

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

урок 30 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→

Статические свойства

Кроме статических методов можно также делать и статические свойства. Такие свойства также объявляются с помощью ключевого слова **static**:

```
1  <?php
2  class Test
3  {
4      // Статическое свойство:
5      public static $property;
6  }
7  ?>
```

Можно что-то записать в статическое свойство и прочитать из него:

```
1  <?php
2  Test::$property = 'test';
3  echo Test::$property; // выведет 'test'
4  ?>
```

Обратите внимание на то, что имя свойства указывается вместе с долларом.

Задача 30.1

Сделайте класс **Num**, у которого будут два публичных статических свойства: **num1** и **num2**. Запишите в первое свойство число 2, а во второе - число 3. Выведите сумму значений свойств на экран.

Статическое свойство внутри класса

Можно также обращаться к статическим свойствам внутри самого класса, используя **self::**, смотрите пример:

```
1  <?php
2  class Test
3  {
4      // Приватное статическое свойство:
5      private static $property;
6
7      // Статический метод для задания значения сво
8      public static function setProperty($value)
9      {
10         self::$property = $value; // записываем дан
11         ные в наше static свойство
12     }
13
14     // Статический метод для получения значения сво
15     public static function getProperty()
16     {
17         return self::$property; // прочитываем зап
```



```
17 |         исанные данные
18 |     }
19 | }>
```

Воспользуемся нашим классом:

```
1 | <?php
2 |     Test::setProperty('test'); // запишем дан
   |     ные в свойство
3 |     e cho Test::getProperty(); // выведем на экран
4 | }>
```

Можно также задать начальное значение свойства:

```
1 | <?php
2 | class Test
3 | {
4 |     // Начальное значение свойства:
5 |     private static $property = 'test';
6 |
7 |     public static function getProperty()
8 |     {
9 |         return self::$property;
10 |    }
11 | }
12 |
13 | echo Test::getProperty(); // выведет 'test'
14 | }>
```

Задача 30.2

Сделайте класс **Num**, у которого будут два приватны статических свойства: **num1** и **num2**. Пусть по умолчанию в свойстве **num1** хранится число 2, а в свойстве **num2** - число 3.

Задача 30.3

Сделайте в классе **Num** метод **getSum**, который будет выводить на экран сумму значений свойств **num1** и **num2**.

Применение

Пусть у нас есть класс **Geometry** для работы с геометрическими фигурами.

В этом классе есть методы для определения площади круга и длины окружности:

```
1 | <?php
2 | class Geometry
3 | {
4 |     // Площадь круга:
5 |     public static function getCircleSquare($ra
   |     dius)
6 |     {
7 |         return 3.14 * $radius * $radius;
8 |     }
9 |
10 |    // Длина окружности:
11 |    public static function getCircleCircuit($ra
   |    dius)
12 | }
```



```
13 |         return 2 * 3.14 * $radius;
14 |     }
15 | }
16 | ?>
```

Как вы видите, в обоих методах используется число Пи, равное **3.14**.

Было бы удобно вынести это число в статическое свойство класса - в этом случае значение числа Пи будет задаваться в одном месте и мы легко сможем поменять его в случае необходимости (например, написать больше знаков в дробной части).

Давайте сделаем это:

```
1 | <?php
2 | class Geometry
3 | {
4 |     private static $pi = 3.14; // вынесем Пи в с
   |     ВОЙСТВО
5 |
6 |     public static function getCircleSquare($ra
   |     dius)
7 |     {
8 |         return self::$pi * $radius * $radius;
9 |     }
10 |
11 |     public static function getCircleCircuit($ra
   |     dius)
12 |     {
13 |         return 2 * self::$pi * $radius;
14 |     }
15 | }
16 | ?>
```

Задача 30.4

Добавьте в наш класс **Geometry** метод, который будет находить объем шара по радиусу. С помощью этого метода выведите на экран объем шара с радиусом 10.

