Magic Methods

## PHP \_\_get() and \_\_set() methods

PHP provides a pair of magic methods that deal with non-existent or private property access:

* The \_\_get() method is called whenever you attempt to read a non-existing or private property of an object.
* The \_\_set() method is called whenever you attempt to write to a non-existing or private property of an object.

Let’s take a look at the following example:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | <?php    class Person{  private $firstName;    public function \_\_get($propertyName){  echo "attempted to read non-existing property: $propertyName  ";  }  public function \_\_set($propertyNane, $propertyValue){  echo "attempted to write to non-existing property: $propertyNane  ";  }    }    $p = new Person();    $p->firstName = 'Doe';  echo $p->firstName;    $p->lastName = 'John';  echo $p->lastName; |

How it works.

* First, we attempted to write and read a private property: $firstName.  The method \_\_get()and \_\_set() methods are called automatically.
* Second, we attempted to write and read a non-existing property: $lastName. The method\_\_get() and \_\_set() methods are also called automatically.

The \_\_get() and \_\_set() methods are used to define properties of a class as an associative array instead of separate properties. The following example demonstrates the idea:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | <?php    class Person{  private $properties;    public function \_\_get($propertyName){  if(array\_key\_exists($propertyName, $properties)){  return $this->properties[$propertyName];  }    }  public function \_\_set($propertyNane, $propertyValue){  $this->properties[$propertyNane] = $propertyValue;  }    }    $p = new Person();  $p->lastName = 'John';  $p->firstName = 'Doe';    var\_dump($p); |

By using the \_\_get() and \_\_set() method, the Person class can hold any number of  “virtual” properties.

## PHP \_\_call() method

Similar to the \_\_get() and \_\_set() methods, the \_\_call() magic method is called automatically when a nonexistent method of the class is called. The \_\_call() method is rarely used in practice. It is useful when you to create a wrapper class that wraps the existing APIs.

The following example illustrates a simple string class that supports three methods: strlen(), strtoupper() and strtolower().

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | <?php    class CString{  private $str = '';    private $APIs = array('strlen','strtoupper','strtolower');    public function \_\_construct($str){  $this->str = $str;    }  public function \_\_call($method,$args){  if(in\_array($method, $this->APIs)){  array\_unshift($args, $this->str);  return call\_user\_func\_array($method, $args);  }else{  die('Error: call to unsupported method: ' . $method);  }  }  }    $str = new CString('This is a string object');  echo $str->strlen() . '  ';  echo $str->strtoupper() . '  ';  echo $str->strtolower() . '  ';    echo $str->len(); // Error: call to unsupported method: len |

How it works.

* Whenever you call a nonexistent method, the \_\_call() method is called automatically. Inside the \_\_call() method, we check to see if the calling method is supported, which is defined in the $APIs array. If it is, we execute the corresponding string function, otherwise, we issue a message.

PHP uses double underscores ( \_\_ )  as a prefix for the magic methods in the class so it is recommended that you should not use the method name that begins with double underscores ( \_\_ ).

Notice that the [constructor and destructor methods](http://www.zentut.com/php-tutorial/php-constructor-and-destructor/) ( \_\_construct() and \_\_destruct() ) are also magic methods. For other magic methods, check it out the [PHP magic methods](http://www.php.net/manual/en/language.oop5.magic.php).

In this tutorial, you have learned how to use three PHP magic methods:\_\_get() \_\_set() and\_\_call() that deals with non-existent members and methods of a class.