

Tema 0: Primeros conceptos

Tema 0: Primeros conceptos

1. Programación web: lenguajes del lado del servidor y del lado cliente
 - Programación del lado del servidor
 - Programación del lado del cliente
2. Diferencias entre compilado e interpretado
3. Diferencias entre HTML y XHTML
4. PHP: Introducción
 - Ventajas de PHP
 - ¿Qué necesitamos?

1. Programación web: lenguajes del lado del servidor y del lado cliente

¿Cómo funciona la web?

La web funciona según un principio realmente sencillo: los servidores web albergan los contenidos y los clientes los solicitan por HTTP (HTTPS) o FTP, entendiendo por clientes a los **navegadores** (Mozilla Firefox o Google Chrome entre otros) instalados en el sistema operativo del usuario (en local), donde se ejecutan.

Por el contrario, los **servidores web**, como Apache o NGINX, forman parte de los proyectos de desarrollo web, se instalan y se ejecutan en este entorno y permiten a los clientes el acceso a los contenidos.

Los contenidos estáticos, como los clásicos elementos de HTML o las imágenes, sencillamente se envían al navegador y allí se visualizan, los **contenidos dinámicos**, como una Wiki, un menú desplegable o cualquier tipo de aplicación web solo funcionan **por medio de scripts**, los cuales se han de ejecutar e interpretar con el **lenguaje de programación web** correspondiente en el lado del servidor (PHP) o en el lado del cliente (JavaScript). Es por esto que se diferencia fundamentalmente entre los lenguajes de programación del lado servidor y los lenguajes del lado cliente.

Programación del lado del servidor

Su principal uso se centra en el desarrollo de aplicaciones web con elementos dinámicos. Esta tecnología de desarrollo web se basa en la utilización de **scripts que ejecuta el servidor web, con ayuda del lenguaje de programación adecuado**, cuando un cliente solicita el contenido. Una tarea frecuente de los scripts consiste en extraer los datos que se necesitan de una base de datos y de integrarlos en el proyecto web. Aunque el usuario accede al proyecto a través de páginas HTML, el código fuente de los scripts permanece oculto.

En los primeros tiempos de la red de redes, la programación del lado servidor se reducía casi exclusivamente al desarrollo de programas en lenguajes C, Perl y de líneas de comando. Estas aplicaciones eran ejecutadas e interpretadas por el **sistema operativo del servidor** y el resultado podía ser transmitido desde el servidor web al navegador que había hecho la petición a través de una interfaz de entrada común. Hoy en día, muchos servidores web pueden ejecutar directamente los scripts, por ejemplo, con ayuda de módulos (XAMPP). El lenguaje de programación del lado servidor más utilizado en la actualidad es [PHP](#) muy similar a C y a Perl.

Programación del lado del cliente

Los lenguajes de programación del lado cliente también se utilizan para realizar proyectos con contenidos dinámicos pero a diferencia de los lenguajes del lado servidor, no es el servidor el que **ejecuta y procesa los scripts, sino el cliente solicitante**. Con esta finalidad, los scripts se incluyen en el documento HTML o XHTML o se escribe en un archivo separado que se enlaza al documento principal.

Cuando un usuario solicita una página web o una aplicación web con un script de este tipo, el servidor web envía el documento HTML y el script al navegador, quien lo ejecuta y presenta el resultado final. Al ejecutarse en el navegador, el usuario puede ver el código fuente, a diferencia de lo que ocurre con los scripts del lado servidor. (ejemplo Javascript).

En teoría, cualquier lenguaje de programación web cabría utilizarse para programar del lado cliente, aunque ello implicaría que los desarrolladores de todos los navegadores más importantes deberían ponerse de acuerdo en soportarlos.

2. Diferencias entre compilado e interpretado

La principal diferencia entre un lenguaje compilado y uno interpretado es que el **lenguaje compilado** requiere un paso adicional antes de ser ejecutado, la compilación, que convierte el código que escribes a **lenguaje de máquina**.

Un **lenguaje interpretado**, por otro lado, es convertido a **lenguaje de máquina** a medida que es ejecutado.

Tanto **compiladores** como **interpretadores** son programas que convierten el código que escribes a **lenguaje de máquina**.

A todos estos lenguajes se les conoce como **lenguajes de alto nivel**.

Lenguajes de bajo nivel, son aquellos más cercanos al hardware (C)

3. Diferencias entre HTML y XHTML

HTML y XHTML son las dos variantes que existen en el lenguaje HTML 5. Las diferencias entre esas dos variantes son básicamente sintácticas: la sintaxis HTML está inspirada en la norma SGML (aunque no la cumple estrictamente), mientras que la sintaxis XHTML está basada en la recomendación XML (aunque tampoco la cumple estrictamente).

En general, la sintaxis XHTML es más "estricta", en el sentido de imponer restricciones en la forma de escribir etiquetas, atributos o valores, mientras que la sintaxis HTML es más "flexible".

[<http://www.mcilibre.org/consultar/htmlcss/html/html-xhtml-diferencias.html>]

4. PHP: Introducción

PHP ((Personal Home Page) es un lenguaje de programación multiplataforma.

A diferencia de JavaScript o HTML que se ejecutan en el navegador, PHP se **ejecuta en el servidor**, por eso nos permite acceder a los recursos que tiene el servidor, por ejemplo a una base de datos.

Los programas escritos en PHP se ejecutan en el servidor (no dentro del navegador en tu máquina local) y el resultado se envía al navegador. El resultado es normalmente una página HTML.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que el navegador lo soporte, es independiente del navegador, pero sin embargo para que las páginas PHP funcionen, **el servidor donde están alojadas debe soportar PHP**. Hoy en día la práctica totalidad de servidores que ofrecen servicios de hosting soportan PHP por defecto (por ejemplo Apache).

PHP es un lenguaje de **programación estructurado** y, como tal, hace uso de variables, sentencias condicionales, bucles, funciones... PHP es también un **lenguaje de programación orientado a objetos** y, por consiguiente, permite definir clases con sus métodos correspondientes y crear instancias de esas clases.

Multiplataforma significa que los scripts PHP pueden ejecutarse en muchos sistemas operativos y servidores web diferentes. PHP está disponible para las dos configuraciones de servidor web más populares, IIS (Windows) y Apache.

El lenguaje de secuencias de comandos incrustado HTML significa que las instrucciones y los comandos PHP están realmente incrustados en sus documentos HTML. Cuando el servidor web ve las declaraciones PHP en la página web, ejecuta las declaraciones y envía el resultado resultante junto con el resto del HTML.

Ventajas de PHP

- PHP es un software de código abierto.
- PHP no cuesta nada, es gratis para descargar y usar.
- PHP es un lenguaje del lado del servidor y se utiliza para sitios web y aplicaciones web.
- Los scripts PHP se ejecutan en el servidor.
- PHP admite una amplia gama de bases de datos.
- PHP se ejecuta en varias plataformas como Linux, Windows, Unix, etc.

- PHP es compatible con la mayoría de los servidores web (por ejemplo, Apache, IIS(windows)).
- PHP conversa con varios protocolos de red.

¿Qué necesitamos?

Los programas necesarios para trabajar en esta clase con PHP son los siguientes:

1. Un navegador web.
2. Un editor de texto.
3. El lenguaje PHP.
4. Un **servidor** web (p. ej. Apache).
5. El **módulo** de PHP para el servidor (para que el Apache pueda ejecutar el PHP).
6. Un **gestor de bases de datos** como por ejemplo MySQL.

Afortunadamente hay paquetes que ya incluyen un entorno que integran todo lo que necesitamos para trabajar, y además disponen de los llamados instaladores automáticos.

Haremos que nuestro ordenador actúe como servidor web. A esta forma de trabajo se le llama coloquialmente, desarrollo en local.

Para poder trabajar en la asignatura de HLC vamos a usar uno de los entornos de desarrollo más populares de PHP, **XAMPP**.

<https://www.apachefriends.org/es/index.html>