



# Clase 02

## Programación Orientada a Objetos (POO)

[Volver](#)

# Temas:

## Programando del lado del Servidor

- Funciones propias en PHP
- Incluir archivos en PHP
- Clases y objetos

# Funciones propias en PHP

La declaración de una función comienza con la palabra **function**.

El nombre de la función puede empezar con una letra o guión bajo (\_), no con números.

Los nombres de las funciones **NO son case-sensitive**.

```
function NombreFuncion() {  
    // código  
}
```

# Funciones propias en PHP

```
// Las funciones pueden recibir parámetros.  
function NombreFuncion($par_1, $par_2, $par_n) {  
    // código  
}  
  
// Las funciones pueden retornar valores.  
function NombreFuncion() {  
    return $valor;  
}  
  
// Los parámetros pueden tener valores por default.  
function NombreFuncion($par_1, $par_2 = "valor") {  
    // código  
}
```

# Temas:

## Programando del lado del Servidor

- Funciones propias en PHP
- Incluir archivos en PHP
- Clases y objetos

# Incluir archivos en PHP

Las declaraciones `include` / `require` copian todo el código existente del archivo especificado dentro del archivo que posee dicha declaración.

Ambas declaraciones `include` y `require` son idénticas, excepto en caso de falla.

`require` producirá un error fatal (`E_COMPILE_ERROR`) y frenará el script.

`include` sólo producirá una advertencia (`E_WARNING`) y el script continuará.

# Incluir archivos en PHP

## Sintaxis para incluir archivos

```
// Formas de usar include
include "ruta_archivo.php";
include_once "ruta_archivo.php";

// Formas de usar require
require "ruta_archivo.php";
require_once "ruta_archivo.php";
```

Las declaraciones `include_once` / `require_once` sólo incluye/requiere una vez al archivo solicitado.

# Temas:

## Programando del lado del Servidor

- Funciones propias en PHP
- Incluir archivos en PHP
- Clases y objetos



# Clases y objetos

La sintaxis básica para declarar una **Clase** y sus miembros (atributos - métodos) en PHP.

```
class NombreClase
{
    // Atributos (private - protected - public/var - static)
    [Modificadores] $nombreAtributo;
    // Métodos (private - protected - public/var - static)
    [Modificadores] function NombreMetodo([parametros])
    { ... }
}
```

# Clases

```
class NombreClase
{
    //Atributos.
    private $_attr1;
    protected $_attr2;

    //Constructor
    public function __construct() { // código }

    //Métodos.
    private function Func1($param) { //código }
    protected function Func2() { //código }
    public function Func3() { //código }
}
```

# Objetos

La sintaxis básica para declarar un **Objeto** en PHP.

El operador **->** es utilizado para acceder a los **miembros de instancia** de la clase. El operador **::** es utilizado para acceder a los **miembros estáticos** de la clase.

```
//Creo el objeto.  
$nombreObj = new NombreClase();  
//Métodos de instancia. //Atributos de instancia.  
$nombreObj->Func3(); $nombreObj->attr3;  
//Métodos de clase. //Atributos estáticos  
NombreClase::Func4(); NombreClase::$attr4;
```

# Clases - \$this

La sintaxis para acceder a un **atributo** o **método** de una instancia en PHP.

```
class ClaseBase {  
    public function __construct($id,$nombre){  
        //Inicializar variables aquí  
        if($this->validar($id)){  
            $this->id=$id;  
            $this->nombre=$nombre;  
        }  
    }  
    public function validar($id){  
        //realiza una validación  
    }  
}
```

# Clases - Herencia

En PHP se indica herencia a partir de **extends**.

```
class ClaseBase {  
    public function __construct(){  
        //Inicializar variables aquí  
    }  
}  
class ClaseDerivada extends ClaseBase {  
    public function __construct(){  
        parent::__construct();  
        //Inicializar variables propias aquí  
    }  
}
```

# Clases - Polimorfismo

En PHP cualquier método puede ser modificado en sus clases derivadas.

```
class ClaseBase {  
    public function Saludar(){  
        return "Hola";  
    }  
}  
class ClaseDerivada extends ClaseBase {  
    public function Saludar(){  
        return parent::Saludar()." ". "mundo";  
    }  
}
```

# Clases - Interfaces

Las interfaces en PHP sólo pueden contener declaraciones de métodos. Y se implementan con **implements**.

```
interface IInterfaz{  
    function Metodo();  
}  
class Clase implements IInterfaz {  
    public function Metodo(){  
        //Implementación aquí  
    }  
}
```

# Clases - Clases abstractas

Las clases abstractas pueden contener atributos y métodos, pero sólo ellas pueden contener métodos con el modificador **abstract**.

```
abstract class ClaseAbstracta {  
    public abstract function Metodo();  
}  
class ClaseDerivada extends ClaseAbstracta {  
    public function Metodo(){  
        //Implementación aquí  
    }  
}
```



# Clases - Constructores

```
class Coche {  
    // Atributos  
    public $marca;  
    public $color;  
  
    // Constructor con parámetros opcionales  
    public function __construct($marca, $color = "") {  
        $this->marca = $marca;  
        $this->color = $color;  
    }  
}  
$coche = new Coche("FIAT", "NEGRO");  
$coche = new Coche("FORD"); // Opcional 2do parámetro
```

**Dudas ó consultas**