

Curso Básico de PHP

Lección 1: Introducción a PHP

¿Qué es PHP?

PHP es un lenguaje de programación del lado del servidor, utilizado principalmente para crear páginas web dinámicas. PHP significa **Hypertext Preprocessor** y fue diseñado específicamente para el desarrollo web. Su principal característica es que se ejecuta en el servidor antes de que se envíe el contenido al navegador.

Cuando un usuario accede a una página PHP, el servidor ejecuta el código PHP, genera contenido (como HTML, imágenes, o JSON) y lo envía al navegador del cliente.

Características principales de PHP:

- **Lenguaje de servidor:** El código PHP se ejecuta en el servidor y genera HTML que es enviado al cliente.
- **Fácil integración con bases de datos:** PHP tiene soporte integrado para bases de datos como MySQL.
- **Flexibilidad:** Se puede usar tanto en scripts sencillos como en aplicaciones web complejas.

¿Cómo funciona PHP?

1. El servidor web recibe una solicitud para un archivo `.php`.
 2. El servidor ejecuta el archivo PHP en el servidor.
 3. El código PHP genera contenido (como HTML, imágenes, o JSON).
 4. El contenido generado es enviado al navegador del cliente.
-

Lección 2: Instalación y Configuración

Opción 1: Usar XAMPP (Recomendado para principiantes)

1. **Descargar XAMPP:**
 - XAMPP es un paquete que incluye Apache (el servidor web), MySQL (la base de datos) y PHP. Es muy fácil de instalar.
 - Puedes descargar XAMPP desde [su página oficial](#).

2. Instalar XAMPP:

- Sigue los pasos del instalador.
- Una vez instalado, abre el **Panel de control de XAMPP** y enciende Apache y MySQL.

3. Probar PHP:

- Abre tu navegador y ve a <http://localhost>. Si ves la página de inicio de XAMPP, significa que tu servidor está funcionando correctamente.
- Crea un archivo llamado `index.php` dentro de la carpeta `htdocs` de XAMPP (por defecto en `C:\xampp\htdocs`).
- Escribe el siguiente código dentro de `index.php`:

php

Copiar código

```
<?php
    echo "¡Hola, mundo!";
?>
```

4.

- Ahora abre tu navegador y visita <http://localhost/index.php>. Si todo está bien, deberías ver el mensaje "¡Hola, mundo!".

Lección 3: Sintaxis Básica de PHP

1. Etiquetas PHP

Los archivos PHP siempre empiezan con `<?php` y terminan con `?>`. Todo lo que esté dentro de estas etiquetas será procesado como código PHP.

php

Copiar código

```
<?php
    // Esto es un comentario en PHP
    echo "Hola desde PHP"; // Imprime texto en la página
?>
```

2. Variables

Las variables en PHP se definen con el símbolo `$`, y no necesitas declarar su tipo.

php

Copiar código

```
<?php
```

```
    $nombre = "Juan";
```

```
    $edad = 25;
```

```
    echo "Hola, mi nombre es $nombre y tengo $edad años."; //
```

Interpolación de variables

```
?>
```

3. Tipos de datos

PHP tiene varios tipos de datos:

- **Cadenas de texto** (**string**): "Hola Mundo".
- **Enteros** (**int**): 42, -3.
- **Flotantes** (**float**): 3.14, -2.5.
- **Booleanos** (**bool**): **true**, **false**.

Ejemplo:

php

Copiar código

```
<?php
```

```
    $cadena = "Hola";
```

```
    $numero = 42;
```

```
    $decimal = 3.14;
```

```
    $esVerdad = true;
```

```
    echo "$cadena, número: $numero, decimal: $decimal, booleano: $esVerdad";
```

```
?>
```

Lección 4: Estructuras de Control

PHP tiene varias estructuras de control para manejar la lógica condicional y los bucles.

1. Condicionales (if, else, elseif)

php

Copiar código

```
<?php
    $edad = 18;

    if ($edad >= 18) {
        echo "Eres mayor de edad.";
    } else {
        echo "Eres menor de edad.";
    }
?>
```

2. Operadores de comparación

- **==**: Igual.
- **!=**: Diferente.
- **>**: Mayor que.
- **<**: Menor que.
- **>=**: Mayor o igual que.
- **<=**: Menor o igual que.

3. Bucles (for, while, foreach)

- **for**: Se utiliza cuando conoces el número de iteraciones.

php

Copiar código

```
<?php
    for ($i = 0; $i < 5; $i++) {
        echo "Número: $i <br>";
    }
?>
```

- **while**: Ejecuta el bloque de código mientras una condición sea verdadera.

php

Copiar código

```
<?php
    $i = 0;
```

```
while ($i < 5) {  
    echo "Número: $i <br>";  
    $i++;  
}  
?>
```

- **foreach:** Se utiliza para iterar sobre arrays.

php

Copiar código

```
<?php  
$frutas = ["Manzana", "Banano", "Cereza"];  
foreach ($frutas as $fruta) {  
    echo $fruta . "<br>";  
}  
?>
```

Lección 5: Funciones en PHP

Las funciones en PHP se definen con la palabra clave `function`, seguida del nombre de la función y un bloque de código.

Sintaxis básica:

php

Copiar código

```
<?php  
function saludar($nombre) {  
    echo "¡Hola, $nombre!";  
}  
  
saludar("Juan"); // Llama a la función y pasa el argumento "Juan"  
?>
```

Función con valor de retorno:

php

Copiar código

```
<?php
    function suma($a, $b) {
        return $a + $b;
    }

    $resultado = suma(5, 3);
    echo "El resultado es: $resultado";
?>
```

Lección 6: Trabajar con Formularios HTML y PHP

Una de las tareas más comunes con PHP es procesar formularios HTML. Vamos a ver cómo hacerlo.

1. Formulario HTML

html

Copiar código

```
<form method="POST" action="procesar.php">
    Nombre: <input type="text" name="nombre">
    <br>
    Edad: <input type="number" name="edad">
    <br>
    <input type="submit" value="Enviar">
</form>
```

2. Procesar datos con PHP (en **procesar.php**)

php

Copiar código

```
<?php
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        $nombre = $_POST['nombre'];
        $edad = $_POST['edad'];

        echo "Nombre: $nombre<br>";
        echo "Edad: $edad<br>";
    }
```

?>

Lección 7: Conectarse a una Base de Datos con MySQL

PHP se usa mucho para interactuar con bases de datos MySQL. Vamos a hacer una conexión básica.

1. Conectar a MySQL con PDO

php

Copiar código

```
<?php
    $dsn = 'mysql:host=localhost;dbname=mi_base_de_datos';
    $usuario = 'root';
    $contraseña = '';

    try {
        $conexion = new PDO($dsn, $usuario, $contraseña);
        $conexion->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
        echo "Conexión exitosa a la base de datos";
    } catch (PDOException $e) {
        echo "Error de conexión: " . $e->getMessage();
    }
?>
```

2. Realizar una consulta SQL

php

Copiar código

```
<?php
    $stmt = $conexion->query("SELECT * FROM usuarios");
    while ($row = $stmt->fetch(PDO::FETCH_ASSOC)) {
        echo $row['nombre'] . "<br>";
    }
?>
```

Lección 8: Introducción a la Programación Orientada a Objetos (OOP)

La programación orientada a objetos es un paradigma de programación que organiza el código en "objetos" que tienen propiedades y métodos.

Ejemplo básico de una clase en PHP:

php

Copiar código

```
<?php
```

```
    class Persona {
        public $nombre;
        public $edad;

        public function __construct($nombre, $edad) {
            $this->nombre = $nombre;
            $this->edad = $edad;
        }

        public function saludar() {
            echo "Hola, mi nombre es " . $this->nombre . " y tengo " .
$this->edad . " años.";
        }
    }

    $persona = new Persona("Juan", 25);
    $persona->saludar();
?>
```

Conclusión del Curso Básico

¡Felicidades! Has completado el curso básico de PHP. Ahora sabes cómo funciona PHP y cómo utilizarlo para crear páginas web dinámicas, manejar formularios, interactuar con bases de datos y más.

Próximos pasos:

1. **Practica más:** Intenta crear proyectos pequeños como un sistema de registro de usuarios o una aplicación de notas.
2. **Aprende más sobre OOP:** La programación orientada a objetos es fundamental para desarrollar aplicaciones más grandes y estructuradas.
3. **Explora más funciones de PHP:** Como el manejo de sesiones, cookies, seguridad, validación de formularios, etc.

¡Si tienes alguna pregunta o deseas que profundice en algún tema, no dudes en preguntar!