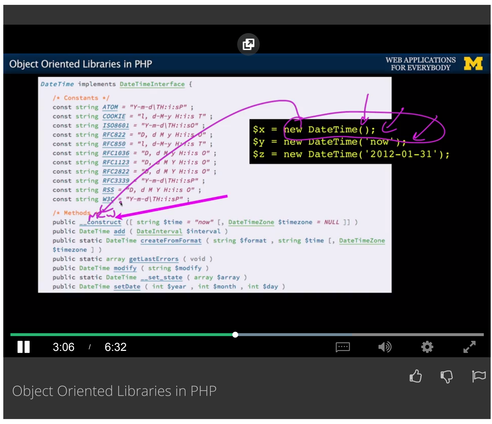
Course Materials

Lecture Content

Bonus Materials



Object Oriented Libraries in PHP

Have a question? Discuss this lecture in the week forums.

Interactive Transcript

English

0:08
So now that I've showed you a little bit about how you might construct an object, and I mean the Person one isn't all that amazing. But you get some of the mechanisms of it, I think it's going to help as you want to read documentation. So to review, we're going to see two new operators. The one we've seen already is the thing that basically says within. So I'm looking for the function named format, within the object name z, right? And the same's true for variable like $colleen first name, that's the firstname variable within the colleen object. That's the within, I think of that as within.

0:45
Okay, it turns out there are things that are in classes that we can access. So even though they're templates, there's certain code that's so static it doesn't change and so it's not dynamic and so we can say, you know what? I want to pull the code, I just want to pull that code out from that template and run that code. A key we'll see in a second, when you're doing a static item, there is no $this.

1:09
So, this is the class, and this is object.

1:15
So, some things can only be run when there's an object that's been populated, but some things are really simple. It's a way I like putting old library code and in this case, it's a constant. So DateTime::RFC822. So that says, there is a thing that's statically defined inside of DateTime, pull it out, I want to use it. It'll be easier when we see an example. So this is within a statically within. So this is, let me just draw this better. This is WITHIN OBJECT. And this is WITHIN CLASS. This is STATIC. This is DYNAMIC.

1:59
Meaning it's gotta be an instance or an object, okay? So, here we're looking at some object oriented documentation, right? So you have constants in there, and in this case these are just statically defined constants. And you say hey, I would like the RFC822 variable constant which is this guy right here, inside of date time. And so that's how you pull this string. Now these are just different date formats that you're going to need for different situations of formatting dates, okay? So that's one thing, constants within classes. That's what you need to know and you need to know how to use those. Then there are some special methods inside the class and we'll talk specifically about constructing. But you don't see a thing that shows the documentation for what you're allowed to do in all these creation, the new operation. Somebody might argue that this should have been called, new. Say like, under new that says, when I'm running new, this is it. But you just have to equate new with construct, because construction of a object is kind of the generic OO notion of take the template, make an instance, right? And so this is the act of constructing. And so new is the operator that caused the constructing to happen. But when you're reading the documentation, you have to go read and find the __construct to tell you what you're allowed to do in those parentheses. So what this really says here is this really just a function call that's saying this is optional, right? These are two optional parameters, the first optional parameter is when. And that's optional and if you don't specify it, that means the default is now. And then the second parameter is a time zone, but I haven't got it in any one of these things so I assume it's the current time zone that's the server's time zone. And so you can read and understand what you're allowed to do on the new calls by looking for the, __construct method. And look at at that. Right here, it says Methods.

4:00
An hour ago you didn't know what a method was. Now, you do know what the method is. It's a bit of code that lives inside of a class and instances of that object.

4:09
Okay, so sometimes we have static methods and they're methods that we can access from the class itself. So here's an object, x = new DateTime, that's making a regular object, but this getLastErrors, if we read the documentation it says static here. And that means it doesn't depend on this, that's the $this. But it also means that you could just call it and call the class directly. So we're saying, go into the DateTime class and call the getLastErrors function, okay? And so that's what this :: is. It says go into the class and get that code. And this is a way, in this case, to go get the errors that might have happened upon constructing. Because when you're constructing using a new, it either gives you back good stuff or an empty thing. If you want to see the errors, went wrong, you gotta call the class and say, hey class, last time you were constructing, what errors might you have encountered?

5:06
Now, a normal variable, the normal method doesn't have the word static in it. And so that we've sort of done, we make a new class, we have the class and we're going to access this format method deep inside the class. And so that's inside the class. So that means it's inside the object, z is in an object, and format is a method inside the object. So you'll see most of the things are not static. You got one static, one static, most of the things here are just dynamic and that means the only way you can access them is with the arrow notation that's within.

5:40
Now, there's a whole bunch of things that are really very cool and it has to do with the fact the PHP came to the object oriented pattern late. So they were able to take and look at all the other OO languages that I talked about a few lectures back and steal all the best stuff. So you see these things that have to do with underscore, and they have to do with sort of the life cycle of objects, which we'll talk about in a bit. But in some ways there's a lot of really cool stuff. And this wakeup is when something comes out of the session back into PHP, we don't even talk about the session yet, this method gets called. So you can build an object that asks to be called and consulted at the moment that it's being reloaded and woke back up in to memory. So up next we'll talk in some more detail about what I just got done talking about and that is the lifecycle of the objects.

Downloads

Lecture Video mp4
Subtitles (English) WebVTT
Transcript (English) txt

Would you like to help us translate the transcript and subtitles into additional languages?