Traits and code inclusion

a trait, a new feature that was introduced into PHP version 5.4, as it allows a class to get its code from more than one trait. In fact, it allows a class to use as many traits as it needs.

Traits resemble classes in that they group together code elements under a common name, and with

the following syntax:

trait TraitName {

// The trait's code

}

Example:

trait Price {

public function changePriceByDollars($price, $change)

{

return $price + $change;

}

}

class Bmw {

use Price;

}

class Mercedes {

use Price;

}

$bmw1 = new Bmw();

// Add 3000$

echo $bmw1 -> changePriceByDollars(45000, +3000);

$mercedes1 = new Mercedes();

// Subtract 2100$

echo $mercedes1 -> changePriceByDollars(42000, -2100);

Result:

48000

39900

Is it possible for a class to use more than one trait?

A class can use more than one trait, after all, this is what traits were invented for.

// The first trait

trait Price {

public function changePriceByDollars($price, $change)

{

return $price + $change;

}

}

// The second trait

trait SpecialSell {

public function annonunceSpecialSell ()

{

return \_\_CLASS\_\_ . " on special sell";

}

}

// The Mercedes class uses both traits

class Mercedes {

use Price;

use SpecialSell;

}

$mercedes1 = new Mercedes();

// Subtract 2100$

echo $mercedes1 -> changePriceByDollars(42000, -2100);

echo $mercedes1 -> annonunceSpecialSell();

Result:

39900

Mercedes on on special sell