PROGRAMACIÓN

PROGRAMACIÓN ORIENTADA A OBJETOS (PHP)

Conceptos básicos

- **Clase**

Es un modelo que se utiliza para crear objetos que comparten un mismo comportamiento, estado e identidad.

- **Objeto/instancia**

Es una entidad provista de métodos o mensajes a los cuales responde (comportamiento); atributos con valores concretos (estado); y propiedades (identidad).

- **Método**

Es el algoritmo asociado a un objeto que indica la capacidad de lo que éste puede hacer.

- **Propiedades/atributos**

Son variables que contienen datos asociados a un objeto.

Ejemplo

CLASE coche

- PROPIEDADES
- marca
- color
- puertas
- MÉTODOS
- arrancar
- parar
- reparar

OBJETO coche1

- PROPIEDADES
- ferrari
- verde
- 4
- MÉTODOS
- arrancar
- parar
- reparar

OBJETO coche2

- PROPIEDADES
- lamborgini
- rosa
- 5
- MÉTODOS
- arrancar
- parar
- reparar

Conceptos básicos

- **Abstracción**

Aislación de un elemento de su contexto. Define las características esenciales de un objeto.

- **Encapsulamiento**

Reúne al mismo nivel de abstracción, a todos los elementos que puedan considerarse pertenecientes a una misma entidad.

- **Modularidad**

Característica que permite dividir una aplicación en varias partes más pequeñas (denominadas módulos),independientes unas de otras.

Conceptos básicos

- **Polimorfismo**

Es la capacidad que da a diferentes objetos, la posibilidad de contar con métodos, propiedades y atributos de igual nombre, sin que los de un objeto interfieran con el de otro.

- **Herencia**

Es la relación existente entre dos o más clases, donde una es la principal (madre) y otras son secundarias y dependen (heredan) de ellas (clases “hijas”), donde a la vez, los objetos heredan las características de los objetos de los cuales heredan.

Definición de clases

```php
//definición de la clase
class Coche {
    //propiedades
    public $marca;
    public $color;
    public $puertas;

    //métodos
    public function arrancar(){
        echo "Coche arrancado<br>";
    }
}
```

Definición de objetos

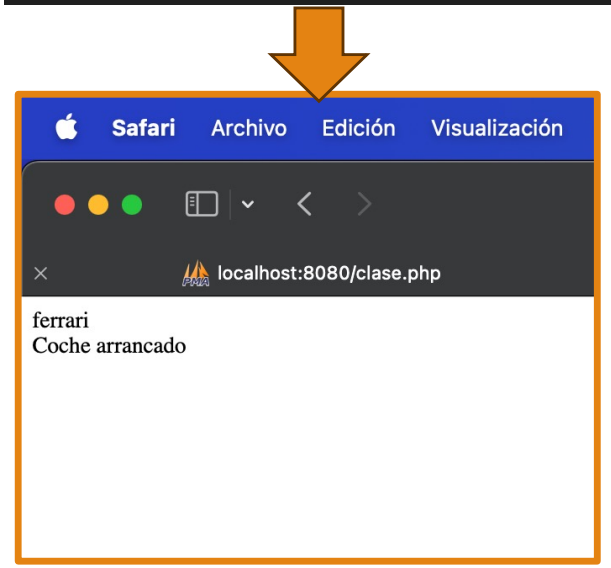
Palabra clave 'new'

Crea un objeto

Operador de objeto ->

Se usa cuando se llama a un método o se accede a una propiedad

```php
//definición de objetos
$coche1 = new Coche();
$coche1->marca = "ferrari";
echo "$coche1->marca <br>";
$coche1->arrancar();
```



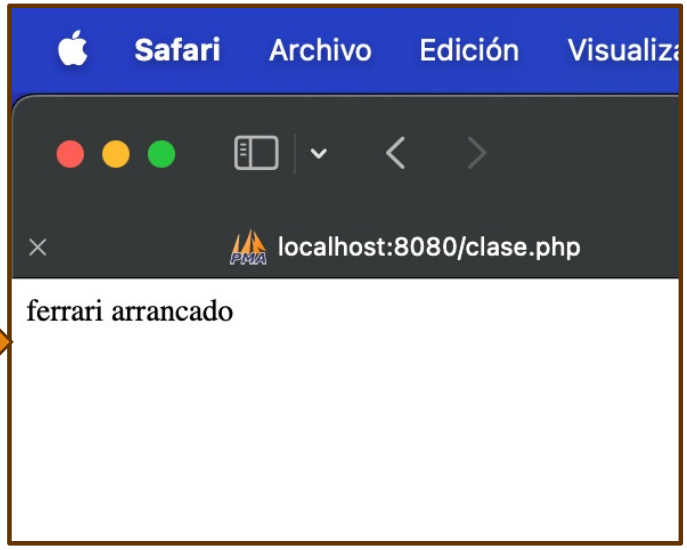
Pseudo-variable $this

Hace referencia al objeto actual, sólo se usa dentro de la clase.

```php
//definición de la clase
class Coche {
    //propiedades
    public $marca;
    public $color;
    public $puertas;

    //métodos
    public function arrancar(){
        echo $this->marca . " arrancado<br>";
    }
}

//definición de objetos
$coche1 = new Coche();
$coche1->marca = "ferrari";
$coche1->arrancar();
```



Visibilidad

- determina cómo se puede acceder a las propiedades o métodos de un objeto:
  - **public**: se puede acceder desde cualquier lugar.
  - **protected**: solo puede acceder la clase y las subclases.
  - **private**: solo puede acceder la clase.
- Una propiedad debe definirse con una de las palabras clave de visibilidad anteriores.
- Un método definido sin ninguno de ellos tendrá visibilidad pública por defecto.

Visibilidad

Por defecto, no deberíamos poder acceder a las propiedades del objeto desde fuera.

Pensemos en lo siguiente: un producto, al cual no se le puede poner un precio por debajo de cierta cantidad.

Visibilidad - problema

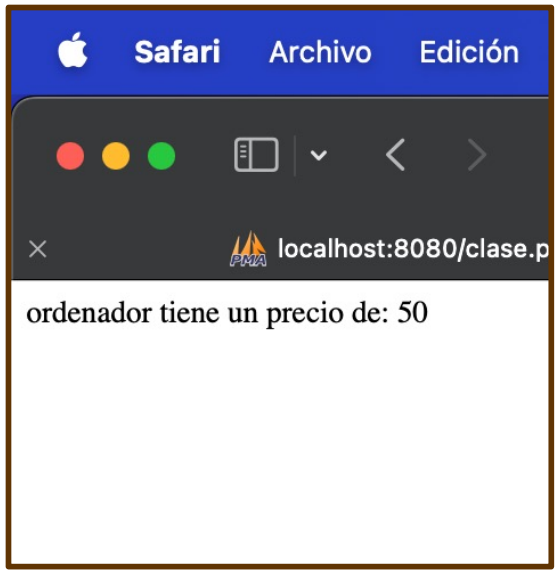
```php
//definición de la clase
class Producto {
    //propiedades
    public $marca;
    public $color;
    public $precio;

    //métodos

}
//definición de objetos
$prod1 = new Producto();
$prod1->marca = "ordenador";
$prod1->precio = 50;

echo $prod1->marca . " tiene un precio de: " . $prod1->precio;
```

Inserto el precio que quiero, aunque la norma sea no inferior a 50€



Visibilidad - solución

```php
//definición de la clase
class Producto {
    //propiedades
    public $marca;
    public $color;
    private $precio;

    //métodos

}
//definición de objetos
$prod1 = new Producto();
$prod1->marca = "ordenador";
$prod1->precio = 50;

echo $prod1->marca . " tiene un precio de: " . $prod1->precio;
```

Declaración privada de una propiedad

Visibilidad – solución (“getters” y “setters)

```php
//definición de la clase
class Producto {
    //propiedades
    public $marca;
    public $color;
    private $precio;

    //métodos
    public function getPrecio(){
        return $this->precio;
    }
    public function setPrecio($euros){
        $this->precio = ($euros>=100) ? $euros : 100;
    }
}
//definición de objetos
$prod1 = new Producto();
$prod1->marca = "ordenador";
$prod1->setPrecio(50);
echo $prod1->marca . " tiene un precio de: " . $prod1->getPrecio();
```