

Class

Define Class	<code>class class_name { }</code>
Define property	<code>var \$property_name ;</code>
Define method	<code>function fun_name() { ... }</code>

Instances

Define Instance	<code>\$Student1 = new student;</code>
set value to property	<code>\$student1->firstName = "ex";</code>
calling object function	<code>\$student->getName();</code>
Refer to the instance	<code>\$this->name;</code>

Visibility modifiers

Public	accessed from anywhere	<code>var \$property_name;</code>
Protected	accessed only from this class and subclasses	<code>Protected \$property_name;</code>
Private	accessed from inside the class only	<code>Private \$property_name;</code>

Inheritance

Define Subclass	<code>class child extends parent { }</code>
-----------------	---

static modifier

Static property	<code>public static \$property_name ;</code>
Class constant	<code>public const CONSTANT_NAME_UPPERCASE;</code>
Calling Static/constant property from inside the class	<code>self::\$property_name;</code>
Calling Static/constant property from outside the class	<code>class_name::\$property_name;</code>

Inheritance	Static property is shared variable between class and its sub classes , any change in one of them will change the others.
-------------	--

Calling parent class static method	<code>parent::method_name();</code>
------------------------------------	-------------------------------------

Late static Binding	<code>static::\$property_name;</code> to allow static property inheritance and don't bind static property to first self use only
---------------------	--

Magic Methods

Magic method	-Magic methods are special methods which override PHP's default's action when certain actions are performed on an object. -Must be Public. -use <code>__</code> before method name.
--------------	---

Magic Methods (cont)

Constructor method	Method will be called each new instance is created <code>public function __construct(\$arg1='Default value',\$arg2,...) public function __construct(\$args=[])</code>
Destructor method	Method will be called when the last reference to instance is destroyed <code>public function __destruct()</code> use <code>unset(\$instance)</code> method to destroy the instance.
Clone method	Method will be called when you use clone keyword <code>\$ins1 = clone \$ins2;</code> method will copy all Instance data to another instance function <code>__clone()</code>
autoload method	Method will be called when PHP encounters an unknown class -Define a function : <code>function my_autoload() { }</code> -Register autoload in php SPL : <code>spl_autoload_register('my_autoload')</code>

Overloading

Dynamic Properties

when you get the value of undefined property -> error notice,

But when you set the value of undefined property -> it will define and set

Example:

```
Class student {
}
$s1 = new student ;
echo $s1->name; //error
$s1->name = waleed //set dynamic property
;
echo $s1->name; //waleed
```

Compare objects

== return true if tow instance :

- have the same refrence
- or have matching properties

=== return true only if Instances have the the same refrence

Functions for Class

get_declared_classes() return array of declared classes in the file

class_exists(\$string) take a string:ClassName and return true if the class is declared

Functions for Instance

get_class(\$object) return object class name

is_a(\$object, \$className) return true if the \$object has the same class name as the \$className

Functions for Properties

get_class_vars(\$className) return list of properties defined in this class using class name

get_object_vars(\$object) return list of properties defined in this class using object instance

property_exists(\$mixed, \$string) return true if the property name exist on the (class or object instance)

get_class_methods(\$mixed) return list of methods defined in this class using class name

method_exists(\$mixed, \$string) return true if the method name exist on the (class or object instance)

Functions For Inheritance

get_parent_class(\$mixed) return the parent class for (ClassName or object Instance)

is_subclass_of(\$mixed, \$string) return true if the (ClassName or object Instance) is child of the given class name

class_parents(\$mixed) get all parent classes of this (ClassName or object Instance)

function for static binding

get_class() return the parent class use this function

get_called_class() return the actual runtime class



By **Waleed Mohamed**

cheatography.com/waleed-mohamed/

Published 16th December, 2021.

Last updated 16th December, 2021.

Page 2 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>