<p>PHP has some of its method’s name with \_\_ (double underscore) as prefix. These methods are called magic method. It is recommended by PHP not to create any custom method as a name with \_\_ prefix. Example of available magic methods in PHP are \_\_toString(), \_\_sleep(), \_\_isset(), \_\_set(), \_\_construct() etc.</p>

<p>Today we are going to talk about \_\_toString method of PHP. </p>

<h2>\_\_toString() Method</h2>

<p>This method allows a class to decide how to react when an object of that class is being used as a string. </p>

<p>For example, let’s think of a very simple class named as <strong>TestClass</strong></p>

<pre><code>&lt;?php

class TestClass{

public $msg = “Hello world”;

}</code></pre>

<br>

<p>Now, let’s create an object of that class as below</p>

<pre><code>$test = new TestClass();</code></pre>

<br>

<p>What if we echo out this $test object of TestClass just like string? Let’s try it out</p>

<pre><code>echo $test;</code></pre>

<br>

<p>The complete code is:</p>

<pre><code>&lt;?php

class TestClass{

public $msg = “Hello world”;

}

$test = new TestClass();

echo $test;</code></pre>

<br>

<p>Let’s execute the code, BOOM!! It’s an error! The error message says: </p>

<pre><code>PHP Catchable fatal error: Object of class TestClass could not be converted to string in TestClass</code></pre>

<br>

<p>The error message has a very clear indication that we are using an instance of TestClass as String. This is a SIN! I mean a <strong>FATAL ERROR</strong></p>

<p>Can’t we do this in PHP? Can’t we use an object(or instance) of a Class as string?

The answer is Yes, we can. But the real question is what do we expect? What do we want to see when that $test object is used as String?</p>

<p>If we just want to see the content of $msg property of the object then we can do this using magic method \_\_toString as below: </p>

<pre><code>&lt;?php

class TestClass{

public $msg = “Hello world”;

public function \_\_toString(){

return $this->msg;

}

}</code></pre>

<br>

<p>What have we done here? We have added the magic method \_\_toString() in our TestClass. That method return the content of $msg. We could have returned what we want as string from the \_\_toString() method. I mean, it’s not mandatory to return a property of the object. </p>

<p>Let’s test it again our modified TestClass now. </p>

<pre><code>$test = new TestClass();

echo $test;</code></pre>

<br>

<p>The complete code is:</p>

<pre><code>&lt;?php

class TestClass{

public $msg = “Hello world”;

public function \_\_toString(){

return $this->msg;

}

}

$test = new TestClass();

echo $test;</code></pre>

<br>

<p>Let’s run it, YES!! The output is<p>

<pre><code>Hello world</code></pre>

<br>

<strong>This is the magic of magic method \_\_toString()</strong>