

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→

# CSS классы в классе Tag ООП на PHP

Пусть у нас дан вот такой инпут с атрибутом **class**:

```
1 | <input class="eee bbb kkk">
```

Как вы видите, атрибут **class** содержит в себе несколько значений, разделенных пробелами. Эти значения представляют собой несколько CSS классов нашего элемента.

Давайте реализуем в нашем классе **Tag** набор методов, которые будут работать с этими CSS классами.

Например, было бы удобно иметь метод **addClass**, добавляющий еще один класс в строку с классами.

При этом наш метод должен проверять то, что добавляемый класс еще не присутствует среди классов элемента, иначе добавление нового класса будет приводить к тому, что у элемента будет несколько одинаковых классов, что, конечно же, неправильно.

Пример использования желаемого нами метода:

```
1 | <?php
2 | // Выведет <input class="eee bbb">:
3 | echo (new Tag('input'))->addClass('eee')->
   |   addClass('bbb')->open();
4 | ?>
```

Было бы также удобно иметь метод **removeClass** для удаления заданного класса элемента.

Практическое применение этих методов вы еще увидите в следующих уроках.

Давайте реализуем описанные методы.

## Добавление класса

Как вы знаете, наш класс **Tag** хранит атрибуты тега в свойстве **\$this->attrs**.

Данное свойство представляет собой массив. В этом массиве может быть элемент с ключом **class**, содержащий CSS классы элемента. Элемента с таким ключом может, однако, и не быть.

Все зависит от того, был ли при создании тега вызван метод **setAttr** для установки атрибута **class** или нет.

Вот пример того, когда он был вызван:

```
1 | <?php
2 | // Выведет <input class="eee bbb">:
3 | echo (new Tag('input'))->setAttr('class', '
   |   eee bbb')->open();
4 | ?>
```

А вот пример того, когда он не был вызван:

```
1 | <?php
2 | // Выведет <input id="test">:
3 | echo (new Tag('input'))->setAttr('id', 'te
   |   st')->open();
4 | ?>
```

В реализации нашего метода **addClass** нужно будет учесть оба варианта.

Получается, что если у элемента не заданы CSS классы, то вызов метода **addClass** должен просто создавать в массиве **\$this->attrs** новый элемент с ключом **class** и записывать в него переданный класс:



```

1  <?php
2  public function addClass($className)
3  {
4      // Если ключа class нет в массиве $this->atts:
5      if (!isset($this->attrs['class'])) {
6          $this->attrs['class'] = $className; //
           запишем наш CSS класс
7      }
8
9      return $this; // возвращаем $this для работы цеп
           очки
10 }
11 ?>

```

Обратите внимание на то, что параметр метода называется **\$className**, а не **\$class**, так как слово **class** является зарезервированным в PHP и его нельзя использовать в качестве имени переменной.

Давайте теперь рассмотрим второй вариант, когда в **\$this->attrs['class']** уже есть один или несколько классов.

Как уже упоминалось выше, эти классы были добавлены ранее с помощью метода **setAttr**. Эти классы также могли быть добавлены с помощью вызова метода (или цепочки методов) **addClass**.

Способ добавления, в общем-то, не имеет никакого значения, главное, что, если классы есть, то они хранятся в виде строки, разделенные пробелами. Либо, если там один класс, то в **\$this->attrs['class']** просто хранится его имя, без пробелов.

Пусть в **\$this->attrs['class']** хранится несколько классов. В этом случае будет удобнее работать не со строкой с пробелами, а с массивом CSS классов. Для этого можно просто разбить нашу строку в массив с помощью функции **explode**:

```

1  <?php
2  public function addClass($className)
3  {
4      if (isset($this->attrs['class'])) {
5          // Получаем массив классов:
6          $classNames = explode(' ', $this->atts
           rs['class']);
7      }
8
9      return $this;
10 }
11 ?>

```

Затем необходимо проверить отсутствие переданного класса в этом массиве классов. Это легко сделать с помощью функции **in\_array**:

```

1  <?php
2  public function addClass($className)
3  {
4      if (isset($this->attrs['class'])) {
5          $classNames = explode(' ', $this->atts
           rs['class']);
6
7          // Если такого класса нет в массиве клас
           ссов:
8          if (!in_array($className, $classNames)) {
9              // добавим новый класс
10         }
11     }
12
13     return $this;
14 }
15 ?>

```



Если переданного класса нет в массиве классов, то добавим его к уже существующим классам. А если есть - то просто ничего не будем делать.

Давайте реализуем добавление класса:

```
1  <?php
2  public function addClass($className)
3  {
4      if (isset($this->attrs['class'])) {
5          $classNames = explode(' ', $this->atts
              ['class']);
6
7          if (!in_array($className, $classNames)) {
8              // Добавим новый класс в массив с классами:
9              $classNames[] = $className;
10
11             // Сольем массив в строку и запишем ее в $this->attrs['class']:
12             $this->attrs['class'] = implode(' ', $classNames);
13         }
14     }
15
16     return $this;
17 }
18 ?>
```

Рассмотрим теперь вариант, когда в **`$this->attrs['class']`** хранится только один класс. На самом деле, реализованный выше код будет прекрасно работать и в этом случае: применение **`explode`** к строке без пробела просто вернет массив из одного элемента, представляющего собой эту строку.

Ну и далее все наши манипуляции будут работать также.

Давайте соберем весь наш код вместе:

```
1  <?php
2  public function addClass($className)
3  {
4      if (isset($this->attrs['class'])) {
5          $classNames = explode(' ', $this->atts
              ['class']);
6
7          if (!in_array($className, $classNames)) {
8              $classNames[] = $className;
9              $this->attrs['class'] = implode(' ', $classNames);
10         }
11     } else {
12         $this->attrs['class'] = $className;
13     }
14
15     return $this;
16 }
17 ?>
```

## Задача 65.1

Самостоятельно реализуйте описанный метод и добавьте его в ваш класс **Tag**.

Проверьте работу созданного метода, используя приведенные ниже примеры:

```
1  <?php
2  // Выведет <input class="eee">:
3  echo (new Tag('input'))->addClass('eee')->
    open();
4  ?>
```

```
1 | <?php
2 | // Выведет <input class="eee bbb">:
3 | echo (new Tag('input'))->addClass('eee')->
   |   addClass('bbb')->open();
4 | ?>
```

```
1 | <?php
2 | // Выведет <input class="eee bbb kkk">:
3 | echo (new Tag('input'))
4 |     ->setAttr('class', 'eee bbb')
5 |     ->addClass('kkk')->open();
6 | ?>
```

```
1 | <?php
2 | // Выведет <input class="eee bbb">:
3 | echo (new Tag('input'))
4 |     ->setAttr('class', 'eee bbb')
5 |     ->addClass('eee') // такой класс уже есть и н
   |       е добавится
6 |     ->open();
7 | ?>
```

```
1 | <?php
2 | // Выведет <input class="eee bbb">:
3 | echo (new Tag('input'))
4 |     ->addClass('eee')
5 |     ->addClass('bbb')
6 |     ->addClass('eee') // такой класс уже есть и н
   |       е добавится
7 |     ->open();
8 | ?>
```

## Удаление класса

Давайте теперь реализуем удаление CSS класса.

Для этого было бы удобно иметь вспомогательный метод, который будет удалять элемент из массива по тексту этого элемента. В PHP, к сожалению, нет такой встроенной функции, поэтому реализуем ее в виде приватного метода:

```
1 | <?php
2 | private function removeElem($elem, $arr)
3 | {
4 |     $key = array_search($elem, $arr); //
   |       находим ключ элемента по его тексту
5 |     array_splice($arr, $key, 1); // удаляем эле
   |       мент
6 |
7 |     return $arr; // возвращаем измененный массив
8 | }
9 | ?>
```

Используя метод **removeElem** мы теперь можем реализовать метод **removeClass** для удаления CSS классов. Реализуем:

```
1 | <?php
2 | public function removeClass($className)
3 | {
4 |     if (isset($this->attrs['class'])) {
5 |         $classNames = explode(' ', $this->att
```



```
        rs['class']);
6 |
7 |     if (in_array($className, $classNames)) {
8 |         $classNames = $this->removeElem($className, $cl
          assNames);
9 |         $this->attrs['class'] = implode(' ', $cl
          assNames);
10 |     }
11 | }
12 |
13 | return $this;
14 | }
15 | ?>
```

## Задача 65.2

Самостоятельно реализуйте описанный метод и добавьте его в ваш класс **Tag**. Проверьте его работу, например, так:

```
1 | <?php
2 | echo (new Tag('input'))
3 |     ->setAttr('class', 'eee zzz kkk') // добавим 3 к
          ласса
4 |     ->removeClass('zzz') // удалим класс 'zzz'
5 |     ->open(); // выведет <input class="eee kkk">
6 | ?>
```

