

PHP 5.4 New Features

Haim Michael

October 14th, 2012

All logos, trade marks and brand names used in this presentation belong to the respective owners.

Table of Content

- Support for Traits
- Arrays Short Syntax
- Function Array Dereferencing
- PHP Short Tags
- Instantiation Member Access
- Binary Number Format
- Session Upload Process
- Development Web Server

Support for Traits

- PHP 5.4 we can define Traits. Defining a trait is very similar to defining a class. Instead of using the keyword `class` we use the keyword `trait`.
- The purpose of traits is to group functionality in a fine grained and consistent way.
- It is not possible to instantiate a trait. The trait serves as an additional capability when using inheritance in our code.

Support for Traits

```
<?php
trait Academic
{
    function think()
    {
        echo "i m thinking!";
    }
}
```

Support for Traits

```
class Person
{
    private $id;
    private $name;
    function __construct($idValue,$nameValue)
    {
        $this->id = $idValue;
        $this->name = $nameValue;
    }
    function __toString()
    {
        return "id=".$this->id." name=".$this->name;
    }
}
```

Support for Traits

```
class Student extends Person
{
    use Academic;
    private $avg;
    function __construct($idVal,$nameVal,$avgVal)
    {
        parent::__construct($idVal,$nameVal);
        $this->avg = $avgVal;
    }
    function __toString()
    {
        $str = parent::__toString();
        return "avg=".$this->avg.$str;
    }
}
```

Support for Traits

```
class Lecturer extends Person
{
    use Academic;
    private $degree;
    function __construct($idVal,$nameVal,$degreeVal)
    {
        parent::__construct($idVal,$nameVal);
        $this->degree = $degreeVal;
    }
    function __toString()
    {
        $str = parent::__toString();
        return "degree=".$this->degree.$str;
    }
}
```

Support for Traits

```
$student = new Student(123123,"mosh",98);  
$lecturer = new Lecturer(42343,"dan","mba");  
  
$student->think();  
echo "<hr/>";  
$lecturer->think();  
  
?>
```


Support for Traits

- Methods of the current class override methods we inserted using the trait.
- Methods inserted by a trait override methods inherited from a base class.
- We can insert multiple traits into one class by listing them in the use statement separated by commas.
- If two traits (or more) insert two (or more) methods with the same name then a fatal error is produced.

Support for Traits

- We can use the `insteadof` operator in order to choose the exact method we want to use.
- We can use the `as` operator in order to include a conflicting method under another name.

Support for Traits

```
<?php
trait Player
{
    function play()
    {
        echo "<h1>whoo-a</h1>";
    }
    function printdetails()
    {
        echo "<h1>player...</h1>";
    }
}
trait Gamer
{
    function play()
    {
        echo "<h1>shooooo</h1>";
    }
    function printdetails()
    {
        echo "<h1>gamer...</h1>";
    }
}
```

Support for Traits

```
class Person
{
    use Gamer, Player
    {
        Gamer::printdetails insteadof Player;
        Player::play insteadof Gamer;
        Gamer::play as xplay;
    }
}

$obj = new Person();
$obj->xplay();
$obj->play();
$obj->printdetails();

?>
```

Support for Traits

- We can define a trait composed of others. Doing so we can put together separated traits into one.

```
trait Gamer
{
    function play()
    {
        echo "play...";
    }
}

trait Gambler
{
    function gamble()
    {
        echo "gamble...";
    }
}
```

Support for Traits

```
trait GamblingGamer
{
    use Gambler, Gamer;
}

class User
{
    use GamblingGamer;
}

$obj = new User();
$obj->gamble();
$obj->play();
?>
```

Support for Traits

- We can define a trait that includes the definition for abstract methods. Doing so, we can use that trait to impose requirements upon the classes that use it.
- It is possible to define our trait with properties. When instantiating a class that uses our trait we will be able to refer those properties in the new created object. If the class that uses our trait includes the definition for a property with the same name we will get an error.

Arrays Short Syntax

- As of PHP 5.4 we can create new arrays in the following new short syntax:

```
<?php
$vec_b = ['a'=>'australia','b'=>'belgium','c'=>'canada'];
foreach($vec_b as $k=>$v)
{
    echo " " . $k . "=>" . $v;
}
?>
```


Function Array Dereferencing

- As of PHP 5.4 we can develop a function that returns an array and use a call to that function as if it was an array.

```
<?php
function countries()
{
    $vec = ["italy","france","israel"];
    return $vec;
}

echo countries()[0];
?>
```

PHP Short Tags

- PHP 5.4 supports the `<?...?>` and `<?= expression ?>` short tags by default. We don't need to introduce any change in `php.ini` in order to use them.

```
<?
$numA = 24;
$numB = 4;
?>
```

```
<h1><?= ($numA+$numB) ?></h1>
```

Instantiation Member Access

- PHP 5.4 allows us to access class members on the object instantiation. It is useful in those cases when we need to access a single member of an object and don't need the object.

```
<?
class Utils
{
    function calc($numA,$numB)
    {
        return $numA+$numB;
    }
}

$temp = (new Utils())->calc(3,4);
echo $temp;
?>
```

Binary Number Format

- PHP 5.4 allows us to write binary numbers. We just need to precede our number with 0b.

```
<?
$a = 0b101;
$b = 0b110;
$c = $a + $b;
echo $c; // 11
?>
```

Session Upload Process

- PHP 5.4 allows us to get detailed information about the files that are currently been uploaded.

```
<?php
$key = ini_get("session.upload_progress.prefix").
        $_POST[ini_get("session.upload_progress.name")];
var_dump($_SESSION[$key]);
?>
```

The form might be looking like this.

```
<form action="upload.php" method="POST" enctype="multipart/form-data">
  <input type="hidden"
    name="<?php echo ini_get("session.upload_progress.name"); ?>"
    value="123" />
  <input type="file" name="file1" />
  <input type="file" name="file2" />
  <input type="file" name="file3" />
  <input type="submit" />
</form>
```

Development Web Server

- PHP 5.4 includes a built-in web server. This server might assist during the development phase.

Questions & Answers

- Two courses you might find interesting include

PHP Cross Platform Mobile Applications

[more info](#)

.NET Cloud Based Web Applications

[more info](#)

Android 4.1 Applications Development

[more info](#)

- If you enjoyed my lecture please leave me a comment at <http://speakermix.com/life-michael>.

Thanks for your time!

Haim.