The PHP Object

Daryl K Wood

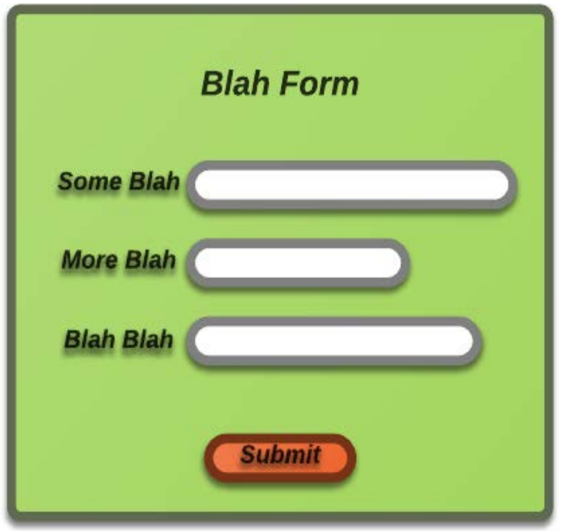
@datashuttle | www.datashuttle.netwww.linkedin.com/in/datashuttle

The PHP Object

The PHP Object

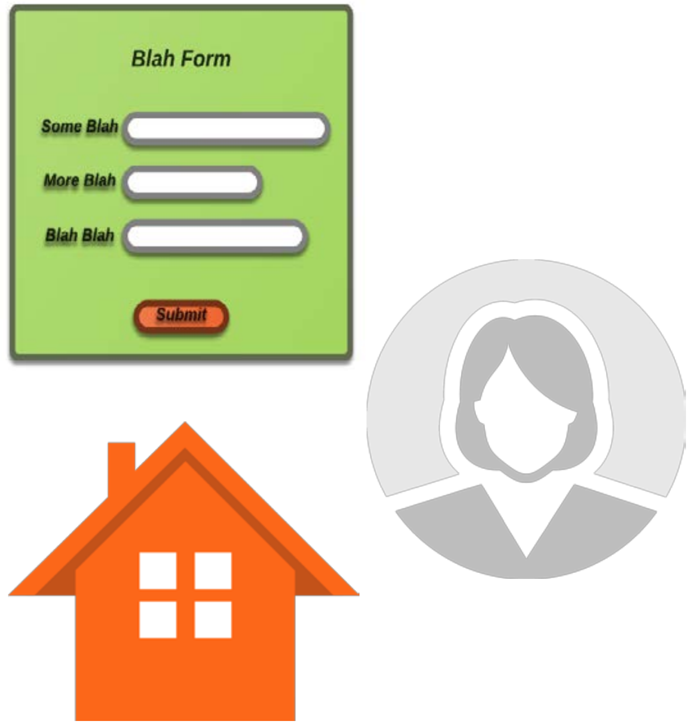
The object and the object data type
Instantiating an object from a class
Working with object property values
Calling object methods
Using constants to our advantage
Best practices

The Object



What is an object
The object data type
Why use an object

What Is an Object?



A container of information

A representation of something

The Object Data Type

The "object" data type
A compound data type

Why Use an Object

To represent a collection of common and similar data values

Allow for easy business modeling

To provide single and multiple use containers for reusable code

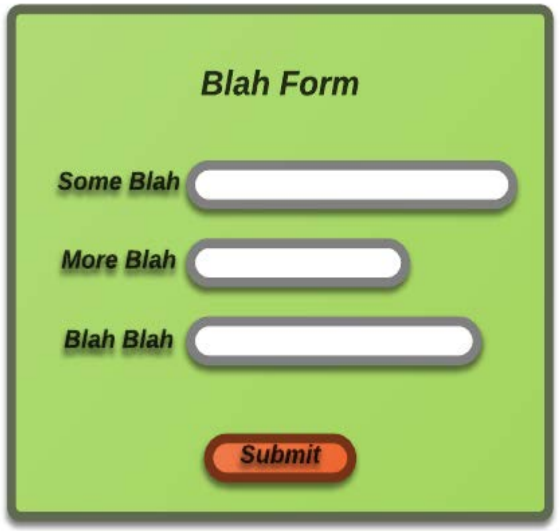
To simplify management of complex software

More scalable and allows for easier automated testing

The PHP Object Summary

The PHP object
The object data type
Why use objects and object-oriented programming

Object Creation and Destruction



How to create an object

How to destroy an object

How to Create an Object

The “new” keyword
The “clone” keyword

Creating with the New Keyword

The class name reference can include parenthesis or not.
Best Practice: Variable name starts with lower case letter or underscore, followed by camel-cased characters.

IDE place holder

Duplicating with the Clone Keyword

The clone keyword will duplicate an object including all the original member properties, methods and constants.

Objects are passed by default without the clone keyword.

Best Practice: Only clone another object if change to object properties is necessary.

IDE place holder

How to Destroy an Object

Using unset() PHP function
Assigning a “null” value
At application termination

Destroy an Object

Destroy an object with unset or re-assigning the object to null.
Explicitly destroying an object is only necessary if required by an application.
Note: an object is destroyed automatically at successful run-time termination.

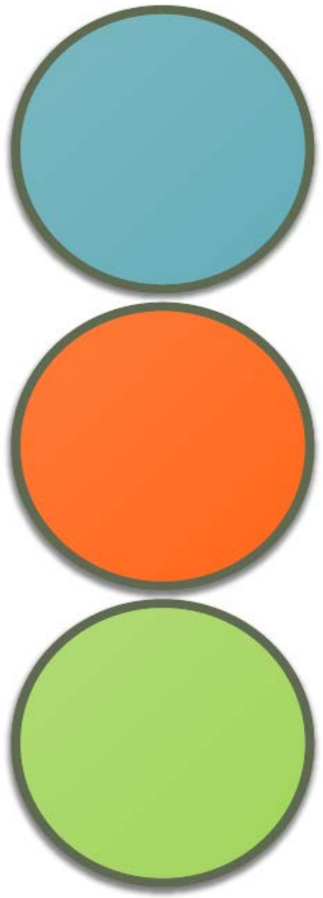
IDE place holder

The PHP Object Summary

Instantiate an object with “new”
Using “clone” to duplicate an object
How to delete an object

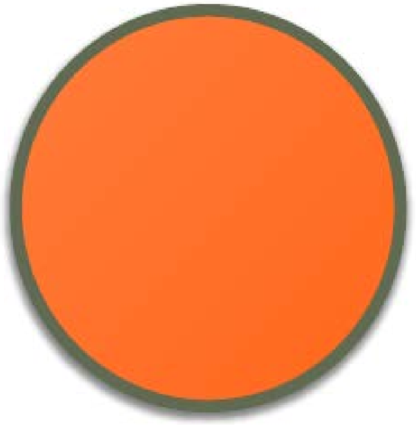
Object Properties

Obtaining Object Properties



Direct Access
Via a getter method
Using $this within class scope

Obtaining an Object Property



Direct access
Via a getter method
Via reference to $this within

Obtaining an Object Property via Direct Access

Obtain the object property with a “->” and assign to a local variable.

IDE place holder

Obtaining Object Properties Summary

Obtain properties by one of three methods

Through direct access

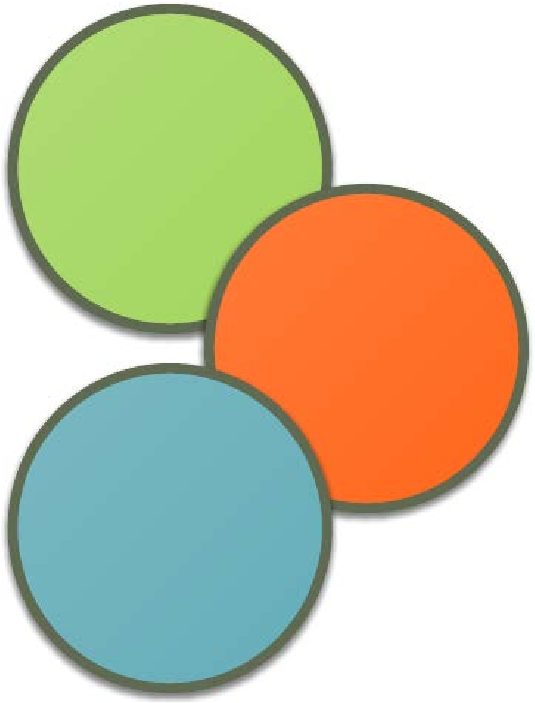
Using a getter method

Internally referencing with $this

Changing Object Properties

Through reassignment
Via a setter method
Using $this within class scope

Changing an Object Property



Through re-assignment
Via a setter method
Via reference to $this within

Changing an Object Property Through Re-assignment

This technique is just like assigning a standard PHP variable, but with object syntax. Note: In this syntax, do not prefix the dollar sign ($) to the property name as it is an explicit “name” property..

IDE place holder

Changing an Object Property via a Setter Method

This technique assumes the method names are defined in the class.
Note: The set method call passes two parameters and assumes a generic set method written to accept two parameters.

IDE place holder

Changing an Object Property via Reference to $this

This technique changes an explicit member property within a member method. Note: The dollar sign ($) is prefixed to the property reference here as it refers to the property parameter.

IDE place holder

Changing Object Properties Summary

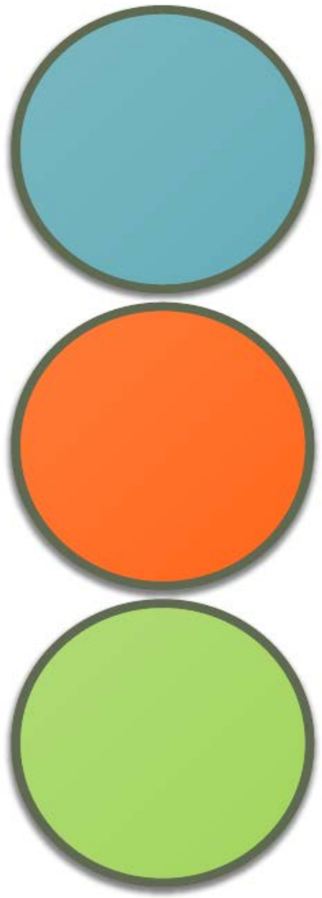
Changing object properties by one of three methods

Through direct access

Using a setter method

Internally referencing with $this

Creating New Object Properties



Through reassignment
Via a setter method
Using $this within class scope

Creating a New Object Property

Through assignment
Via a setter method
Via reference to $this within

Creating a New Object Property

This technique assigns a property that did not initially exist in the class declaration. Note: In this syntax, do not prefix the dollar sign ($) to the property name as it is an explicit “id” property.

Changing an Object Property via Reference to $this

This technique changes a member property within another member method.
Note: The dollar sign ($) is prefixed to the property reference here as it refers to the property parameter.

IDE place holder

Creating New Object Properties Summary

Creating new object properties by one of three methods

Through direct access

Using a setter method

Internally referencing with $this

Destroying Object Properties

Through reassignment
Via a setter method
Using $this within class scope

Destroying an Object Property

Using the PHP unset() function
Via a setter method
Via reference to $this within

Destroying an Object Property Using unset()

This technique destroys the name property value.
Note: If the property was declared within the class, then the property is set to null and can be reassigned. The class property declaration remains after using unset().

IDE place holder

Destroying an Object Property via a Setter Method

This technique destroys a member property from a generic set() method. It is assumed a second parameter for the set() method is declared with an optional second parameter set to null.

IDE place holder

Destroying an Object Property via Reference to $this

Note: This set() method has a second parameter set to null by default.

IDE place holder

Destroying Object Properties Summary

Destroying object properties by one of three methods

Through assignment

Using a setter method

Internally referencing with $this

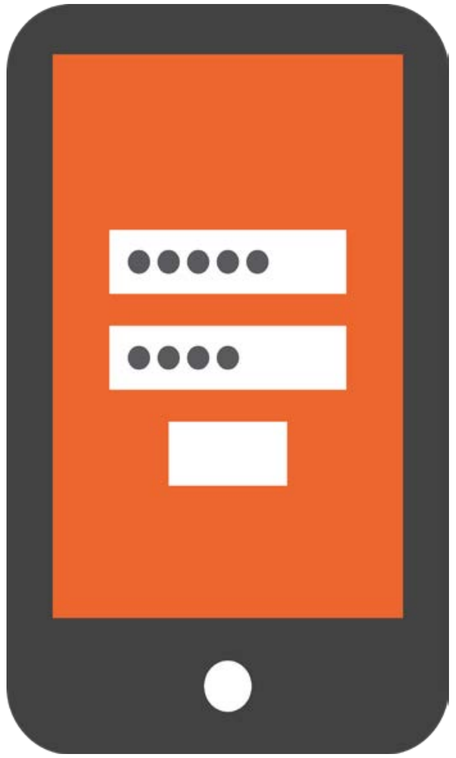
Object Methods

Object Methods

```
public function phoneHome(){}
public function getLunch(){}
public function eatLunch(){}
```

Calling an object method
Returning values
Setters (mutators) and getters (accessors) and more
Adding new methods as needed

Calling an Object Method



Direct call from the object
From within using $this

Calling an Object Method Directly

This technique calls a declared method from the object.
Note: The called method has to be declared within the class, unless using a magic method. If the method is not declared, and a magic method is not used, then a warning is issued.

IDE place holder

Calling an Object Method from Within via Reference to $this

This technique delegates setting form tag arguments to dedicated methods.

IDE place holder

Other Method Techniques

Other Method Techniques

Returning Values
Setters and getters
Chaining methods

Returning Values

Returning null
Returning boolean
Returning values

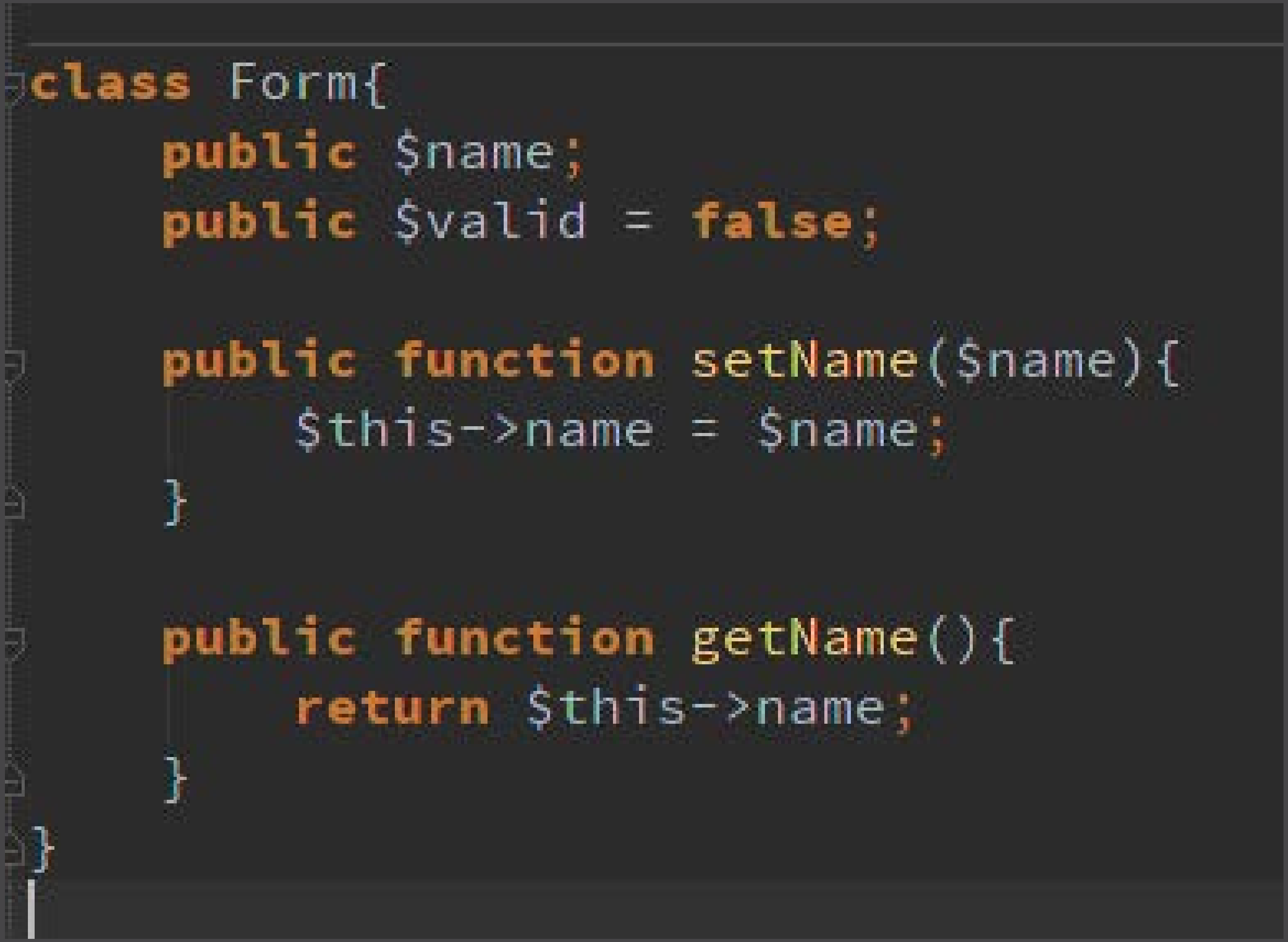
Returning Values from Object Methods

Just like functions, object methods return null if not explicitly used. Use the return keyword if needed.
Returning boolean is appropriate if testing at call time for method success is required.

Setters and Getters

Setters (mutators)
Getters (accessors)

Setters/Getters



```php
class Form{
     public $name;
     public $valid = false;

     public function setName($name){
          $this->name = $name;
     }

     public function getName(){
          return $this->name;
     }
}
```

Method Chaining

Returning $this
Simplifies multiple actions

Chaining Object Methods

Chaining object methods helps to reduce the amount of coding required to set or execute multiple method actions in sequence.

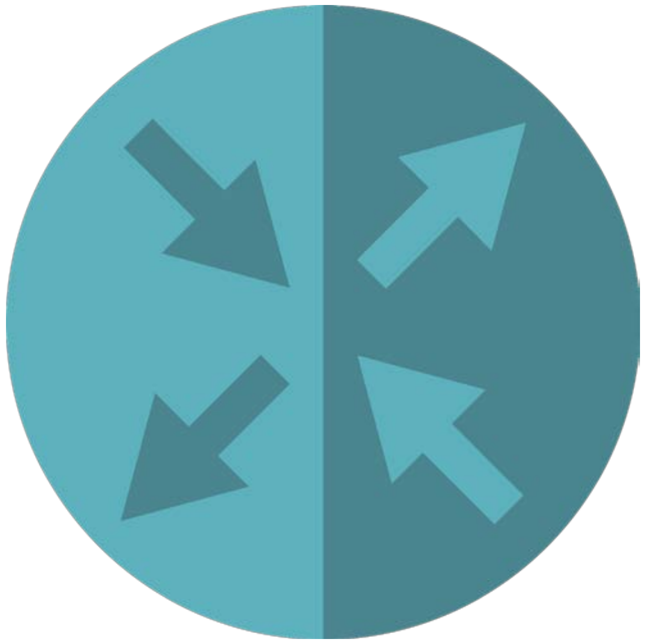
IDE place holder

Method Techniques Summary

Returning nulls, values or booleans
Getters/Setters
Method chaining

Object Constants

Object Constants



Scope relevance

Common use case

Declared within the class prior to run time

Scope Relevance

Within the class as declared
Available if sub classed
Protected from global naming collisions

Class Constants

Available within the class scope.
Protected from global scope naming collisions.

Common Use Case

As a class level reference
As a static reference

IDE place holder

Object Constants Summary

Scope relevance

Common use case