

Ausarbeitung Lerngruppe 12/12/2013

Mir ist aufgefallen, dass ich vergessen hab euch noch neben dem Objekt Operator -> auch den Gültigkeitsbereichsoperator :: zu erklären. Deshalb hab ich mir die Freiheit genommen und euch noch zusätzlich eine kurze Erläuterung zu dem Thema von der PHP Documentation eingefügt.

Inhaltsverzeichnis

Objekt Operator (->).....	2
Gültigkeitsbereichsoperator (::)	3
Beispiel #1 :: außerhalb der Klassendefinition	3
Beispiel #2 :: innerhalb der Klassendefinition	3
Beispiel #3 Eine Vatermethode aufrufen.....	4
PHPDOC	4

Objekt Operator (->)

Der Pfeil(->) Operator ist einfach ausgedrückt ein Weg um in PHP Variablen einen Wert zuzuweisen, auf Variablen zuzugreifen und Methoden auszuführen.

Anhand des Beispiels aus der Stunde eine Erklärung des Operators.

```
class MyClass {  
    public $myVar;  
  
    public function myMethod() { }  
}
```

```
$instance = new MyClass();
```

```
$instance->myVar = "Hello World"; // Weist myVar den Wert "Hello World" zu.  
$instance->myMethod(); // Führt "myMethod()" aus
```

Als erstes haben wir uns eine kleine Klasse MyClass erstellt, welche eine Variable \$myVar und eine Funktion myMethod() besitzt.

Als nächstes haben wir uns dann ein Objekt der Klasse MyClass erstellt um später dann damit zu arbeiten.

Mittels des Operators -> greifen wir nun auf die Variable \$myVar zu und weisen ihr gleich einen Wert zu. Des Weiteren führen wir die Methode myMethod() aus.

Man kann diesen Operator auch in Strings anwenden, wie das folgende Beispiel zeigt.

```
$s = "$this -> a"; // Gibt den Wert von a aus.
```

```
$s = "$this -> method()"; // Aufgrund des -> nimmt PHP an, dass es sich um eine Variable method handelt. Da  
    // diese jedoch noch nicht existiert wird sie automatisch erzeugt besitzt aber noch  
    // keinen spezifischen Wert. Dadurch, wenn man $s nun ausgeben will bekommt  
    // man nur ()
```

Gültigkeitsbereichsoperator (::)

Der Gültigkeitsbereichsoperator ein Kürzel, das Zugriff auf statische, konstante und überschriebene Member oder Methoden einer Klasse erlaubt.

Wenn Sie diese Elemente außerhalb der Klassendefinition ansprechen, benutzen Sie den Namen der Klasse.

Beginnend mit PHP 5.3.0 ist es möglich eine Variable als Klassenreferenz zu nutzen. Der Variablenwert kann kein Schlüsselwort (wie *self*, *parent* oder *static*) sein.

Beispiel #1 :: außerhalb der Klassendefinition

```
<?php
class MyClass {
    const CONST_VALUE = 'Ein konstanter Wert';
}

$classname = 'MyClass';
echo $classname::CONST_VALUE; // Ab PHP 5.3.0

echo MyClass::CONST_VALUE;
?>
```

Die drei speziellen Schlüsselwörter *self*, *parent* und *static* werden benutzt, um auf Member und Methoden von innerhalb der Klassendefinition zuzugreifen.

Beispiel #2 :: innerhalb der Klassendefinition

```
<?php
class OtherClass extends MyClass
{
    public static $my_static = 'statische var';

    public static function doubleColon() {
        echo parent::CONST_VALUE . "\n";
        echo self::$my_static . "\n";
    }
}

$classname = 'OtherClass';
echo $classname::doubleColon(); // Ab PHP 5.3.0

OtherClass::doubleColon();
?>
```

Wenn eine abgeleitete Klasse die Definition der Methode eines Vaters überschreibt, wird PHP die Methode des Vaters nicht aufrufen. Es obliegt der abgeleiteten Klasse, ob die Methode der Vaterklasse aufgerufen wird oder nicht. Dies gilt ebenfalls für Konstruktoren, Überladung und magische Methodendefinitionen.

Beispiel #3 Eine Vatermethode aufrufen

```
<?php
class MyClass
{
    protected function myFunc() {
        echo "MyClass::myFunc()\n";
    }
}

class OtherClass extends MyClass
{
    // Die Definition des Vaters überschreiben
    public function myFunc()
    {
        // Aber dennoch die Funktion des Vaters aufrufen
        parent::myFunc();
        echo "OtherClass::myFunc()\n";
    }
}

$class=new OtherClass();
$class->myFunc();
?>
```

Falls ihr noch Fragen zu dem Thema habt, könnt ihr diese innerhalb unserer Lerngruppe, aber auch außerhalb dieser Zeit gerne an mich richten ☺.

Auflistung und Verwendung der Operatoren in PHP:

http://www.w3schools.com/php/php_operators.asp

PHPDOC

Hier habt ihr ein paar Links, welche die Verwendung von PHPDocs noch einmal sehr gut erläutern und weiters noch zusätzlich die Installation von **phpDocumentator** erklären.

Verwendung PHPDoc und Erklärung der Installation von **phpDocumentator** mittels pear:

<http://professionelle-softwareentwicklung-mit-php5.de/tools.code-documentation.phpdocumentor.html>

phpDocumentator ausführen in Eclipse als externes Tool:

<http://www.speich.net/articles/2008/07/25/using-categories-in-own-nav/>

Und falls noch immer was unklar ist zum Thema PHPDoc hier nochmal der Link von eLearning:

<http://www.phpdoc.de/kongress/phpdoc.html>