

Programación

PHP

Vigésimo segunda semana

Abril 2022

Cruz García, Iago



[Introducción](#)

[PHP](#)

[Instalación XAMPP](#)

[Primera prueba de PHP](#)

[Variables](#)

[Condicionales](#)

[Bucles](#)

Introducción

PHP es un lenguaje de programación mediante scripts interpretado, por lo que necesita ser compilado en tiempo real, como JavaScript, y por lo tanto permite ser ejecutado en navegadores y embebido en HTML.

Veremos este tema como trabajar con PHP de forma aislada y en el siguiente cómo realizar conexiones con HTML y JavaScript.

Recomendación: A partir de ahora es necesario que a la hora de realizar ejercicios nos acostumbremos a buscar información en la documentación oficial. En caso de no poder resolver las dudas con la documentación ofrecida, el siguiente paso será preguntar las dudas en el foro de clase.

[Documentación oficial de JavaScript](#)

[Foro de la asignatura](#)

[XAMPP para utilizar PHP](#)

[Página oficial de PHP](#)

PHP

PHP requiere un servidor para trabajar. Como no hemos visto como establecer servidores desde 0 ni la arquitectura cliente-servidor, vamos a utilizar un entorno muy popular conocido como XAMPP (Windows Apache MariaDB PHP PERL) que nos permite desplegar rápidamente un servidor con múltiples funcionalidades. Entre ellas, las conexiones a base de datos con MariaDB y el uso de PHP.

Instalación XAMPP

Vamos a comenzar instalando XAMPP accediendo a la página oficial y descargando la versión de Windows.



Página oficial de XAMPP (Fuente: Elaboración propia)

Descargamos la versión 8.1.4.



XAMPP for Windows 7.4.28, 8.0.17 & 8.1.4

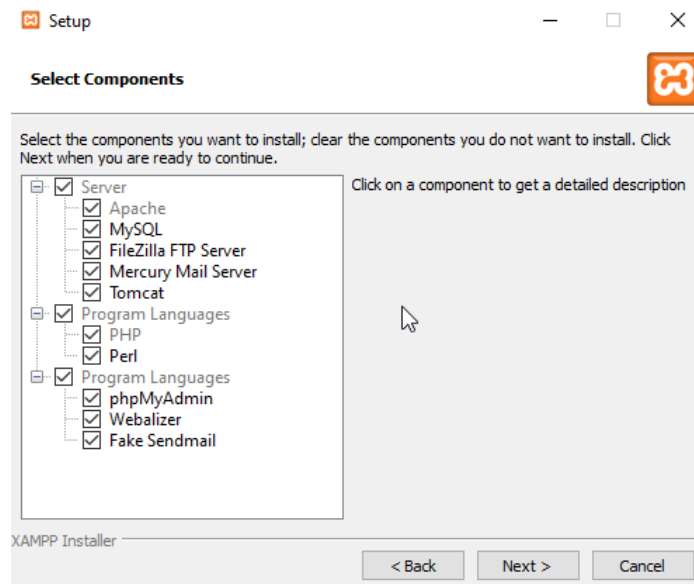
Version	Checksum	Size
7.4.28 / PHP 7.4.28 What's Included?	md5 sha1	Download (64 bit) 159 Mb
8.0.17 / PHP 8.0.17 What's Included?	md5 sha1	Download (64 bit) 160 Mb
8.1.4 / PHP 8.1.4 What's Included?	md5 sha1	Download (64 bit) 164 Mb

[Requirements](#) [Add-ons](#) [More Downloads »](#)

Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms [here](#).

Pantalla de descarga de XAMPP (Fuente: Elaboración propia)

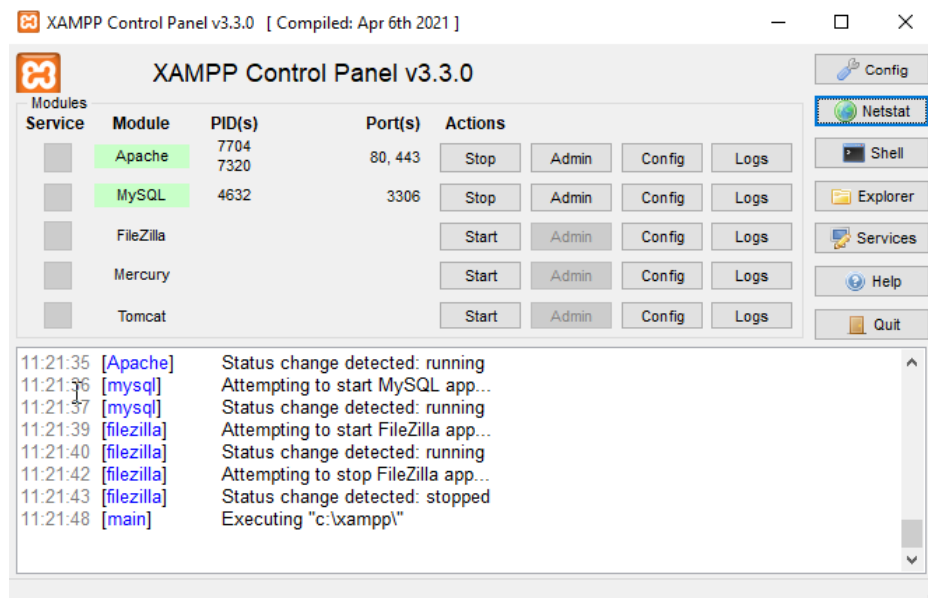
Una vez descargado, ejecutamos el fichero de instalación. El proceso es muy sencillo, dejando por defecto los valores de las aplicaciones a instalar. Podemos seleccionar otra ruta para instalar la aplicación o seleccionar el idioma entre inglés y alemán.



Selección de componentes (Fuente: Elaboración propia)

En la instalación nos saldrá la opción de instalar Bitnami, que permite instalar aplicaciones a mayores sobre XAMPP. Desactivamos esta opción ya que no la necesitaremos en estos momentos.

Una vez termine la instalación, podremos ejecutar XAMPP. Ejecutaremos los servicios de MySQL y de Apache, los otros podemos dejarlos dormidos, presionando en el botón de "Start". Una vez hecho esto, podremos acceder en nuestro navegador a la ruta "localhost" e iniciará la página de prueba de XAMPP.



Panel de control de XAMPP (Fuente: Elaboración propia)

Primera prueba de PHP

Vamos a crear nuestro primer archivo PHP y comprobar que funciona. Para ello, localizaremos la carpeta “htdocs” en la instalación pulsando la opción “Explorer” en la aplicación. Crearemos un fichero comprimido de esta carpeta para tener una copia de seguridad.

Borramos el contenido de la carpeta y creamos en su lugar un fichero que se llame “index.php”. Dentro, escribiremos el siguiente código:

index.php

```
<html>
<body>
<?php
    echo "Hola mundo";
?>
</body>
</html>
```

Si volvemos a cargar "localhost" en el navegador, el texto "Hola mundo" debería ser visible.

Podemos observar que el código php está encerrado entre '<?php "codigo" ?>', terminando cada sentencia con ";" punto y coma obligatoriamente.

Vamos a ver cómo trasladar las estructuras básicas de programación a PHP: variables, condicionales y bucles.

Salida

Para realizar salida de datos por pantalla se pueden utilizar dos métodos: *echo* y *print*. La ventaja de PHP es que al hacer los métodos de salida, PHP interpreta las etiquetas de HTML, por lo que si, por ejemplo, dentro de *echo* escribimos `<h1>Título grande</h1>` creará la etiqueta correspondiente:

index.php

```
<html>
<body>
```



```
<?php
    echo "<h1>Hola mundo</h1>";
?>
</body>
</html>
```

Variables

Las variables en PHP, al igual que en JavaScript, son de tipado dinámico por lo que no hace falta atribuirles un tipo de dato. Para escribir variables utilizaremos el símbolo '\$' dolar seguido del nombre de la variable. Para usarla, usaremos el mismo símbolo y nombre:

index.php

```
<html>
<body>
    <?php
        $valor = "mundo";
        echo "Hola $valor";
    ?>
</body>
</html>
```

Los array se declaran de forma parecida, incluyendo la cláusula "array".

index.php

```
<html>
<body>
```

```
<?php
    $valor = array("Manzana", "Pera", "Plátano");
    echo "Hola $valor[0]";
?>
</body>
</html>
```

Condicionales

Al igual que en JavaScript y en otros lenguajes, existen las sentencias "if-else if-else" y "switch". En caso de PHP, las sentencias son "if", "elseif" y "else", y además la estructura de "switch case" es la misma. Los comparadores lógicos son los mismos que en JavaScript:

index.php

```
<html>
<body>
    <?php
        $valor = "mundo";
        $cuenta = 3;
        if($cuenta == 2){
            echo "Hola $valor";
        }else if($cuenta == 1){
            echo "Adiós $valor";
        }else{
            echo "Este $valor";
        }
    ?>
</body>
</html>
```

Bucles

Los bucles que vimos hasta ahora “while”, “do while”, “for” y “foreach” también existen en PHP. Los bucles “while”, “do while” y “for” son idénticos que en JavaScript mientras que el bucle “foreach” cambia completamente.

index.php

```
<html>
<body>
  <?php
    $valor = "mundo";
    for($i = 0; $i < 10; $i++){
      echo "Hola $valor";
    }
  ?>
</body>
</html>
```

El bucle foreach sin embargo se transforma en:

index.php

```
<html>
<body>
  <?php
    $valor = array("Manzana","Plátano","Pera");
    foreach ($valor as $fruta) {
      echo "Hola $fruta";
    }
  ?>
</body>
</html>
```

