

Forum:[Hack Forums](#)

Thread: [Intro into PHP OOP. From a beginner to a beginner.](#)

sdk - June 5, 2015, 10:46 pm

Object-Oriented Programming is a tad different. It contains Classes and basically structure to the code. Classes and functions are the main thing PHP brings.

Example:

PHP Code

```
<?php
//fnc.php
class Stats {
    //Called with no params
    function GetName() {
        //There are functions like this that are called with anything.
        //This we are calling to echo the username from the session. You could do it normally but it's
just an example.
        return $_SESSION['username'];
    }

    //Called with params
    function GetName($uid) {

//This functions grabs the username from a database. The $con is a database connection in the PD
O format.
        $Query = $con -> prepare("SELECT * FROM `users` WHERE `id` = :id");
        $Query -> execute(array(":id" => $uid));
        $row = $Query -> fetch(PDO::FETCH_ASSOC);
        return $row['username'];
    }
}
?>
```

Now Fusion, how do I call these??

You can call them by first declaring the class as a var.

Code:

PHP Code

```
<?php
require 'fnc.php';
$st = new Stats;

//No Param Call.
echo $st->GetName();

//Param Call

echo $st->GetName($_SESSION['uid']);
```

?>

I will be appending more to this soon :).

Any feedback is greatly appreciated. Also let me know what you would like to know :) I can try to show you!

Skits.

Tony S - June 6, 2015, 10:35 am

Your example doesn't give any insight as to why it's useful to use an object orientated design pattern instead of a (presumably) procedural one. In the example you posted, you're just overcomplicating everything for absolutely no reason. For example,

PHP Code

```
<?php
function GetName() {
    return $_SESSION['username'];
}
```

Really? That doesn't follow any OOP principals. You aren't returning a property of the object, so why wouldn't one just call `$_SESSION['username']` instead of instantiating the class and then calling the `GetName()` method?

I have a feeling that you don't understand the fundamental concepts of OOP yourself. Also, your coding style is horrendous, but that's being nit-picky.

0/5 from me

sdK - June 6, 2015, 11:25 am

(Jun 6, 2015 10:35 AM)Tony Spilotro Wrote: [\[CITE\]](#) Your example doesn't give any insight as to why it's useful to use an object orientated design pattern instead of a (presumably) procedural one. In the example you posted, you're just overcomplicating everything for absolutely no reason. For example,

PHP Code

```
<?php
function GetName() {
    return $_SESSION['username'];
}
```

Really? That doesn't follow any OOP principals. You aren't returning a property of the object, so why wouldn't one just call `$_SESSION['username']` instead of instantiating the class and then calling the `GetName()` method?

I have a feeling that you don't understand the fundamental concepts of OOP yourself. Also, your coding style is horrendous, but that's being nit-picky.

0/5 from me

If you read the whole thing, I said it was just an example of using functions and classes. At the bottom if you read. I said "I will add more tho this later." I was wanting to get something out fast so I just added a bit too it and gonna make it a megathread. Well shit dude sorry you feel the need to throw some bullshit down.

ManDoR - June 7, 2015, 5:15 am

(Jun 6, 2015 11:25 AM)Skits Vicious Wrote: [CITE](#) If you read the whole thing, I said it was just an example of using functions and classes. At the bottom if you read. I said "I will add more tho this later." I was wanting to get something out fast so I just added a bit too it and gonna make it a megathread. Well shit dude sorry you feel the need to throw some bullshit down. Actually, he's right. Please don't "make it a megathread".

Strider64 - June 7, 2015, 9:45 am

You be better off doing something like the following:

Page.php

PHP Code

<?php

// This script defines the Page class

/* Class Page.

* The class contains seven attributes, id, creatorId, sticky, title, content, dateAdded, and dateUpdated.

* The attributes match the corresponding database columns.

* The class contains eight methods:

* - getId()

* - getCreatorId()

* - isTopicSticky()

* - getTitle()

* - getContent()

* - getDateAdded()

* - getDateUpdated()

* - getIntro()

*/

class Page {

 // All attributes correspond to database columns.

 // All attributes are protected.

 protected \$id = null;

 protected \$creatorId = null;

 protected \$sticky = null;

 protected \$title = null;

 protected \$content = null;

 protected \$dateAdded = null;

 protected \$dateUpdated = null;

 // Six methods for returning attribute values:

 function getId() {

 return \$this->id;

 }

 function isTopicSticky() {

 return \$this->sticky;

 }

 function getCreatorId() {

 return \$this->creatorId;

 }

```

function getTitle() {
    return $this->title;
}

function getContent() {
    return $this->content;
}

function getDateAdded() {
    return $this->dateAdded;
}

function getDateUpdated() {
    return $this->dateUpdated;
}

// Method returns the first X characters from the content:
function getIntro($count = 501) {
    return substr(nl2br(strip_tags($this->content)), 0, $count) . '...';
}
}

```

then to use it:

PHP Code

```

<?php
try { // Fetch the page
    $query =
'Select id, creatorId, category, sticky, title, content, DATE_FORMAT(dateUpdated, "%M %e, %Y") as
dateUpdated FROM pages WHERE id=:id';
    $result = $pdo->prepare($query);
    $result->execute(array(':id' => $pageId));
    if ($result && $result->rowCount() > 0) { // Check that rows were returned:
        $result->setFetchMode(PDO::FETCH_CLASS, 'Page'); // Arm the class by fetching it:
        $page = $result->fetch();
    } else { // Problem!
        header('Location: index.php');
        exit();
    }
} catch (Exception $e) { // Catch generic exceptions
}

```

sdh - June 7, 2015, 10:47 am

(Jun 7, 2015 09:45 AM)Strider64 Wrote: [CITE](#) You be better off doing something like the following:

Page.php

PHP Code

```

<?php

```

```

// This script defines the Page class

```

```

/* Class Page.

```

* The class contains seven attributes, id, createId, sticky, title, content, dateAdded, and dateUpdated.

* The attributes match the corresponding database columns.

* The class contains eight methods:

* - getId()

* - getCreatorId()

* - isTopicSticky()

* - getTitle()

* - getContent()

* - getDateAdded()

* - getDateUpdated()

* - getIntro()

*/

class Page {

// All attributes correspond to database columns.

// All attributes are protected.

protected \$id = null;

protected \$creatorId = null;

protected \$sticky = null;

protected \$title = null;

protected \$content = null;

protected \$dateAdded = null;

protected \$dateUpdated = null;

// Six methods for returning attribute values:

function getId() {

return \$this->id;

}

function isTopicSticky() {

return \$this->sticky;

}

function getCreatorId() {

return \$this->creatorId;

}

function getTitle() {

return \$this->title;

}

function getContent() {

return \$this->content;

}

function getDateAdded() {

return \$this->dateAdded;

}

function getDateUpdated() {

return \$this->dateUpdated;

```

    }

    // Method returns the first X characters from the content:
    function getIntro($count = 501) {
        return substr(nl2br(strip_tags($this->content)), 0, $count) . '...';
    }
}

```

then to use it:

PHP Code

```

<?php
try { // Fetch the page
    $query =
'Select id, creatorId, category, sticky, title, content, DATE_FORMAT(dateUpdated, "%M %e, %Y") as dateUpdated FROM pages WHERE id=:id';
    $result = $pdo->prepare($query);
    $result->execute(array(':id' => $pageId));
    if ($result && $result->rowCount() > 0) { // Check that rows were returned:
        $result->setFetchMode(PDO::FETCH_CLASS, 'Page'); // Arm the class by fetching it:
        $page = $result->fetch();
    } else { // Problem!
        header('Location: index.php');
        exit();
    }
} catch (Exception $e) { // Catch generic exceptions
}

```

Thanks a bunch, like it says to a beginner from a beginner so this is just basics/what I am picking up doing my current project.

MineSQL - June 7, 2015, 1:27 pm

Your in no position to make a tutorial if your a beginner as well, it just doesn't make sense. I commend you for putting effort into it though. Perhaps I'll make a oop tutorial if any of you want one. I have a mini tutorial in my laravel thread, but it can be expanded on a lot more.

sdk - June 7, 2015, 1:44 pm

(Jun 7, 2015 01:27 PM)MineSQL Wrote: [\[CITE\]](#) Your in no position to make a tutorial if your a beginner as well, it just doesn't make sense. I commend you for putting effort into it though. Perhaps I'll make a oop tutorial if any of you want one. I have a mini tutorial in my laravel thread, but it can be expanded on a lot more.

Well most "experts" will not explain well enough Im not a PHP noob but I am a OOP PHP noob.

DogEatsDog - June 7, 2015, 1:50 pm

(Jun 7, 2015 01:44 PM)Skits Vicious Wrote: [\[CITE\]](#) Well most "experts" will not explain well enough Im not a PHP noob but I am a OOP PHP noob.

If you do not fully grasp a subject, don't teach others that subject. You are more likely to teach them bad habits.

sdk - June 7, 2015, 1:50 pm

(Jun 7, 2015 01:50 PM)DogEatsDog Wrote: [\[CITE\]](#) If you do not fully grasp a subject, don't teach others that subject. You are more likely to teach them bad habits.

I grasp it well. I am just learning other things at the moment. I know many other OOP languages im just learning PHPs methods and shit.

DogEatsDog - June 7, 2015, 1:55 pm

(Jun 7, 2015 01:50 PM)Skits Vicious Wrote: [\[CITE\]](#) I grasp it well. I am just learning other things at the moment. I know many other OOP languages im just learning PHPs methods and shit.

Then don't post incomplete stuff. Others might not know why things are like that if you don't explain it. And no, your "expansion" later on won't fix that for those that read it now.

MineSQL - June 7, 2015, 2:48 pm

(Jun 7, 2015 01:44 PM)Skits Vicious Wrote: [\[CITE\]](#) Well most "experts" will not explain well enough Im not a PHP noob but I am a OOP PHP noob.

I never claimed to be an expert, and I was trying to be as courtesy as possible in my wording. But, since your immature and take offence to my fair analysis of your tutorial, I will speak my mind.

Your code follows no practices and I doubt you know php as well as you say. These are the first posts I see you make in this section, and your 'tutorial' doesn't actually teach anything. If you run your 'code' in php as is, it won't do anything.

You can continue taking offence to my posts, and I will continue burying you into the ground, or you can move on and accept that you shouldn't be making a thread like this. The choice is yours.

sdk - June 7, 2015, 3:02 pm

(Jun 7, 2015 02:48 PM)MineSQL Wrote: [\[CITE\]](#) I never claimed to be an expert, and I was trying to be as courtesy as possible in my wording. But, since your immature and take offence to my fair analysis of your tutorial, I will speak my mind.

Your code follows no practices and I doubt you know php as well as you say. These are the first posts I see you make in this section, and your 'tutorial' doesn't actually teach anything. If you run your 'code' in php as is, it won't do anything.

You can continue taking offence to my posts, and I will continue burying you into the ground, or you can move on and accept that you shouldn't be making a thread like this. The choice is yours.

Never said you said you where an expert. I was generalizing. Good for you if you are smarter than me.

ASDA - June 7, 2015, 3:21 pm

If you don't understand the fundamentals of OOP I don't think you're in a place to be helping with Java or C# at all, never mind making a PHP tutorial on the topic.

My head is absolutely fucking whizzed when people say shit like "I am not noob at PHP but I am noob at OOP" fuck knows how you get anywhere in PHP without knowing the object oriented side of it all. I don't know how you can say you're "not a noob at PHP" when you don't know the basics of the language. You can't find one job listing for PHP without the asking for knowledge with object-oriented programming, that to me puts you back in the "noob" category as you put it

sdk - June 7, 2015, 3:29 pm

(Jun 7, 2015 03:21 PM)ASDA Wrote: [\[CITE\]](#) If you don't understand the fundamentals of OOP I don't think you're in a place to be helping with Java or C# at all, never mind making a PHP tutorial on the topic.

My head is absolutely fucking whizzed when people say shit like "I am not noob at PHP but I am noob at OOP" fuck knows how you get anywhere in PHP without knowing the object oriented side of

it all. I don't know how you can say you're "not a noob at PHP" when you don't know the basics of the language. You can't find one job listing for PHP without the asking for knowledge with object-oriented programming, that to me puts you back in the "noob" category as you put it Omfg. Please read. I know OOP just learning PHPs methods thats all. Since fuckboys can't read closing cause of idiots.