PROGRAMACIÓN

POO - HERENCIA

Introducción

- La herencia es el proceso de extender una clase existente (clase madre/padre) a una nueva clase (subclase) usando la palabra clave ‘extends’.
- Una subclase hereda todas las propiedades y métodos de su super-clase (principal) excepto las privadas.
- La herencia se usa para la reutilización de códigos y en el polimorfismo.
- PHP no permite herencia múltiple (como máximo una superclase) pero si herencia multinivel.
- Una clase declarada con la palabra clave ‘final’ no se puede extender.

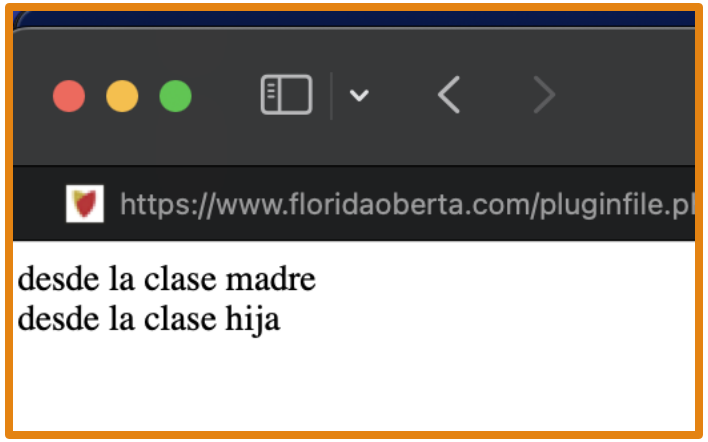
Definición

```php
class SuperClase
{
    protected $var;

    function displaySuper(){
        echo $this->var . " desde la clase madre<br>";
    }
}

class SubClase extends SuperClase
{
    function displaySub(){
        echo $this->var . " desde la clase hija<br>";
    }
}

$v1 = new SubClase();
$v1->displaySuper();
$v1->displaySub();
```



Sobreescritura de métodos

- Es el proceso en el que una subclase redefine un método de clase principal para cambiar su comportamiento.
  - La declaración debe ser exactamente la misma.
  - En caso de que queramos acceder a las funciones de nivel primario desde una subclase, utilizaremos ‘parent::’
- Palabra clave ‘final’
  - Una subclase no puede sobre-escribir un método declarado con la palabra clave ‘final’ en la super-clase.

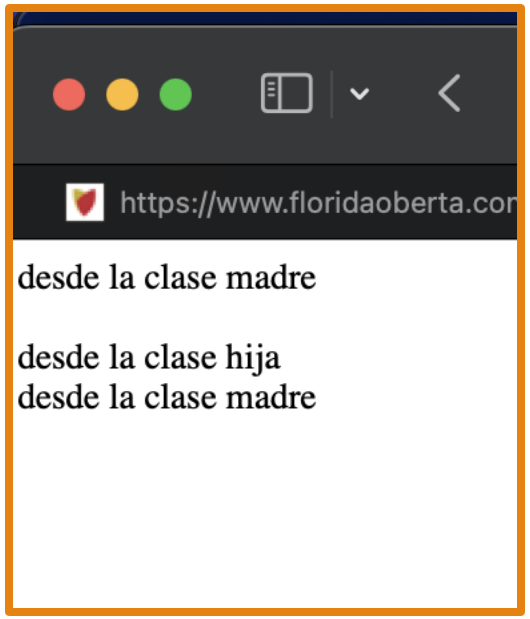
Sobreescritura de métodos

```php
class SuperClase
{
    protected $var;

    function display(){
        echo $this->var . " desde la clase madre<br>";
    }
}

class SubClase extends SuperClase
{
    function display(){
        echo $this->var . " desde la clase hija<br>";
        parent::display();
    }
}

$v1 = new SuperClase();
$v1->display();
echo "<br>";
$v2 = new SubClase();
$v2->display();
```



Constructores

El constructor de una super-clase puede ser llamado con **parent::__construct();**

```php
class Animal
{
    protected $nombre;

    function __construct($name = ""){
        $this->nombre = $name;
    }
}

class Perro extends Animal
{
    private $ladra;
    function __construct($nameDog, $bark = "yes"){
        parent::__construct($nameDog);
        $this->ladra = $bark;
    }
    public function display(){
        echo $this->nombre . "<br>" . $this->ladra;
    }
}

$perrete = new Perro("Boby", "no");
$perrete->display();
```