

Forum:[Hack Forums](#)

Thread: [An Introduction to OOP \(Object-Oriented-Programming\) In PHP](#)

GaBaNo - November 24, 2012, 11:41 am

Let me explain, before I start, that this might be a little off. It's mostly community and feedback created, so yeah. I don't use OOP a lot, and am kind of new to it. The original tutorial was shit, but I've added a lot to it from feedback and articles. Credits are listed at the bottom of the page.

OOP is mainly used for variable reusage and, as many people say, relates to real life situations so it's easier to understand.

PHP is a multiple-paradigm language, meaning you can write it in either procedural or object orientated. The language itself is not fully object orientated, and you cannot make it so.

So, in OOP, we usually have two files: the class, and then the handler. Let's get started with our class:

```
class.php
PHP Code
<?php
class OOP {
public $variable;
}
?>
```

As you can see, the class name does not have to be the same as the file name (but it is highly recommended to be so), like how it is in java and many other programming languages. Also, notice how we defined our variable as a public status. This is because if you do not put anything before the variable, it gives you an error. Please note that there are a bunch of things you can put in front of your variable - it all really depends to be honest. They are called modifiers and modify how the variable is used. For some of the modifiers, please look at the bottom of the page. You may have also noticed that our variable is undefined. It can be defined or kept this way, it does not matter.

Now, we need to define our variables. Well, if we did not define them straight away, then how do we do this? It's simple, and here's how:

```
PHP Code
<?php
class OOP {
public $variable;

function __construct() {
$this->variable = "Hello";
}

}
?>
```

As you can see, we defined it using the `$this->` method. Now, let me explain what `$this->` means:

In any programming language, "this" references to the class you are working in. If you attempt to do

a "this" in a regular PHP file with no class, you will get an error as there is no object/class to reference the variable to. Also, we defined it in a function and not just out in the open. Why? Because we would have gotten a syntax error (explained below).

* QUICK STOP NOTE *Syntax Error: You would have received the "EXPECTING T_FUNCTION" error when using any function besides your own in a class

public function __construct {} - This is a constructor (listed at the bottom of the page). A constructor is called whenever a object initializes as a result from a class.

Now that we have defined our variable, how would we display it on the page? Well, we certainly wouldn't go straight out and echo it through the class. You can do this, but you would need a whole function, and then you would need to display the function. This isn't as neat, and personally, I like to have a handler:

handler.php

PHP Code

```
<?php
```

```
$OOP = new OOP();
```

```
?>Note: Your object should be named after your class. Read the next paragraph for information on objects!
```

Now, as you can see, we are creating a new object named \$OOP (objects are the result of initiating a class) and running it through our handler. If you run this, however, you will get an error. For all of you Java Guru's, this is not like Java anymore, and let me explain:

In PHP, since classes do not have to be named after the file name, you must include the class first. In Java, however, all classes must be named after their filename. This is why in java you do not need to include the file, but only define it.

So, to include our file (for those of you who don't know this), we would do this:

PHP Code

```
<?php
```

```
require_once('class.php'); # once to prevent errors
```

```
$OOP = new OOP();
```

```
?>
```

Now it can fetch the class correctly. :)

So, to echo our variable, we would then do this:

PHP Code

```
<?php
```

```
echo $OOP->variable; # Calls the variable $variable from the object $OOP (class: OOP)
```

And Voila, you are done!

Notes/ReferencesYou cannot echo or do anything of the sort in a class, you can only define variables and functions

OOP Can be very confusing to some people, so if you do not get this, I advise you to go read up on it more

DefinitionsClass: This is a programmer-defined datatype, which includes local functions as well as local data. You can think of a class as a template for making many instances of the same kind (or

class) of object.

Object: An individual instance of the data structure defined by a class. You define a class once and then make many objects that belong to it. Objects are also known as instance.

Constructor: Refers to a special type of function which will be called automatically whenever there is an object formation from a class.

ModifiersPublic: Accessible by everything and anything

Protected: Accessible only in the same class or classes that extend the class it is in

Private: Accessible ONLY in the class it is declared in

Credits

- http://www.tutorialspoint.com/php/php_object_oriented.htm - A good-looking tutorial for OOP users. I got most of the definitions here, but none of my notes were from here. I haven't read it, so feedback would be appreciated

- Eight : Really helped me with some additions and suggestions. :) Shoutout to him.

Muhammad Ali - November 24, 2012, 11:46 am

When i started learning OOP, i needed to re-read the whole introduction and beginning. I don't think anyone will actually get this, as i'm decent with procedural, and already struggled with the basics of OOP and had to read and watch guides on the OOP introduction.

Thus you should make this guide allot longer and advanced.

0xStack - November 24, 2012, 11:48 am

I think you related things to Java way too much for a PHP section.

Also

```
public $variable;
```

```
$this->variable = "Define Here";
```

could just be

```
public $variable = "Define Here";
```

This isn't really a good subject to teach unless you are highly experienced with all the concepts. This can't really be considered an introduction, there is way too much to learn.

GaBaNo - November 24, 2012, 11:52 am

I'm planning on building on this tutorial a lot. I referenced it to Java because quite some people that know PHP know Java also, and Java is an OOP language I believe (or, atleast it follows along the guidelines of what makes something OOP)

Muhammad Ali - November 24, 2012, 11:55 am

(Nov 24, 2012 12:48 PM)YvesSaintGreg Wrote: [\[CITE\]](#) I think you related things to Java way too much for a PHP section.

Also

```
public $variable;
```

```
$this->variable = "Define Here";
```

could just be

```
public $variable = "Define Here";
```

This isn't really a good subject to teach unless you are highly experienced with all the concepts. This can't really be considered an introduction, there is way too much to learn.

No, what he does is right, but what you did is right also.

```
public $variable = "Define Here";
```

Is the same as \$OOP->variable = "Define Here"; .

And that means, you could re-use \$Variable anytime you want.

So it could be:

PHP Code

```
<?php
```

```
$Oop->variable = "First text";
```

```
$OOP->variable = "Second text";
```

Short story short, you could re-use variables like an array and echo out different things with the same variable.

0xStack - November 24, 2012, 11:59 am

(Nov 24, 2012 12:55 PM)Yadinho? Wrote: [\[CITE\]](#) No, what he does is right, but what you did is right also.

```
public $variable = "Define Here";
```

Is the same as \$OOP->variable = "Define Here"; .

And that means, you could re-use \$Variable anytime you want.

So it could be:

PHP Code

```
<?php
```

```
$Oop->variable = "First text";
```

```
$OOP->variable = "Second text";
```

Short story short, you could re-use variables like an array and echo out different things with the same variable.

If you read his post you will see that he gives the variable data inside the class not in the usage example, therefore our code is equivalent.

GaBaNo - November 24, 2012, 12:00 pm

(Nov 24, 2012 12:55 PM)Yadinho? Wrote: [\[CITE\]](#) No, what he does is right, but what you did is right also.

```
public $variable = "Define Here";
```

Is the same as \$OOP->variable = "Define Here"; .

And that means, you could re-use \$Variable anytime you want.

So it could be:

PHP Code

```
<?php
$Oop->variable = "First text";
$OOP->variable = "Second text";
```

Short story short, you could re-use variables like an array and echo out different things with the same variable.

Yes, I edited my tutorial to fit what you guys said. I just remembered getting an error whenever I defined a variable straight away, but now that I redid it, I didn't get the error. Oh well.

You're not as stupid as I thought Yadin. ;)

Muhammad Ali - November 24, 2012, 12:11 pm

(Nov 24, 2012 01:00 PM)FallingRain Wrote: [\[CITE\]](#) Yes, I edited my tutorial to fit what you guys said. I just remembered getting an error whenever I defined a variable straight away, but now that I redid it, I didn't get the error. Oh well.

You're not as stupid as I thought Yadin. ;)

You should also use Require instead of include, it's just better and more trustier than include.

jQuery - November 24, 2012, 12:20 pm

Your code structure is extremely messy and distracts the reader from the main purpose. You should probably stick to 1 file when writing something like this.

I noticed that you used include (as the user above stated), it should not be used. Include only produces a warning if there is an error during the process, whereas require produces a fatal error which will then kill the process.

CarrotMilk - November 24, 2012, 12:25 pm

In addition to what has already been said:

I think it's great that you want to make tutorials and try to help out, but something like that been done a million times over. I mean, fair enough if it's a more complicated aspect/side to OOP, but as far as this goes to being "an introduction to OOP", it falls extremely short.

GaBaNo - November 24, 2012, 12:39 pm

(Nov 24, 2012 01:25 PM)CarrotMilk Wrote: [\[CITE\]](#) In addition to what has already been said:

I think it's great that you want to make tutorials and try to help out, but something like that been done a million times over. I mean, fair enough if it's a more complicated aspect/side to OOP, but as far as this goes to being "an introduction to OOP", it falls extremely short.

I called it an introduction, because it is. It teaches you the basics and generally how to use it. I didn't go in depth that much, I know, but it's still useful to those who don't know what it is.

CarrotMilk - November 24, 2012, 12:44 pm

(Nov 24, 2012 01:39 PM)FallingRain Wrote: [\[CITE\]](#) I called it an introduction, because it is. It teaches you the basics and generally how to use it. I didn't go in depth that much, I know, but it's still useful to those who don't know what it is.

I'm not trying to flame you at all here, but this doesn't really teach you the basics at all. From the perspective of someone new to the OOP with PHP scene, would this provide any insight into what it's all about? That's why I'm saying it's not an "introduction" because it doesn't introduce what OOP is or does. In addition, there are plenty of other tutorials which DO. It's not a bad thing having more than one tutorial on a particular subject, but, this could be a million times better and what is posted now is practically useless.

I do appreciate you (or anyone else) taking their time to try and help out, I'm just saying, though.

GaBaNo - November 24, 2012, 1:28 pm

(Nov 24, 2012 01:44 PM)CarrotMilk Wrote: [\[CITE\]](#) I'm not trying to flame you at all here, but this doesn't really teach you the basics at all. From the perspective of someone new to the OOP with PHP scene, would this provide any insight into what it's all about? That's why I'm saying it's not an "introduction" because it doesn't introduce what OOP is or does. In addition, there are plenty of other tutorials which DO. It's not a bad thing having more than one tutorial on a particular subject, but, this could be a million times better and what is posted now is practically useless.

I do appreciate you (or anyone else) taking their time to try and help out, I'm just saying, though. It's better something then nothing. And I did explain what OOP was, or at least, what it's generally used for and why people prefer it.

Like I said, some of this stuff might be off because I'm new to it too, and all the information I've posted is stuff I've learned without looking at definitions, etc. to learn about it.

I never said you were flaming, either.

Eight - November 24, 2012, 1:29 pm

Without trying to sound like a complete douchebag, there are some things I'd like to point out :

> PHP by default is not OOP, you have to make it OOP.

Hmm.. The idea is there but the terminology is not quite right. PHP is a multi paradigm language, meaning you are free to write object oriented or procedural code but the language's structure itself is not fully object oriented and you can't make it so. What I mean to say is that it's not a "mode" that you can activate the the php.ini file or something.

> As you can see, the class name does not have to be the same as the file name, like how it is in java, etc.

It doesn't, but it's highly recommended especially when making autoloaders for your classes. Look up the PSR standards. Technically nothing prevents you from putting two classes in the same file and giving it a strange name but then you would be coding like a pig.

> As you can see, we defined it using the \$this-> method.

Did you try this ? It gave me a syntax error. This syntax is only valid inside your methods. To give your properties a default value, you can simply do public \$variable = 3. If the value requires some relatively complex computations (return value of a function or a multi-line associative array or something), then you can do those inside the constructor, or write specific methods for them then call those from your constructor. For example, you can't do public \$rand = rand() but you can do :

PHP Code

```
<?php
class OOP
{
    public $rand;
    public function __construct()
    {
        $this->rand = rand();
    }
}
```

> Now, as you can see, we are creating a new instance of the OOP class and running it through our

handler.

I think you meant object, an object is the result of instantiating a class.

> And whala, you are done!

Voilà* :^p

In addition to what the other folks said, I would recommend you use `require_once` instead of `require`, this will ensure you don't include the same class definition twice and end up with "class already defined in.." errors. That being said, I think this is a very basic tutorial, in an introduction to OOP I would expect at least some basic, uhm, lexicology ? I mean, what a class is, what an object is, what encapsulation is, what methods and properties are, etc. Then some basic syntax, the different visibility keywords and their meaning, inheritance, and maybe abstract classes and interfaces. I attended a few "intro to C++" classes in the university and that's what the teacher taught them, though he began by stating the differences between C & C++.

Sir Asylum - November 24, 2012, 1:44 pm

This doesn't really explain OOP at all.

Only 1 little thingy but that's it.

GaBaNo - November 24, 2012, 2:09 pm

(Nov 24, 2012 02:29 PM)Eight Wrote: [\[CITE\]](#) Without trying to sound like a complete douchebag, there are some things I'd like to point out :

> PHP by default is not OOP, you have to make it OOP.

Hmm.. The idea is there but the terminology is not quite right. PHP is a multi paradigm language, meaning you are free to write object oriented or procedural code but the language's structure itself is not fully object oriented and you can't make it so. What I mean to say is that it's not a "mode" that you can activate in the `php.ini` file or something.

> As you can see, the class name does not have to be the same as the file name, like how it is in java, etc.

It doesn't, but it's highly recommended especially when making autoloaders for your classes. Look up the PSR standards. Technically nothing prevents you from putting two classes in the same file and giving it a strange name but then you would be coding like a pig.

> As you can see, we defined it using the `$this->` method.

Did you try this ? It gave me a syntax error. This syntax is only valid inside your methods. To give your properties a default value, you can simply do `public $variable = 3`. If the value requires some relatively complex computations (return value of a function or a multi-line associative array or something), then you can do those inside the constructor, or write specific methods for them then call those from your constructor. For example, you can't do `public $rand = rand()` but you can do :

PHP Code

```
<?php
class OOP
{
    public $rand;
    public function __construct()
    {
        $this->rand = rand();
    }
}
```

> Now, as you can see, we are creating a new instance of the OOP class and running it through our handler.

I think you meant object, an object is the result of instantiating a class.

> And whala, you are done!

Voilà* :^p

In addition to what the other folks said, I would recommend you use `require_once` instead of `require`, this will ensure you don't include the same class definition twice and end up with "class already defined in.." errors. That being said, I think this is a very basic tutorial, in an introduction to OOP I would expect at least some basic, uhm, lexicology ? I mean, what a class is, what an object is, what encapsulation is, what methods and properties are, etc. Then some basic syntax, the different visibility keywords and their meaning, inheritance, and maybe abstract classes and interfaces. I attended a few "intro to C++" classes in the university and that's what the teacher taught them, though he began by stating the differences between C & C++.

Holy lord that was a lot of feedback!

Thanks, and as a result, I have added most of what you explained into the tutorial. I also added some definitions and notes throughout the tutorial to help explain it more to new users. :)

Also gave you some credits, since a lot of my additions were on feedback you gave me.

colody32 - November 24, 2012, 2:31 pm

What is the purpose of classes. I know how to code them but don't know why someone would use it rather than just write functions?

I know what people told me that it's reusable and cuts down development time and does reduce having to rewrite code or repeat code.

Can someone here give us an example of a real world way of using a class.

Just write us some code or tell us when you have used classes and why. Why did you make the class rather than use functions.

I think I am not the only one that have these feelings. I do know now the uses of it. I seen tutorial videos of a person in PHP writing classes for different monsters. You can then inherit methods from one class to another or inherit properties.

I would like an example. I think real world examples are best.

GaBaNo - November 24, 2012, 2:35 pm

(Nov 24, 2012 03:31 PM)colody32 Wrote: [\[CITE\]](#) What is the purpose of classes. I know how to code them but don't know why someone would use it rather than just write functions?

I know what people told me that it's reusable and cuts down development time and does reduce having to rewrite code or repeat code.

Can someone here give us an example of a real world way of using a class.

Just write us some code or tell us when you have used classes and why. Why did you make the class rather than use functions.

I think I am not the only one that have these feelings. I do know now the uses of it. I seen tutorial videos of a person in PHP writing classes for different monsters. You can then inherit methods from

one class to another or inherit properties.

I would like an example. I think real world examples are best.

We used a class to make the object. Then we can reference the object to use functions outside of our handler/file. It's a confusing concept to some people. All in all, it's mainly a neat template we can use to program. It's not required at all, but is recommended. It's also good practice if you program Java, as OOP PHP and Java are extremely alike (along with some other programming languages).

colody32 - November 24, 2012, 2:44 pm

(Nov 24, 2012 03:35 PM)FallingRain Wrote: [\[CITE\]](#) We used a class to make the object. Then we can reference the object to use functions outside of our handler/file. It's a confusing concept to some people. All in all, it's mainly a neat template we can use to program. It's not required at all, but is recommended. It's also good practice if you program Java, as OOP PHP and Java are extremely alike (along with some other programming languages).

I never programmed in java. I don't think I be learning that. I will take your word for it that it's similar to php.

I own 2 websites and haven't used classes. I am thinking once I am finished with the projects. I am thinking to rewrite the code to use classes.

I wrote the website piss poor. I now understand my mistakes but will eventually have to go back and rewrite everything. However, before I do that I rather learn more about optimizing the websites code and obfuscation. I am not talking about PHP to obstructed but talking about javascript and html. I just want to make my source code not readable when you first look at it. You have to do some extra work to read the source code.

I was told the only solution to that is to do an based64 encoding. The data encoded in base64 will be able to be readable by the browser. I am not sure if that's true or not.

GaBaNo - November 24, 2012, 2:47 pm

(Nov 24, 2012 03:44 PM)colody32 Wrote: [\[CITE\]](#) I never programmed in java. I don't think I be learning that. I will take your word for it that it's similar to php.

I own 2 websites and haven't used classes. I am thinking once I am finished with the projects. I am thinking to rewrite the code to use classes.

I wrote the website piss poor. I now understand my mistakes but will eventually have to go back and rewrite everything. However, before I do that I rather learn more about optimizing the websites code and obfuscation. I am not talking about PHP to obstructed but talking about javascript and html. I just want to make my source code not readable when you first look at it. You have to do some extra work to read the source code.

I was told the only solution to that is to do an based64 encoding. The data encoded in base64 will be able to be readable by the browser. I am not sure if that's true or not.

You can use Javascript to base64 decode an encoded page and it will display it on any browser that supports Javascript, as it is a built in function.

colody32 - November 24, 2012, 3:21 pm

(Nov 24, 2012 03:47 PM)FallingRain Wrote: [\[CITE\]](#) You can use Javascript to base64 decode an encoded page and it will display it on any browser that supports Javascript, as it is a built in function. but the browser cannot natively understand encoded data in the base64 form?

GaBaNo - November 24, 2012, 3:37 pm

(Nov 24, 2012 04:21 PM)colody32 Wrote: [\[CITE\]](#) but the browser cannot natively understand encoded data in the base64 form?

No, it cannot. Only if the browser supports javascript, and if you have javascript enabled.

colody32 - November 24, 2012, 3:59 pm

(Nov 24, 2012 04:37 PM)FallingRain Wrote: [\[CITE\]](#) No, it cannot. Only if the browser supports javascript, and if you have javascript enabled.

Ya, if javascript is enabled and working. If I had html code and javascript code encoded in base64. Would the browser accept the encoded data without me having to directly decode it?

I know the eval function where I can run a decode type function.

However, was told I don't need it. Was told the browser can understand base64 encoded data or character code.

I just want to hide the html and javascript of my website. I want to avoid having to decode it for the browser. I was told you can either encoded it in base64 or convert the characters to their numerical value.

CarrotMilk - November 24, 2012, 4:02 pm

There's no way you can hide front-end code. Don't even bother. Not worth it.

GaBaNo - November 24, 2012, 4:16 pm

(Nov 24, 2012 05:02 PM)CarrotMilk Wrote: [\[CITE\]](#) There's no way you can hide front-end code. Don't even bother. Not worth it.

This is completely true. I think view-source is pointless. Just another way for leechers to get some rep for something they didn't do <.<

0xStack - November 24, 2012, 4:34 pm

(Nov 24, 2012 05:16 PM)FallingRain Wrote: [\[CITE\]](#) This is completely true. I think view-source is pointless. Just another way for leechers to get some rep for something they didn't do <.<

Viewing webpage sources is not pointless and it can be very helpful if you are a front-end developer.

Also I believe CM was saying that it is pointless to try and encrypt or obfuscate it because it can be decrypted any which way.

GaBaNo - November 24, 2012, 4:37 pm

(Nov 24, 2012 05:34 PM)YvesSaintGreg Wrote: [\[CITE\]](#) Viewing webpage sources is not pointless and it can be very helpful if you are a front-end developer.

Also I believe CM was saying that it is pointless to try and encrypt or obfuscate it because it can be decrypted any which way.

I understand what he was saying. My point was that being able to see someone else's code is sort of stupid sometimes.

CarrotMilk - November 24, 2012, 4:40 pm

(Nov 24, 2012 05:37 PM)FallingRain Wrote: [\[CITE\]](#) I understand what he was saying. My point was that being able to see someone else's code is sort of stupid sometimes.

Well, how do you expect browsers to be able to render said code without, well...seeing it?

#OpenSourceYOLO

GaBaNo - November 24, 2012, 4:53 pm

(Nov 24, 2012 05:40 PM)CarrotMilk Wrote: [\[CITE\]](#) Well, how do you expect browsers to be able to

render said code without, well...seeing it?

#OpenSourceYOLO

Well, I get that... but an option that allows kids to leech and see code?

jQuery - November 24, 2012, 5:01 pm

(Nov 24, 2012 05:53 PM)FallingRain Wrote: [\[CITE\]](#) Well, I get that... but an option that allows kids to leech and see code?

Holy shit you sound like Apple. The world would be no where without sharing creations.

GaBaNo - November 24, 2012, 5:51 pm

(Nov 24, 2012 06:01 PM)jQuery Wrote: [\[CITE\]](#) Holy shit you sound like Apple. The world would be no where without sharing creations.

I get that too. But when all people do is abuse this and leech code most of the time, it's not very useful, correct?

0xStack - November 24, 2012, 6:41 pm

(Nov 24, 2012 06:51 PM)FallingRain Wrote: [\[CITE\]](#) I get that too. But when all people do is abuse this and leech code most of the time, it's not very useful, correct?1 stay 4bus1ng HTML cod3 i aer l337

barricas - November 24, 2012, 8:14 pm

(Nov 24, 2012 06:51 PM)FallingRain Wrote: [\[CITE\]](#) I get that too. But when all people do is abuse this and leech code most of the time, it's not very useful, correct?

Even if view-source didn't exist in browsers, people would still download the pages and open the html file anyway...

You can also obfuscate your code like google does. Doesn't hide, just makes harder to read/understand.

Anyway I don't care with that shit.

colody32 - November 24, 2012, 10:53 pm

(Nov 24, 2012 05:34 PM)YvesSaintGreg Wrote: [\[CITE\]](#) Viewing webpage sources is not pointless and it can be very helpful if you are a front-end developer.

Also I believe CM was saying that it is pointless to try and encrypt or obfuscate it because it can be decrypted any which way.

Ya, but I never said I want to make 100% sure bullet proof way.

It be the same way with software programs and computer chips.

You can always reverse engineer computer software / firmware.

However, my point is to hide my source from noobs. People that don't know how to decryption the stuff will spend time having to learn it. Eventually they will not learn it and just move on to easy victims. Those type of people are looking for quick way to copy and paste code.

if someone knows how to decrypt my javascript and html it don't matter. I am sure if someone has that ability they won't actually be stealing my code. They may look at it to learn if I did anything new.

If you look at javascript and html. pretty much anyone that writes such code can't really be creative with it. It's more static.

I mean I am pretty sure we are all writing the same code to do the same effects. I doubt anyone is doing anything different.

I hide most of Processing that I cannot afford to let the public view it. I have it on the server side.

The javascript is just the GUI and layout that is it nothing really worth protecting. I just want extra work. So I don't care if it can be decrypted. As long no one can just go to my website and just click view source and go oh boy let me copy that and this and that.

I am using jquery and alot of people that aren't technical or programmers. If they see this in the source. They will think less of my website. They will think it's unprofessional. They would rather have me use javascript. Using libraries is a turn off to people that isn't technical.

To them it's like me trying to compete with apple. I build my own computer by going out and buying a motherboard and computer parts. Which most will be like well apple makes the boards themselves. I would on the other hand take what others made and just assemble them together which isn't actually called creating your own computer. More like customizing what you're already given by the market.

I just don't want my average users to see that I use jquery. To prevent them from thinking I am unprofessional.

so it's for non programmers to not easily take a peak at my source.

(Nov 24, 2012 05:40 PM)CarrotMilk Wrote: [\[CITE\]](#) Well, how do you expect browsers to be able to render said code without, well...seeing it?

#OpenSourceYOLO

that's the same argument with software and computer chips.

Did you know you can actually open up your computer and actually use logic probs to actually see what is being transmitted from one chip to another.

You can run a program and actually see the data go from one side to another. At this level you cannot actually encrypt the data.

This is actually what they do when they make mod chip for the game console. They use these methods only if they don't have access to reading material of the op codes for the chips.

ASM even allows you to easily read or see what a software is doing on a low level.

So then I ask why do big corps spend millions to encrypt their source code?

Simcity 4 a pc video game the source code is encrypted. They won't allow anyone to read the source. It's closed sourced.

I want to decrypt their sources. However, have a hard time figuring out how to do it. I am not the only one. One guy spent 4 years trying to figure it out and failed.

This game was made in 2004 and still today no one has decrypt the source code. However, people did make a pirated copy of it. Which isn't hard to do.

(Nov 24, 2012 09:14 PM)barricas Wrote: [\[CITE\]](#) Even if view-source didn't exist in browsers, people would still download the pages and open the html file anyway...

You can also obfuscate your code like google does. Doesn't hide, just makes harder to read/understand.

Anyway I don't care with that shit.

Ya, but my point is that they will need to know how to do that. The time that is needed to not just learn that but to actually go looking for the files. It won't be worth the noobs. The noobs are looking for quick ways.

Right now if you go to your website all you need to do is right click and hit view source. Then bam you got access to the websites source code. You can copy it.

Also to others saying that we should allow open source. html and java script is more static meaning theirs isn't much room of creativity or creating your own work. Pretty much we all are using the same code.

I mean in html we all have to be using the <body> tags. so, what I am saying it's nothing worth stealing unless you don't know much about html.

my point is to make my source harder to read and make someone work for it.

badbrains - November 25, 2012, 7:46 am

This is almost as useful as Oats "[TUT.] THE BASICS O? PHP" ?

CarrotMilk - November 25, 2012, 7:49 am

(Nov 25, 2012 08:46 AM)PeopleUnderTheStairs Wrote: [\[CITE\]](#) This is almost as useful as Oats "[TUT.] THE BASICS O? PHP" ?

Almost. Not quite that retarded, though ?

GaBaNo - November 29, 2012, 2:19 pm

Bump, edited it a bit... still not the best, but hey... At least I'm contributing. :)

CarrotMilk - November 29, 2012, 2:21 pm

(Nov 29, 2012 03:19 PM)[MULTIPLAYER] Wrote: [\[CITE\]](#) Bump, edited it a bit... still not the best, but hey... At least I'm contributing. :)

That's a terrible attitude, because...you're not, for the reasons aforementioned.