | e cho $tag->open(); // выведет <input id="test" cl
   | ass="eee bbb">
4 | ?>
```

Давайте сделаем в нашем классе вспомогательный приватный метод **getAttrsStr**, который параметром будет получать массив, а возвращать строку с атрибутами. Пример работы нашего метода:

```
1 | <?php
2 | $attrs = ['id' => 'test', 'class' => 'eee
   | bbb'];
3 | e cho $this->getAttrsStr($attrs); // выведет ' i
   | d="test" class="eee bbb"'
4 | ?>
```

Давайте напишем его реализацию:

```
1 | <?php
2 | private function getAttrsStr($attrs)
3 | {
4 |     if (!empty($attrs)) {
5 |         $result = '';
6 |
7 |         foreach ($attrs as $name => $value) {
8 |             $result .= " $name=\"$value\"";
9 |         }
10 |
11 |         return $result;
12 |     } else {
13 |         return '';
14 |     }
15 | }
16 | ?>
```

Добавим созданный метод в наш класс:

```
1 | <?php
2 | class Tag
3 | {
4 |     private $name;
5 |     private $attrs;
6 |
7 |     public function __construct($name, $attrs = [])
8 |     {
9 |         $this->name = $name;
10 |         $this->attrs = $attrs; // записываем атр
    |         ибуты в свойство
```

```
11 |     }
12 |
13 |     public function open()
14 |     {
15 |         $name = $this->name;
16 |         $attrsStr = $this->getAttrsStr($this->attrs); // формируем строку с атрибутами
17 |
18 |         return "<$name$attrsStr>"; // добавляем атрибуты после имени тега
19 |     }
20 |
21 |     public function close()
22 |     {
23 |         $name = $this->name;
24 |         return "</$name>";
25 |     }
26 |
27 |     // Формируем строку с атрибутами:
28 |     private function getAttrsStr($attrs)
29 |     {
30 |         if (!empty($attrs)) {
31 |             $result = '';
32 |
33 |             foreach ($attrs as $name => $value) {
34 |                 $result .= " $name=\"$value\"";
35 |             }
36 |
37 |             return $result;
38 |         } else {
39 |             return '';
40 |         }
41 |     }
42 | }
43 | ?>
```

Проверим работу нашего метода:

```
1 | <?php
2 |     $tag = new Tag('input', ['id' => 'test', 'class' => 'eee bbb']);
3 |     echo $tag->open(); // выведет <input id="test" class="eee bbb">
4 | ?>
```

## Задача 60.1

Самостоятельно, не подсматривая в мой код, добавьте в класс **Tag** возможность добавления атрибутов тега.

