Inheritance

A class can inherit methods and properties from its parent class:

class Mother

{

    public function getEyeCount()

    {

        return 2;

    }

}

//inherits mother

class Child extends Mother

{

}

$jimmy = new Child();

//child can call mother's methods

$jimmy->getEyeCount(); //2

//alternative syntax to instantiate child -> (new Child)->getEyeCount();

Besides inheriting from the parent class, a child class can also overwrite the default method of the parent class (see below):

//SHAPES EXEMPLE

class Shape

{

    protected $length = 4;

    public function getArea()

    {

        return pow($this->length, 2);

    }

}

//this class can use the 'default' getArea() method from the Shape parent class

class Square extends Shape

{

}

//for this Triangle class (as the algorithm for area calculation is different, we can overwrite the parent method in the child)

class Triangle extends Shape

{

    protected $base = 4;

    protected $height = 7;

    //this overwrites the getArea from the Square (parent class)

    public function getArea()

    {

        return .5 \* $this->base \* $this->height;

    }

}

//when we use inheritance, if we need to, we can overwrite the functionality of behaviour from the parent class by creating a different method with the same name in the child class

//echo (new Square)->getArea();

echo (new Triangle)->getArea();

\*in this case, as the formula for calculating the area for a triangle is different than the one for a square, the Triangle class overwrites the default method getArea() from the parent with an own getArea() method which implements the right algorithm for triangle area calculation;