PROGRAMACIÓN WEB EN ENTORNO SERVIDOR UD-8

VALIDACIÓN

ÍNDICE

[VARIABLES DE SERVIDOR 3](#_Toc147867406)

[A. $\_SERVER[] 3](#_Toc147867407)

[VALIDACIÓN DE DATOS 4](#_Toc147867408)

[TEXTO IMPRIMIBLE 4](#_Toc147867409)

[INTRODUCCIÓN DE <br /> 4](#_Toc147867410)

[FORMATEO SALIDA CON PRINTF 5](#_Toc147867411)

[FORMATEO SALIDA CON NUMBER\_FORMAT 5](#_Toc147867412)

[FUNCIONES Y CONSTANTES MATEMÁTICAS 6](#_Toc147867413)

# VARIABLES DE SERVIDOR

PHP almacena la información del servidor y de las peticiones HTTP en seis arrays globales:

* $\_ENV: información sobre las variables de entorno.
* $\_GET: parámetros enviados en la petición GET.
* $\_POST: parámetros enviados en el envío POST.
* $\_REQUEST: parámetros enviados con GET o POST.
* $\_COOKIE: contiene las cookies de la petición, las claves del array son los nombres de las cookies.
* $\_SERVER: información sobre el servidor.
* $\_FILES: información sobre los ficheros cargados via upload.

## $\_SERVER[]

Si nos centramos en el array $\_SERVER podemos consultar las siguientes propiedades:

* PHP\_SELF: nombre del script ejecutado, relativo al document root (p.ej: /tienda/carrito.php)
* SERVER\_SOFTWARE: (p.ej: Apache)
* SERVER\_NAME: dominio, alias DNS (p.ej: www.elche.es)
* REQUEST\_METHOD: GET
* REQUEST\_URI: URI, sin el dominio
* QUERY\_STRING: todo lo que va después de ? en la URL (p.ej: nombre=Pepe&apellido=Perez)
* PATH\_INFO: ruta extra tras la petición. Si la URL es http://www.php.com/php/pathInfo.php/algo/cosa?foo=bar, entonces
* $\_SERVER['PATH\_INFO'] será /algo/cosa.
* REMOTE\_HOST: hostname que hizo la petición
* REMOTE\_ADDR: IP del cliente
* AUTH\_TYPE: tipo de autenticación (p.ej: Basic)
* REMOTE\_USER: nombre del usuario autenticado

Apache crea una clave para cada cabecera HTTP, en mayúsculas y sustituyendo los guiones por subrayados:

* HTTP\_USER\_AGENT: agente (navegador)
* HTTP\_REFERER: página desde la que se hizo la petición

# VALIDACIÓN DE DATOS

Para validar los datos en el servidor, PHP nos proporciona una función llamada filter\_var(dato,constante\_validación[,modificador]), que devuelve true o false según el dato cumpla con la constante de validación.

Podemos validar un conjunto de datos con filter\_var\_array

Los posibles valores de la constante de validación son:

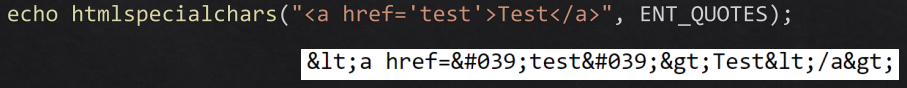
* FILTER\_VALIDATE\_BOOLEAN
* FILTER\_VALIDATE\_INT
* FILTER\_VALIDATE\_FLOAT
* FILTER\_VALIDATE\_DOMAIN
* FILTER\_VALIDATE\_EMAIL
* FILTER\_VALIDATE\_IP
* FILTER\_VALIDATE\_MAC
* FILTER\_VALIDATE\_URL

Se pueden consultar en el manual los modificadores.

# TEXTO IMPRIMIBLE

Si se desea imprimir un texto, independientemente del contenido que tenga (da igual que tenga formato en html, por ejemplo), debemos usar la función htmlspecialchars(cadena[, formato[, cofificación]]).

Con el formato puedo especificar por ejemplo si deseo dejar solamente las comillas simples… y con la codificación indico el juego de caracteres a utilizar, por ejemplo UTF-8.



# INTRODUCCIÓN DE <br />

Otra función que puede ser útil es introducir automáticamente <br/> cada vez que encuentre saltos de línea. Esto se consigue con la función nl2br.

Esta función la podríamos cambiar con un str\_replace( "\n", '<br/>', $Text );

Texto

Descripción generada automáticamente

# FORMATEO SALIDA CON PRINTF

La función printf (cadena con comodines, tipos de datos separados por comas), presente en otros muchos lenguajes como java, nos permite a través de modificadores dar un formato específico a las variables.

Posibles valores de formato:

* %% - Devuelve un signo de porcentaje
* % B - número binario
* % C - El carácter de acuerdo con el valor ASCII
* % d - Firmado número decimal (negative, zero or positive)
* % e - La notación científica usando una minúscula (eg 1.2e+2)
* % E - La notación científica usando una mayúscula (eg 1.2E+2)
* % u - número decimal sin signo (equal to or greather than zero)
* % f - número de coma flotante (local settings aware)
* % F - número de coma flotante (no conscientes configuración local)
* % G - más corto de %e y% f
* % G - más corto de %E y% f
* % D - número octal
* % S - Cadena
* % x - número hexadecimal (lowercase letters)
* % X - número hexadecimal (uppercase letters)

# FORMATEO SALIDA CON NUMBER\_FORMAT

La función number\_format(valor, num\_decimales, separador\_decimal, separador\_miles) devuelve un string con el número correctamente formateado.

Su ventaja radica en ser más fácil de formatear que con printf.

Texto

Descripción generada automáticamente

# FUNCIONES Y CONSTANTES MATEMÁTICAS

Disponemos tanto de constantes como funciones ya definidas para trabajar con operaciones matemáticas: https://www.php.net/manual/es/ref.math.php

* Constantes ya definidas
  + M\_PI, M\_E, M\_EULER, M\_LN2, M\_LOG2E
  + PHP\_INT\_MAX, PHP\_FLOAT\_MAX
* Funciones de cálculo
  + pow, sqrt, log, decbin, bindec, decoct, dechex, base\_convert, max, min
* Funciones trigonométricas
  + sin, cos, tan, deg2rad, rad2deg
* Funciones para trabajar con números aleatorios
  + rand, mt\_rand (más rápida)

Para redondear, tenemos abs para el valor absoluto y round para redondear, ceil para aproximación por exceso y floor por defecto.

Texto

Descripción generada automáticamente