eZ systems

Share your information

Welcome!

Migrating from PHP 4 to PHP 5

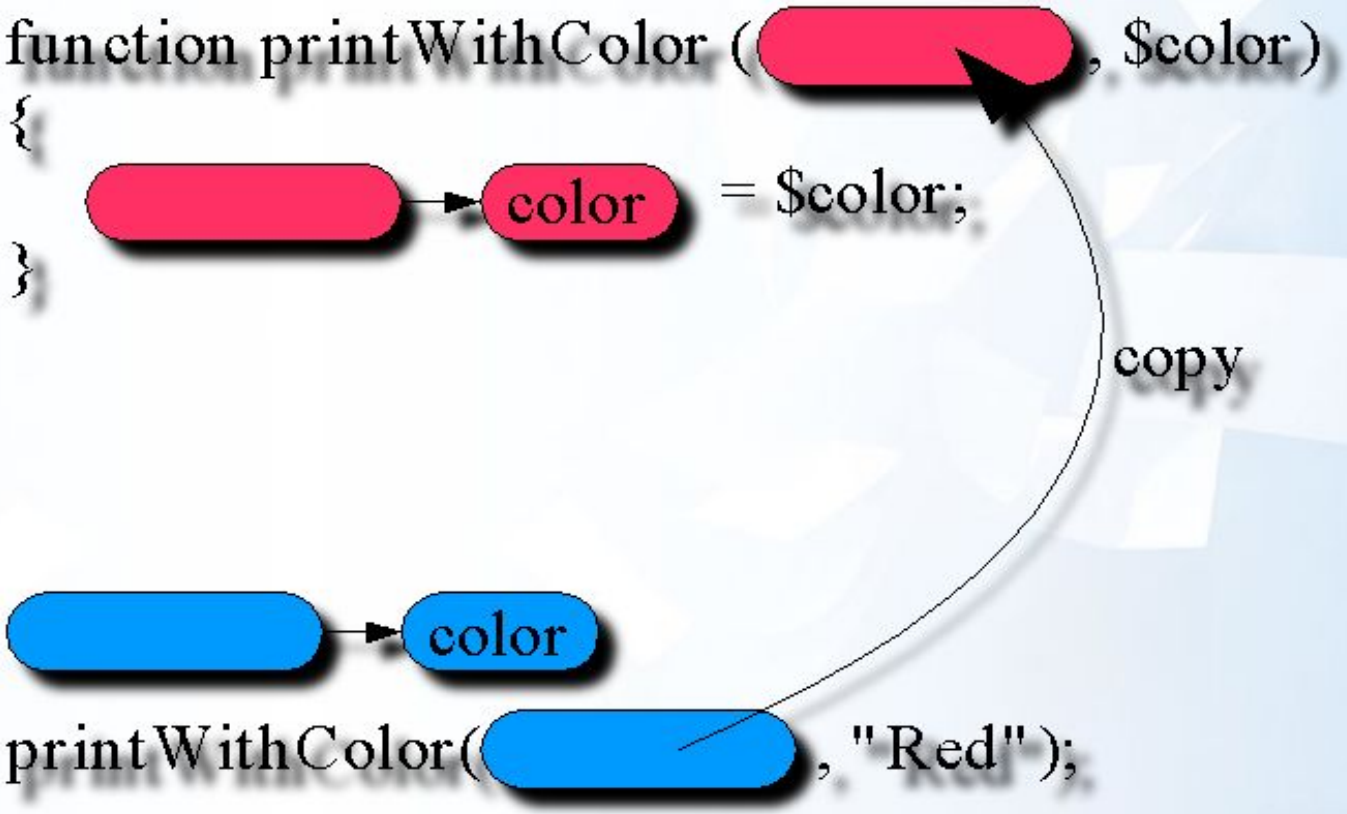
Intl. PHP Conference

Nov 9, 2004. Frankfurt, Germany

Derick Rethans <derick@php.net>

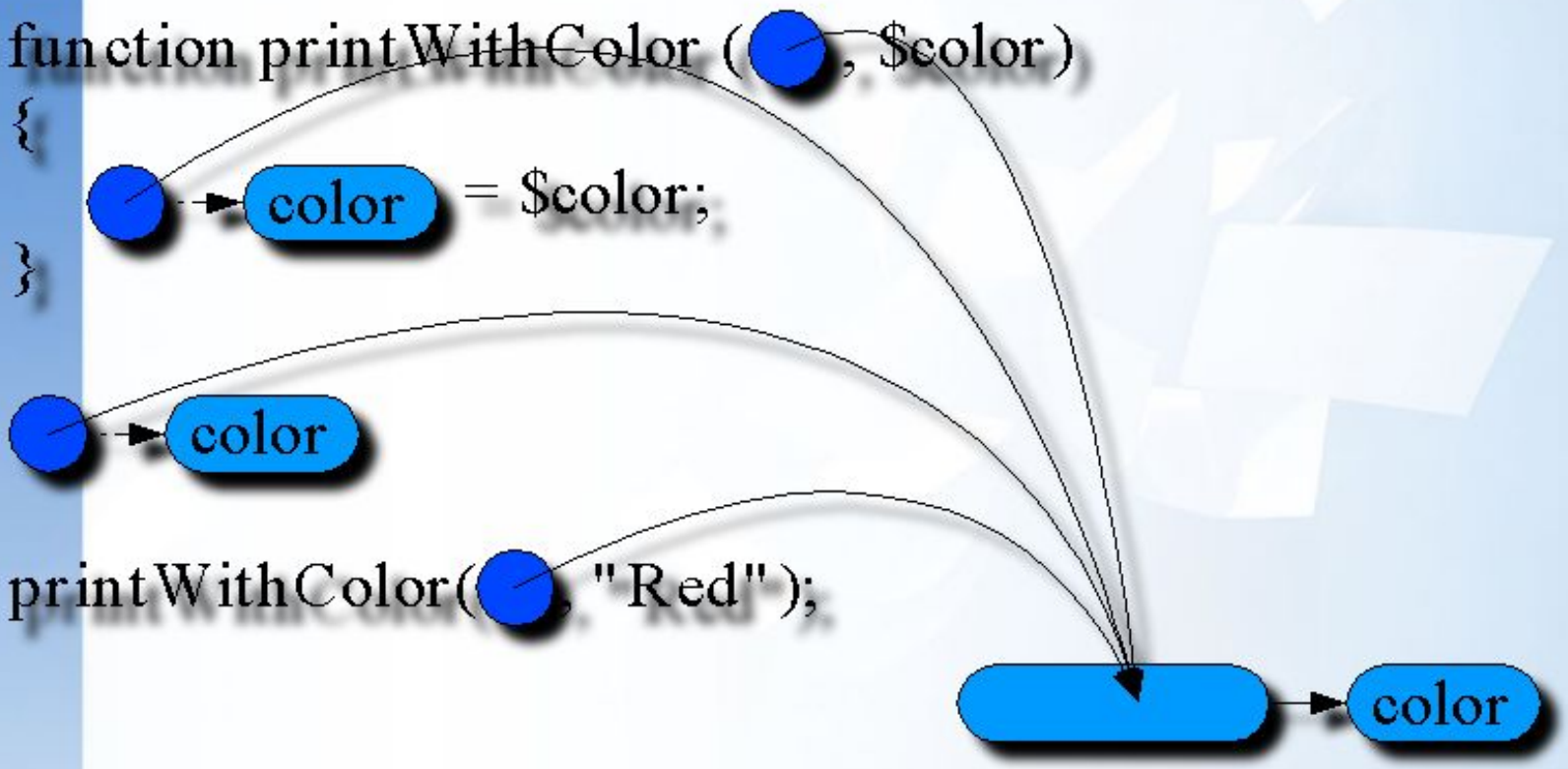
http://pres.derickrethans.nl/migrating-ffm2004

Objectmodel in PHP 4



- Pass by value
- Use of references needed almost everywhere...
- at call time, and even during instantiation.

Objectmodel in PHP 5



- Pass by reference
- Objects are referenced by a handle.

eZ systems

Clone

The PHP 5 Way

Share your information

clone keyword:

```php
<?php
    class OS {
        var $name;

        function OS($name) {
            $this->name = $name;
        }
    }

    function changeName($obj, $name) {
        $obj->name = $name;
    }

    $linux = new OS('linux');
    $win = clone $linux;
    changeName($win, 'windows');
    echo $linux->name. "\n";
    echo $win->name;

?>
```

output:

```
linux
windows
```

eZ systems

Clone

The Compatible Way

zend.ze1_compatibility_mode:

```php
<?php
    ini_set('zend.ze1_compatibility_mode', '1');

    class OS {
        var $name;

        function OS($name) {
            $this->name = $name;
        }
    }

    function changeName(&$obj, $name) {
        $obj->name = $name;
    }

    $linux = new OS('linux');
    $win = $linux;
    changeName($win, 'windows');
    echo $linux->name. "\n";
    echo $win->name;

?>
```

output:

```
linux
windows
```

Share your information

eZ systems

Casting Objects

PHP 4

```php
<?php
    class Country {
        function set_name($name) {
            $this->name = $name;
        }
    }

    $jurop = new Country;
    $dutchieland = new Country;
    $dutchieland->set_name('The Netherlands');

    echo (int) $jurop, "\n";
    echo (int) $dutchieland, "\n";
?>
```

output:

```
0
1
```

- In PHP 4 **(int) $object** results in:
- "0" when there are no properties set
- "1" when there are properties set

Share your information

eZ systems

Casting Objects

PHP 5

```php
<?php
    class Country {
        function set_name($name) {
            $this->name = $name;
        }
    }

    $jurop = new Country;
    $dutchieland = new Country;
    $dutchieland->set_name('The Netherlands');

    echo (int) $jurop, "\n";
    echo (int) $dutchieland, "\n";
?>
```

output:

```
Object id #1
Object id #2
```

In PHP 5 **(int) $object** results in:

- "Object id #" + Object ID
- unless **zend.ze1_compatibility_mode = 1**

Share your information

Comparing Objects

PHP 4

```php
<?php
    class Country {
        var $area;
        function Country($area) {
            $this->area = $area;
        }
    }

    $deutschland = new Country('West Europe');
    $die_schweiz = new Country('West Europe');
    $norway      = new Country('Scandinavia');

    if ($deutschland == $die_schweiz) {
        echo "Deutschland ist die Schweiz.\n";
    }
    if ($deutschland === $die_schweiz) {
        echo "Deutschland ist die Schweiz.\n";
    }
    if ($deutschland == $norway) {
        echo "Deutschland er Norway.\n";
    }
?>
```

In PHP 4 **$obj1 == $obj2** or **$obj1 === $obj2** results in **true**:

- when the object's properties are the same

Comparing Objects

PHP 5

```php
<?php
    class Country {
        var $area;
        function Country($area) {
            $this->area = $area;
        }
    }

    $deutschland = new Country('West Europe');
    $die_schweiz = new Country('West Europe');

    if ($deutschland == $die_schweiz) {
        echo "Deutschland ist die Schweiz.\n";
    }
    if ($deutschland === $die_schweiz) {
        echo "Deutschland ist die Schweiz.\n";
    }
?>
```

In PHP 5 **$obj1 == $obj2** results in **true:**

- when the object's properties are the same

In PHP 5 **$obj1 === $obj2** results in **true:**

- when the object's **handles** are the same

Compatibility Mode

php.ini:

```
zend.ze1_compatibility_mode = 1
```

Affects:

- cloning objects
- casting objects
- comparing objects

E_STRICT

E_STRICT is a new error level.

- Not part of E_ALL
- Numerical value is 2048 (and E_ALL is still 2047)
- To see all errors: E_ALL | E_STRICT

Affects:

- Automagically creating objects
- var / public modifiers
- Constructor naming

Assigning to $this

PEAR uses this for two things mainly:

1. returning errors from constructors (**$this = PEAR_Error();**)
2. driver model (**$this = new Slider('Pager');**)

This is no longer allowed in PHP 5, so the classes should be redesigned.

Solutions (not compatible with PHP 4).

1. Use Exceptions
2. driver model (**$this = new Slider('Pager');**)

constructors

```php
<?php
    class Weather {
        var $temperature;

        function __construct($temp)
        {
            echo "New version\n";
        }

        function Weather($temp)
        {
            echo "Old version\n";
        }
    }

    $w = new Weather(7);
?>
```

PHP 5:

```
Strict Standards: var: Deprecated. Please use the public/private/protected
modifiers in - on line 3

Strict Standards: Redefining already defined constructor for class Weather
in - on line 9

New version
```

get_class()

```php
<?php
    class TheNetherlands {
        var $area;
        function TheNetherlands($area) {
            $this->area = $area;
        }
    }

    $holland = new TheNetherlands('Holland');

    echo get_class($holland), "\n";
?>
```

PHP 4:

```
thenetherlands
```

PHP 5:

```
TheNetherlands
```

The same is true for get_parent_class() and get_class_methods().

var vs. public

```php
<?php
    class Russia {
        var $local_name;

        function Russia() {
            $this->local_name = "Roosija";
        }
    }
?>
```

PHP 5:

```
Strict Standards: var: Deprecated. Please use the public/private/protected
modifiers in - on line 3
```

Automagic Objects

```php
<?php
    $country->name = 'Moldovia';

    var_dump($country);
?>
```

PHP 5:

```
Strict Standards: Creating default object from empty value in - on line 2
object(stdClass)#1 (1) {
    ["name"]=>
    string(8) "Moldovia"
}
```

Signatures

```php
<?php
    class Scandinavia {
        function get_countries() {
            return array('Norway', 'Sweden', 'Denmark', 'Finland');
        }
    }

    class Europe extends Scandinavia {
        function get_countries($area) {
            if ($area == "Scandinavia") {
                return Scandinavia::get_countries();
            }
        }
    }
?>
```

PHP 5:

```
Strict Standards: Declaration of Europe::get_countries() should be
compatible with that of Scandinavia::get_countries() in - on line 14
```

Define classes before using them

In PHP 4 it was never necessary to define a class before using it in your code, although it would decrease script execution times a bit. The same is true for PHP 5, but if you are using Interfaces in your code, then you are **required** to define a class before using it.

array_merge()

array_merge() no longer accepts non-array parameters to merge.

```php
<?php
    $scandinavia = array('Denmark', 'Norway', 'Sweden', 'Finland');
    $uk = "The UK";

    $europe = array_merge($scandinavia, $uk);

    print_r($europe);
?>
```

PHP 4:

```
Array
(
    [0] => Denmark
    [1] => Norway
    [2] => Sweden
    [3] => Finland
    [4] => The UK
)
```

PHP 5:

```
Warning: array_merge(): Argument #2 is not an array in - on line 7
```

strrpos() and strripos()

strr(i)pos() now searches the full needle inside the haystack, and not only the first character of it.

```php
<?php
    $haystack = "There are now ten more countries in Europe";
    $needle   = "The Netherlands";

    $pos = strrpos($haystack, $needle);
    var_dump($pos);
?>
```

PHP 4:

```
int(0)
```

PHP 5:

```
bool(false)
```

ip2long()

```php
<?php
    $ips = array("129.51.31.51", "129.256.13.15", "255.255.255.255");
    foreach ($ips as $ip) {
        $v = ip2long($ip);
        printf("%s %u\n", var_export($v, true), $v);
    }
?>
```

PHP 4:

```
-2127356109 2167611187
-1 4294967295
-1 4294967295
```

PHP 5:

```
-2127356109 2167611187
false 0
-1 4294967295
```

Command Line Interface

In the PHP 4 distributions you will find the following binaries:

- php.exe (The CGI binary to use with a web server)
- **cli/**php.exe (The CLI binary to use with command line scripts)

In the PHP 5 distributions you will find the following binaries:

- php**-cgi**.exe (The CGI binary to use with a web server)
- php.exe (The CLI binary to use with command line scripts)

Changes in MySQL

In PHP 5 the MySQL client library is no longer bundled because:

- Most systems have the library installed
- Multiple versions of libraries can cause problems (mod_auth_mysql)
- Maintainability of the bundled library
- libmysql changed license to GPL, which means we can not bundle it with PHP anymore.

Solution:
Install the MySQL client libraries and add `--with-mysql=[DIR]` to ./configure

eZ systems

Share your information

Questions

Resources

These Slides: http://pres.derickrethans.nl

PHP: http://www.php.net

PHP 4 to 5 Migration manual page: http://php.net/migration5

Questions?: derick@php.net