

# Primeros pasos con PHP

---

- [Primeros pasos con PHP](#)
  - [Ejecutar PHP](#)
    - [Consola](#)
    - [Navegador](#)
  - [Sintaxis](#)
    - [Case sensitive](#)
  - [Comentarios](#)
    - [Comentarios de línea](#)
    - [Comentarios de bloque](#)
  - [Imprimir](#)
  - [Primer programa con PHP](#)
  - [Variables y constantes](#)
    - [Variables](#)
    - [Constantes](#)
    - [Imprimir variables y constantes](#)

## Ejecutar PHP

Al igual que en otros lenguajes de programación podemos ejecutar PHP por consola o en un servidor web con PHP instalado, vamos a ver como ejecutar el programa de ambas formas.



**RECOMENDADO** si lo prefieres puedes seguir la [instalación y configuración en PHP.net](#) desde la página oficial de PHP

### Consola

PHP es un lenguaje de scripting que permite ejecutar código fuera de un servidor web.

1. Abre la consola o terminal de tu sistema operativo.
2. Navega hasta el directorio donde se encuentra el archivo PHP que deseas ejecutar. Puedes usar el comando `cd` para cambiar de directorio.
3. Una vez en el directorio correcto, puedes ejecutar el script PHP usando el siguiente comando:

```
php nombre_del_archivo.php
```



**ADVERTENCIA** Recuerda que en la consola, no tienes acceso a las superglobales de PHP relacionadas con el servidor web, como `$_SERVER`, `$_POST`, `$_GET`, etc. Sin embargo, puedes utilizar todas las demás características y funciones de PHP en tu script de línea de comandos.

### Navegador

En cambio la forma más utilizada para ejecutar PHP es desde el navegador ya que nos permite acceder a todas las características de PHP.

1. El primer paso sería comprobar que tenemos instalado el servidor XAMPP

:back: **PARA ENTRAR EN CONTEXTO** si aún no has instalado XAMPP puedes volver a la sección [tutorial de instalación](#) para instalar y configurar PHP

2. Comprobar que el servidor web (Apache) esté en estado "running"
3. Comprobar que puertos se han asignado al servidor. Por defecto los puertos del servidor web son 80 y 443.
4. Copiar los archivos de nuestro programa en la siguiente ruta
  1. Linux (/opt/lampp/htdocs/)
  2. Windows (C:\xampp\htdocs\)
  3. MacOS (/Applications/XAMPP/htdocs)



**RECOMENDADO** es una buena práctica crear una carpeta dentro de htdocs por cada proyecto.

5. Abrir una ventana del navegador y acceder a [http://localhost/tu\\_fichero.php](http://localhost/tu_fichero.php) o <http://localhost/tuproyecto/tufichero.php>



**NOTA** si los puertos asignados a tu servidor web Apache no son 80 y 443 tienes que especificarlos en el navegador de la siguiente forma:

<http://localhost:puerto/tufichero.php>. Ejemplo: <http://localhost:8080/index.php>

## Sintaxis

Los script de PHP comienzan siempre con `<?php` y terminan con `?>`. Estos caracteres marcan que el código que se encuentre dentro de ellos será ejecutado como código PHP independientemente de si se encuentra en un fichero con otro tipo de código.

```
<?php
//código que queremos ejecutar
?>
```

Siempre que un fichero contenga código PHP deberá llevar la extensión de fichero `.php`. Por ejemplo, si tenemos un fichero `mipagina.html` que contiene código HTML y código PHP, ese fichero deberá llevar la extensión `.php` y no `.html`, quedaría como `mipagina.php`

Todas las instrucciones que acabemos en PHP tienen que estar cerradas con `;` si no el programa fallará por que no encontrará donde termina una instrucción

### Case sensitive

En PHP podemos escribir las palabras reservadas (if, switch, echo, etc) con mayúsculas, minúsculas o alternando, ya que reconoce que es una palabra reservada independientemente de como esté escrito.

Aunque no influya en el funcionamiento del programa es recomendable, siguiendo las buenas prácticas, escribir las palabras reservadas en minúsculas

:woman\_teacher: **EXPLICACIÓN** case sensitive

```
<?php
    echo "Hola Mundo"; //lo recomendado
    ECHO "Hola Mundo"; //es mejor que no lo hagas así ...
    EcHo "Hola Mundo"; //horrible!!
?>
```

## Comentarios

Los comentarios son líneas que no se ejecutan como parte del programa. Son un punto muy importante en todos los lenguajes de programación ya que nos ayuda a dejar notas sobre el funcionamiento de nuestro código y que, en el futuro, tanto nosotros como otros programadores entiendan más fácilmente que hace el código.

PHP permite establecer comentarios de línea y comentarios de bloque:

### Comentarios de línea

Estos comentarios solo se aplican a la línea en la que están declarados los caracteres `//` o `#`

```
<?php
    // esto es un comentario de línea
    # otra forma de establecer comentarios de línea
?>
```

### Comentarios de bloque

Estos comentarios se aplican a todo lo que esté dentro del caracter de apertura `/*` y el caracter de cierre `*/`

También son útiles para comentar troxos de código dentro de una instrucción

```
<?php
    /*
    todo este
    texto
    va a ser
    ignorado
    */

    echo (1 /*+ 2*/ + 3);
?>
```

## Imprimir

En todos los lenguajes de programación, en algún momento, necesitamos imprimir texto, variables, funciones, etc. PHP no iba a ser menos y, además, tenemos dos formas de imprimir.

`echo` y `print` son palabras que se usan para imprimir. Son casi lo mismo y podemos utilizar la que más nos guste pero hay que aclarar la única diferencia que tienen: `echo` no devuelve ningún valor pero `print` devuelve 1 por lo que se puede utilizar en expresiones.

Ambas pueden utilizarse con `()` al final para añadir parámetros o sin paréntesis. `echo` o `echo()` y `print` o `print()`

```
<?php
//ejemplos de echo y print

echo "Imprimo con echo"
echo ("Imprimo con echo")

print "Imprimo con print"
print ("Imprimo con print")

?>
```

## Primer programa con PHP

Una vez que conocemos lo básico sobre PHP ya estamos preparados para hacer nuestro primer programa 'Hola Mundo'.

Lo primero que vamos a hacer es crear un fichero `hola.php` y vamos a imprimir el mensaje `Hola Mundo` típico.

```
<?php
//primer programa para aprender a imprimir texto
echo "Hola Mundo"

?>
```

Ahora simplemente vamos a comprobar el resultado en el navegador

 imagen de la ejecucion hola mundo de php

:back: **PARA ENTRAR EN CONTEXTO** si no sabes como ejecutar PHP puedes echar un vistazo a la sección anterior donde vemos como configurar nuestro editor de código y navegador. [Instalación, configuración y ejecución de PHP](#)

## Variables y constantes

EL uso de variables y constantes es una parte fundamental de los lenguajes de programación, nos permiten almacenar datos y utilizarlos en el código y también variar su valor en base a la ejecución.

PHP es un lenguaje de tipado débil por lo que no es necesario definir que tipo de dato establecemos en las variables

✓ **RECOMENDADO** echa un vistazo a la sección de [Lenguajes de programación: Tipados](#) ✓

**RECOMENDADO** echa un vistazo a la sección de [Scope de las variables](#)

## Variables

Las variables son contenedores de valores que pueden cambiar su valor según definamos en la ejecución del código. En PHP se declaran utilizando el carácter `$` y seguido el nombre de la variable: `$variable`

✓ **RECOMENDADO** echa un vistazo a la sección de [Buenas prácticas de programación: variables y constantes](#)

```
<?php

$variable1 = "Hola mundo";
$var2 = 6;
$v3 = "probando variables";

?>
```

## Constantes

Las constantes también son contenedores de valores pero, a diferencia de las variables, no cambian su valor durante la ejecución si no que se establece un valor y será el mismo durante todo el programa.

Se declaran utilizando `define()` y tenemos que indicar un nombre de constante, el valor y si queremos que sea 'case-insensitive' que por defecto es 'false'

Sintaxis: `define (nombre_constante, valor, case-insensitive)`

```
// no es necesario establecer 'false' por que está establecido por defecto
define (primeraconstante, 'he desactivado case-insensitive')
define (primeraconstante, 'he activado case-insensitive', true)
```

## Imprimir variables y constantes

```
<?php

echo $variable1 //resultado: Hola Mundo
echo $var2 //resultado: 6
echo $v3 //resultado: probando variables

echo primeraconstante //case-insensitive deshabilitado //resultado: he
desactivado case-insensitive
echo PRIMERACONSTANTE //case-insensitive habilitado //resultado: he
habilitado case-insensitive

?>
```