

PHP oops - Magic Method

Topics : [PHP](#)

Written on [February 17, 2021](#)

What is Magic Method in php?

Magic methods are special methods that are used to perform certain tasks in PHP. Magic methods are started with double underscore (__) as prefix.

- Magic methods is called automatically and does not require any function call to execute the code inside these functions in php.
- The function `__construct()`, `__destruct()`, `__call()`, `__callStatic()`, `__get()`, `__set()`, `__isset()`, `__unset()`, `__sleep()`, `__wakeup()`, `__serialize()`, `__unserialize()`, `__toString()`, `__invoke()`, `__set_state()`, `__clone()` and `__debugInfo()` are called as Magic Methods in php
- Magic method in a class must be declared public.

List of Magic Methods in PHP

1. `__construct()`

`__construct()` method is called when object is created.

PHP Example:

```
<?php class Aryatechno {
    public function __construct()
    {
        echo "construct is called when object is created.";
    }
}
```

```
$obj=new Aryatechno();
```

```
?>
```

Output:

construct is called when object is created.

2. `__destruct()`

`__destruct()` method is called when object is destroyed.

PHP Example:

```
<?php class Aryatechno {
    public function __destruct()
    {
```

```

        echo "Destructor is called when object is destroyed.";
    }
}

```

```

$obj=new Aryatechno();
?>

```

Output:

Destructor is called when object is destroyed.

3. **__call(\$function_name, \$argument)**

__call() is triggered when trying to access inaccessible methods in an object context.

PHP Code Example:

```

<?php
//code by aryatechno for Method Overloading
class Aryatechno {

    function __call($name,$arg){

        echo "<br/> Magic method __call is triggered when trying to acces non-existing method
course()";
        echo "<br/> Method Name :".$name;
        echo "<br/> Passes Argument in array:".$arg[0];
    }
}
$obj = new Aryatechno();
$obj->course("php");
?>

```

Output:

Magic method __call is triggered when trying to acces non-existing method course()
Method Name :course
Passes Argument in array:php

4. **__callStatic(\$function_name, \$argument)**

__callStatic() is triggered when trying to access inaccessible methods in a static context.

PHP Code Example:

```

<?php
class Aryatechno
{

    public static function __callStatic($fun, $arg) {
        echo "<br/>Method that does not exist[]" . $fun . " ";
        echo "<br/>parameter list of method that does not exist:"; print_r($arg);
    }
}
$obj = new Aryatechno();
$obj::course("PHP"); // If the method does not exist is called within the object in static
manner, then the __callStatic() method will be called automatically.
?>

```

Output:

Method that does not exist course

parameter list of method that does not exist:Array ([0] => PHP)

5. __get(\$property_name)

__get() is called when reading data from inaccessible (like protected or private) or non-existing properties.

PHP Example:

```
<?php
//code by aryatechno for Method Overloading
class Aryatechno
{
    function __get($name)
    {
        echo "<br/> __get magic method is called when reading inaccessible property";
        echo "<br/> Inaccessible variable Name :".$name;
    }
}

$obj = new Aryatechno;
echo $obj->j; //variable j is not available
?>
```

Output:

__get magic method is called when reading inaccessible property
Inaccessible variable Name :j

6. __set(\$property_name, \$value)

__set() is triggered when writing data to inaccessible (like protected or private) or undefined properties.

PHP Example:

```
<?php
//code by aryatechno for Method Overloading

class Aryatechno
{
    function __set($name,$val)
    {
        echo "<br/> __set magic method is called when writing value into inaccessible
property";
        echo "<br/> Inaccessible variable Name :".$name;
        echo "<br/> variable value :".$val;
    }
}
```

```
}  
  
$obj = new Aryatechno;  
$obj->i=100; ////variable i is not available  
?>
```

Output:

__set magic method is called when writing value into inaccessible property
Inaccessible variable Name :i
variable value :100

7. **__isset(\$property_name)**

__isset() is triggered when isset() or empty() is called on inaccessible (like protected or private) or non-existing properties.

8. **__unset(\$property_name)**

__unset() is triggered when unset() is called on inaccessible (like protected or private) or non-existing properties.

PHP Example:

```
<?php  
//code by aryatechno for Method Overloading  
  
class Aryatechno  
{  
  
    function __isset($name)  
    {  
        echo "<br/> __isset magic method is called when isset() method is called for inaccessible  
property";  
        echo "<br/> variable Name :".$name;  
    }  
  
    function __unset($name)  
    {  
        echo "<br/> __unset magic method is called when unset() method is called for  
inaccessible property";  
        echo "<br/> Inaccessible variable Name :".$name;  
    }  
  
}  
  
$obj = new Aryatechno;  
isset($obj->p); //variable p is not available  
unset($obj->p);  
  
?>
```

Output:

__isset magic method is called when isset() method is called for inaccessible property variable Name :p

__unset magic method is called when unset() method is called for inaccessible property Inaccessible variable Name :p

9. __sleep()

__sleep() method is called first while executing serialize(). It returns the object's property array on cleaning PHP class objects before serialization.

PHP Example:

```
<?php
//code by aryatechno for Method Overloading

class Aryatechno
{

    public function __construct($val)
    {
        $this->name = $val;
    }

    public function __sleep()
    {
        echo "<br/>It is called when the serialize() method is called outside the class.";
        return array('name');
    }

}

$obj = new Aryatechno("PHP");
echo serialize($obj);

?>
```

Output:

It is called when the serialize() method is called outside the class.O:10:"Aryatechno":1:{s:4:"name";s:3:"PHP";}

10. __wakeup()

__wakeup() method is called while deserialization() is executed. It would reverse work to restore objects properties and resources on invoking deserialization().

PHP Example:

```
<?php
//code by aryatechno for Method Overloading

class Aryatechno
{

    public function __construct($val)
    {
```

```

        $this->name = $val;
    }

    public function __sleep()
    {
        echo "<br/>It is called when the serialize() method is called outside the class.";
        return array('name');
    }

    public function __wakeup()
    {
        echo "It is called when the unserialize() method is called outside the class.
";
        $this->name = "java";
    }
}

$obj = new Aryatechno("PHP");
echo serialize($obj);
print_r(unserialize(serialize($obj)));
?>

```

Output:

It is called when the serialize() method is called outside the class.O:10:"Aryatechno":1:{s:4:"name";s:3:"PHP";}

It is called when the unserialize() method is called outside the class. Aryatechno Object ([name] => java)

11. **__toString()**

__toString() method will be called while using echo method to print an object directly. It is expected to return a string value while using class instances with PHP printing statements.

12. **__invoke()**

__invoke() method is called while trying to call an object in a way of calling function.

13. **__set_state(\$array)**

__set_state method is triggered while calling **var_export()**. It is a static method invoked while exporting objects property array and expects such array variable as its argument.

14. **__clone()**

__clone() method is triggered when the object is copied.

PHP Example:

```

<?php
//code by aryatechno for Method Overloading

class Aryatechno
{

    public function __clone()
    {

```

```
        echo " You are cloning the object.".__METHOD__;
    }
}

$obj = new Aryatechno();
$obj2 = clone $obj;
?>
```

Output:

You are cloning the object. Aryatechno::__clone

15. **__debugInfo()**

__debugInfo() method is called when using var_dump()

PHP Example:

```
<?php class Aryatechno {
    public function __construct()
    {
    }

    public function __debugInfo()
    {
        echo "<br/> Debug info.";
    }
}
var_dump(new Aryatechno());
?>
```

Output:

Debug info.object(Aryatechno)#1 (0) { }

Output :