

Classes and Objects in PHP

PHP extends keyword

In PHP, to define a class that inherits from another, we use the keyword `extends` :

```
class ChildClass extends  
ParentClass {  
}
```

The newly defined class can access members with `public` and `protected` visibility from the base class, but cannot access `private` members. The newly defined class can also redefine or override class members.

PHP Constructor Method

A constructor method is one of several magic methods provided by PHP. This method is automatically called when an object is instantiated. A constructor method is defined with the special method name

`__construct` . The constructor can be used to initialize an object's properties.

```
// Dog class inherits from Pet class.  
class Dog extends Pet {  
    function bark() {  
        return "woof";  
    }  
}
```

```
// constructor with no arguments:  
class Person {  
    public $favorite_color;  
    function __construct() {  
        $this->favorite_color = "blue";  
    }  
}  
  
// constructor with arguments:  
class Person {  
    public $favorite_color;  
    function __construct($color) {  
        $this->favorite_color = $color;  
    }  
}
```

PHP class



In PHP, classes are defined using the `class` keyword.

Functions defined within a class become methods and variables within the class are considered properties.

There are three levels of visibility for class members:

- `public` (default) - accessible from outside of the class
- `protected` - only accessible within the class or its descendants
- `private` - only accessible within the defining class

Members can be defined to be static:

- Static members are accessed using the Scope Resolution Operator (`::`)

Classes are instantiated into objects using the `new` keyword. Members of an object are accessed using the Object Operator (`->`).

```
// Test class
class Test {
    public $color = "blue";
    protected $shape = "sphere";
    private $quantity = 10;
    public static $number = 42;
    public static function
    returnHello() {
        return "Hello";
    }
}

// instantiate new object
$object = new Test();

// only color can be accessed from
the instance
echo $object->color; # Works
echo $object->shape; # Fails
echo $object->quantity; # Fails
echo $object->number; # Fails

// we use the static class to access
number
echo Test::$number;
```