

Object-oriented programming in PHP

Objective

For this tutorial, I'll explain the basics of PHP object-oriented programming. Examples in this tutorial will use the *PHP Simple HTML DOM Parser*. By the end of this tutorial you should be able to write object-oriented code in PHP and use PHP to parse HTML files.

PHP oop

For people familiar with java, C#, or other object-oriented languages, this code will be very intuitive.

```
<?php

class ExplainOOP{

    private $i_priMemberVar;
    public $s_pubMemberVar;

    public function __construct(){
        $this->i_priMemberVar = 100;
        $this->s_pubMemberVar = "public string";
    }

    public function incPriMemberVar(){
        $this->i_priMemberVar++;
    }

    public function getPriMemberVar(){
        return $this->i_priMemberVar;
    }

}

?>
```

In the *ExplainOOP* class we see that there are two member variables, two member functions, and one constructor. *\$i_priMemberVar* is private and can only be accessed, but methods within the class. Typically, you have setters and getters which are public functions that access private variables. The methods *incPriMemberVar* and *getPriMemberVar* are examples of such.

When using variables within your class, it's always good practice to use *this*. *this* means the variable or function from this class. It always represents the member variables or functions for the class you're using.

Object-oriented programming in PHP

For example.

```
$o_myObject = new ExplainOOP();  
$o_myObject->incPriMemberVar();  
echo '$i_priMemberVar: ' . $o_myObject->getPriMemberVar();
```

This code will display on our webpage.

```
$i_priMemberVar: 101
```

The first line in this code creates a new object *\$o_myObject*, and when this object is made the *__construct* is called setting *\$i_priMemberVar* to 100. Calling the method *incPriMemberVar* allows you to add one this number, and finally this value is displayed on our webpage as 101.

Similarly, we can access and change *\$s_pubMemberVar*.

```
$o_myObject->s_pubMemberVar .= " here";  
echo '<br>' . $o_myObject->s_pubMemberVar;
```

Output

```
public string here
```

But since *\$s_pubMemberVar* is a public variable, we can modify and use it like any other variable. This at times is useful, but in most cases we want to stop other programmer from messing up our classes member variables. For instance, if you had a class that calculated values for a mathematical equation and stored these in an array. It wouldn't be useful for other programmers if the array was public, in fact it could be harmful.

Let's say this programmer (thinking herself smart) modifies your public array, this could make other methods in your class fail. To prevent these annoying programmers from messing up your beautifully written code make the array private and create a getter method.

Object-oriented programming in PHP

PHP Simple HTML DOM Parser

Now go and download the latest version of the *HDOM parser* from:

<http://sourceforge.net/projects/simplehtmldom/>

And place this in directory you have your PHP code in. We'll be parsing a webpage in these examples; it has a funny definition of the word *Java*.

This code will extract the title and definition from *java.html*.

```
<?php

include 'simple_html_dom.php';

$HDOM_parserObject = new simple_html_dom();
$HDOM_parserObject->load_file("./java.html");

$HDOM_elementTitle = $HDOM_parserObject->find('title',0);

echo '<br> title: ' . $HDOM_elementTitle->plaintext;

foreach($HDOM_parserObject->find('p') as $HDOM_elementP)
    echo $HDOM_elementP->outertext;

?>
```

Now that we are familiar with OOP in PHP, we can recognize a few things in this code. We can see that on the second line, we are creating a new *simple_html_dom* object. And that on the third line we're setting the file we want to parse. We can imagine that the *load_file* method is setting some private variable and running some other private method to gather the data contained in the html file.

On lines, four and six there's the find method is used. This method will search the html file for a specific tag and return an element object. This element object can be manipulated with *magic attributes*.

A list of the attributes from the *HDOM parser* documentation.

```
$html = str_get_html("<div>foo <b>bar</b></div>");
$e = $html->find("div", 0);

echo $e->tag; // Returns: " div"
echo $e->outertext; // Returns: " <div>foo <b>bar</b></div>"
echo $e->innertext; // Returns: " foo <b>bar</b>"
echo $e->plaintext; // Returns: " foo bar"
```

Looking back at line four we see that it search for a *title* tag. The zero represents the index of the *title* tag because there's only one in this html file, it only makes sense to return the zeroth index. Otherwise, it would return an element array only containing one element.

Object-oriented programming in PHP

Finally on line six, we use a foreach loop to go through the each element in the *p* tag array. Printing the *outtext* of these elements will retain the *p* tags keeping the text in its original state.

Output

title: Java

An object-oriented language originally developed at Sun by James Gosling (and known by the name “Oak”) with the intention of being the successor to [C++](#) (the project was however originally sold to Sun as an embedded language for use in set-top boxes). After the great Internet explosion of 1993-1994, Java was hacked into a byte-interpreted language and became the focus of a relentless hype campaign by Sun, which touted it as the new language of choice for distributed applications.

Java is indeed a stronger and cleaner design than C++ and has been embraced by many in the hacker community – but it has been a considerable source of frustration to many others, for reasons ranging from uneven support on different Web browser platforms, performance issues, and some notorious deficiencies in some of the standard toolkits (AWT in particular). [Microsoft](#)'s determined attempts to corrupt the language (which it rightly sees as a threat to its OS monopoly) have not helped. As of 2003, these issues are still in the process of being resolved.

Despite many attractive features and a good design, it is difficult to find people willing to praise Java who have tried to implement a complex, real-world system with it (but to be fair it is early days yet, and no other language has ever been forced to spend its childhood under the limelight the way Java has). On the other hand, Java has already been a big [win](#) in academic circles, where it has taken the place of [Pascal](#) as the preferred tool for teaching the basics of good programming to the next generation of hackers.

Object-oriented programming in PHP

Complete code

```
<?php
    class ExplainOOP{

        private $i_priMemberVar;
        public $s_pubMemberVar;

        public function __construct(){
            $this->i_priMemberVar = 100;
            $this->s_pubMemberVar = "public string";
        }

        public function incPriMemberVar(){
            $this->i_priMemberVar++;
        }

        public function getPriMemberVar(){
            return $this->i_priMemberVar;
        }
    }

    $o_myObject = new ExplainOOP();
    $o_myObject->incPriMemberVar();

    echo '<br> $i_priMemberVar: ' . $o_myObject->getPriMemberVar();

    $o_myObject->s_pubMemberVar .= " here";

    echo '<br>' . $o_myObject->s_pubMemberVar;
?>

<?php
    include 'simple_html_dom.php';

    $HDOM_parserObject = new simple_html_dom();
    $HDOM_parserObject->load_file("./java.html");

    $HDOM_elementTitle = $HDOM_parserObject->find('title',0);

    echo '<br> title: ' . $HDOM_elementTitle->plaintext;

    foreach($HDOM_parserObject->find('p') as $HDOM_elementP)
        echo $HDOM_elementP->outertext;
?>

<html>
<title>example</title>
<body></body>
</html>
```