

Diploma in Web Development – Part II



PHP Development – Week 1 Object-Oriented PHP

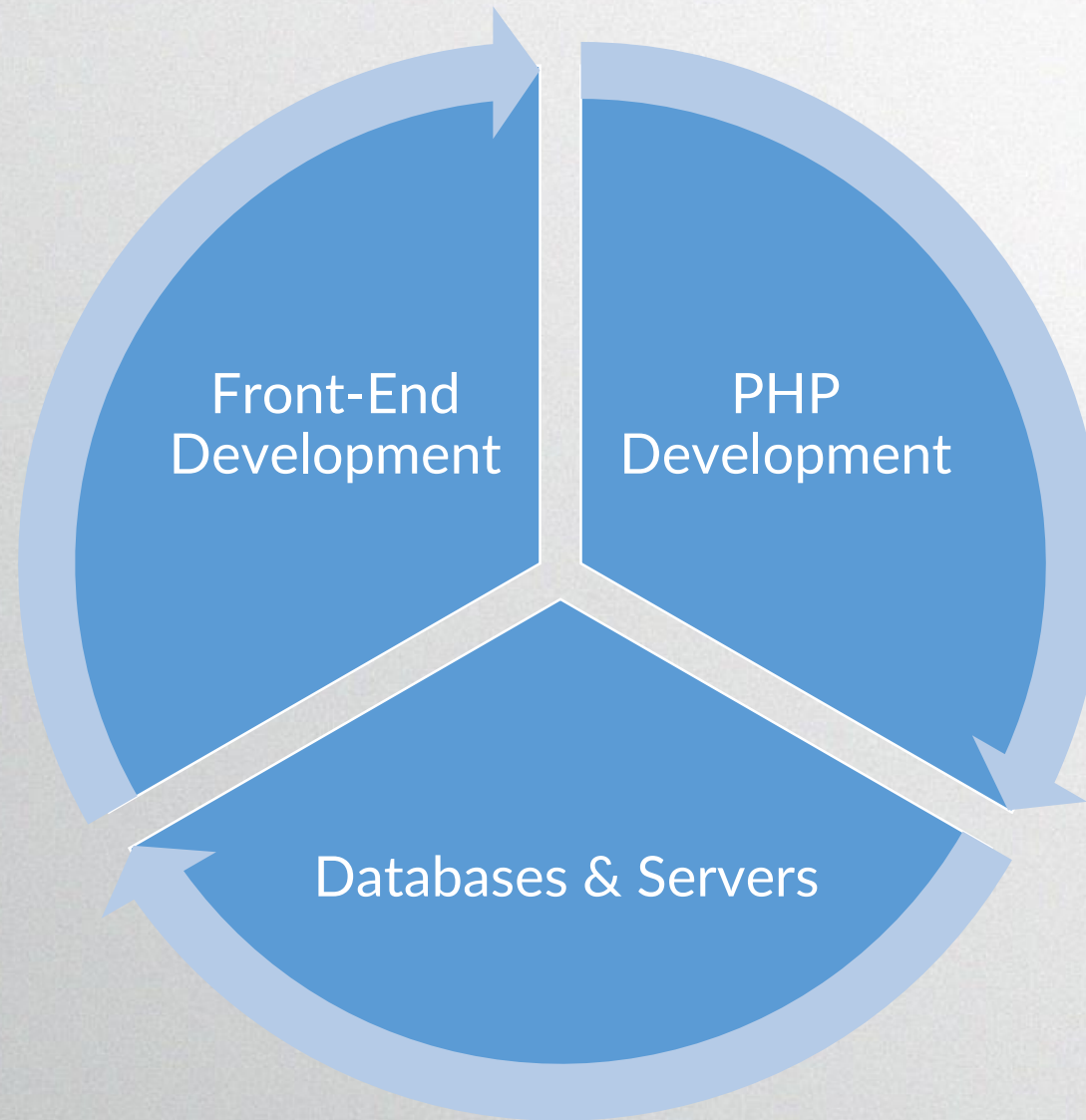
Presented by:
Julian Quirke
Web Development Educator



Welcome New Students!



Course Overview



PHP Development



Object-Oriented PHP

- 1
 - Introduction to PHP
 - PHP Class Overview
 - Magic Methods

Practical:

- Shawpify: PHP Structures for Website Template



Agenda!



Cookies & PHP Sessions

- 2 ➤ What is a Cookie?
- Creating User Sessions
- PHP Sessions Demo

Practical:

- Shawpify: Creating a User Session



Agenda!



Error Handling & Advanced Development

- 3 ➤ Member Visibility
 - Abstract Classes & Interfaces
 - Error Handling in PHP

Practical:

- Shawpify: Login Functionality



Agenda!



PHP & Security

- 4 ➤ Encrypted Data & HTTPS
- Data Validation with Hashing
- Storing Passwords Securely

Practical:

- Shawpify: User Login (I)



Agenda!



Let's Begin!



PHP HyperText Preprocessor



PHP HyperText Preprocessor

is an open-source general-purpose scripting language that is especially suited for web development

PHP can be embedded directly into HTML



Object-Oriented PHP

Generates HTML code

Embedded into HTML

Executes on the **Web Server**



PHP

Demo



PHP Class Overview



Class



Class

is a template from which an object instance can be created



Creating a Class

- Use the “**class**” keyword
- Naming classes in PHP:
 - **MyFirstClass**
- Note: Classes must be declared before an object can be instantiated

```
class MyFirstClass {  
  
}
```



Creating Methods & Properties

```
class MyFirstClass {  
    $myProperty = "a property";  
  
    function myFunction() {  
        echo "this is a class method";  
    }  
}
```

(Also known as **members**)



PHP Class Overview

Instantiating & Using an Object

- Use the “**new**” keyword
- Skip the properties’ “\$” symbol!
- Call Properties & Methods of instances with the “**->**” operator
- Calling class methods (“**::**”) is Bruce Lee stuff (we’ll see later!)

```
$myInstance = new MyFirstClass();  
echo $myInstance->myProperty; //a property  
$myInstance->myFunction(); //call to instance method
```



Creating a Child Class

- Use the “**extends**” keyword
- All public & protected methods & properties from the parents can be accessed from the child

```
class MySecondClass extends MyFirstClass {  
}  
  
$myInstance = new MySecondClass();  
echo $myInstance->myProperty; //a property  
$myInstance->myFunction(); //call to instance method
```



PHP

Demo (II)



Magic Methods



Magic Methods

in PHP are methods which have additional functionality automatically built into the PHP programming language



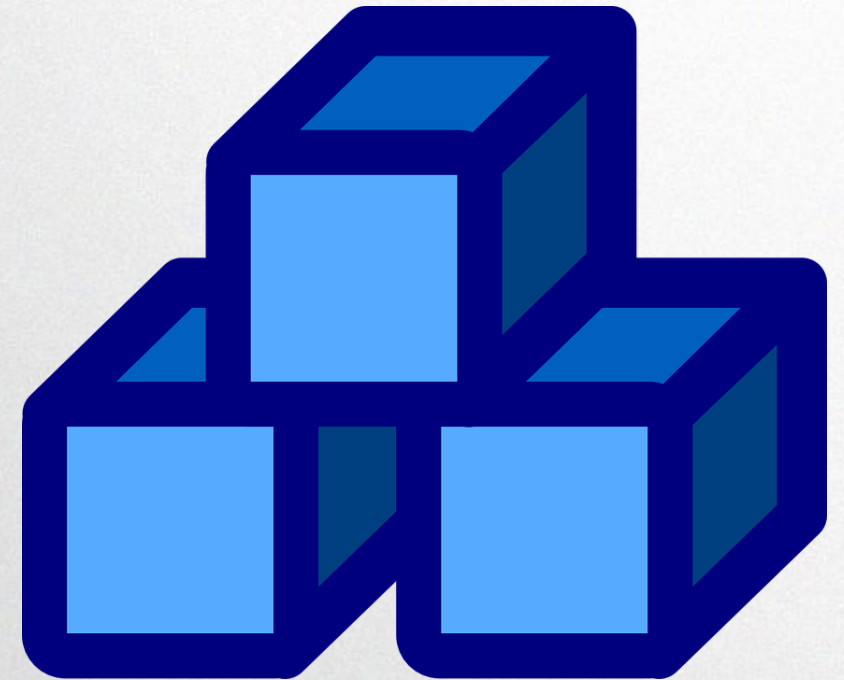
Magic Methods

- Special method names that have “magical” functionality
- There are **exactly** 15 magic methods in PHP
- Denoted by preceeding “__” i.e. two underscores
 - **Only** use this naming convention if using a predefined magic method



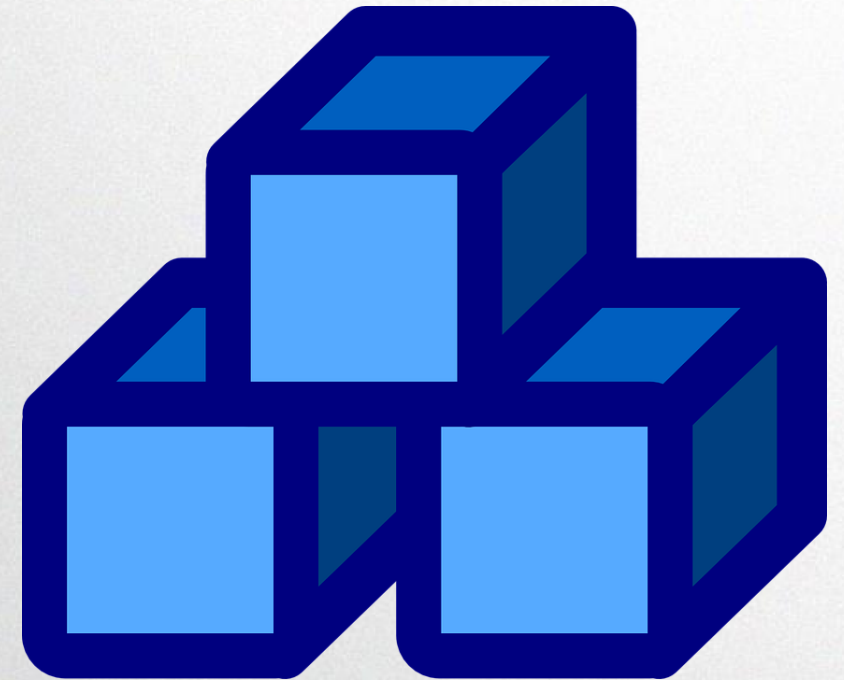
“__construct()”

- Invoked when an object instance is created
- Allows us to initialize in a special way
- We can add our own construct method to our classes
- Child classes should run their parent construct methods



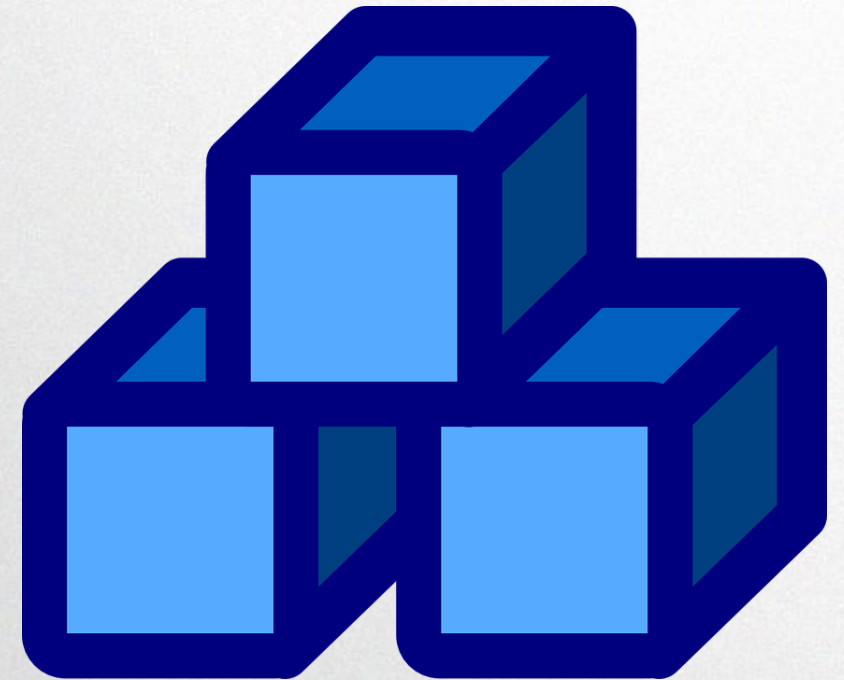
“__get()”

- Invoked when the value of an inaccessible property is accessed
- Allows us to control how our object's assets are seen
- Allows us to show otherwise protected information via property access
- Can be used to generate php errors of improper use of the object (if we wish)



“__set()”

- Invoked when an inaccessible property is assigned to a new value
- Allows us to control how properties are changed outside the class
- Can be used to generate php errors of improper use of the object (if we wish)



Magic Methods

Demo



PHP Development Semester

- The next session is “Cookies & PHP Sessions”
 - What is a Cookie?
 - Creating User Sessions
 - PHP Sessions Demo

- Recordings are available within 24 hours after the live webinar
 - Go to www.shawacademy.com and then the Top Right Corner – **Members Area**



Next Lesson is

Cookies & PHP Sessions

- Understand the purpose and makeup of browser cookies.
- You will learn how to create a **memorable interaction** between a user and a web application using sessions.



www.shawacademy.com



www.facebook.com/shawacademy



www.twitter.com/shawacademy



support@shawacademy.com

