

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

урок 61 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→](#)

# Методы цепочкой в классе Tag ООП на PHP

Давайте вынесем добавление атрибутов из конструктора в отдельный метод **setAttr**.

Пусть при этом метод **setAttr** позволяет добавлять только один атрибут, а чтобы добавить несколько атрибутов - нужно будет вызывать несколько **setAttr** в виде цепочки.

Смотрите пример использования нашего метода:

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

С комментариями:

```
1 <?php
2 $tag = new Tag('input');
3
4 echo $tag
5     ->setAttr('id', 'test') // добавляем атрибут 'id'
6     ->setAttr('class', 'eee bbb') // добавляем атрибут 'class'
7     ->open(); // выведет <input id="test" class="eee bbb">
8 ?>
```

Пока мы предполагаем, что наша цепочка всегда завершается методом **open**. Если тег должен иметь еще и закрывающую часть, то мы будем вызывать метод **close** вне цепочки, вот так:

```
1 <?php
2 $tag = new Tag('div');
3 echo $tag->setAttr('id', 'test')->open(); // откроем тег
4 echo $tag->close(); // закроем тег
5 ?>
```

Итак, давайте реализуем метод **setAttr** в нашем классе:

```
1 <?php
2 class Tag
3 {
4     private $name;
5     private $attrs = []; // изначально пустой массив
6
7     public function __construct($name)
8     {
9         $this->name = $name;
10    }
11
12    // Реализуем метод для атрибутов:
13    public function setAttr($name, $value)
14    {
15        $this->attrs[$name] = $value;
```

```
16 |         return $this; // возвращаем $this чтобы была цеп
    |         очка
17 |     }
18 |
19 |     public function open()
20 |     {
21 |         $name = $this->name;
22 |         $attrsStr = $this->getAttrsStr($th
    |         is->attrs);
23 |
24 |         return "<$name$attrsStr>";
25 |     }
26 |
27 |     public function close()
28 |     {
29 |         $name = $this->name;
30 |         return "</$name>";
31 |     }
32 |
33 |     private function getAttrsStr($attrs)
34 |     {
35 |         if (!empty($attrs)) {
36 |             $result = '';
37 |
38 |             foreach ($attrs as $name => $value) {
39 |                 $result .= " $name=\"$value\"";
40 |             }
41 |
42 |             return $result;
43 |         } else {
44 |             return '';
45 |         }
46 |     }
47 | }
48 | ?>
```

## Задача 61.1

Самостоятельно добавьте метод **setAttr** в созданный вами в предыдущем уроке класс **Tag**.

## Задача 61.2

Реализуйте метод **removeAttr**, который будет удалять переданный параметром атрибут.

Сделайте так, чтобы этот метод также мог принимать участие в цепочке.

