

Programación de documentos Web utilizando lenguajes de «script» de servidor (Implantación de aplicaciones web)

Clasificación

Los primeros servidores web permitían trabajar con información estática, es decir, la información permanecía invariable hasta que volvíamos hacer una petición al servidor. La primera solución técnica que se realizó fue utilizar aplicaciones que estaban instaladas en el propio servidor, esta tecnología, conocida como Common Gateway Interface (CGI) permitía lanzar programas escritos en C o Perl para refrescar la información y que fuera un poco más dinámica.

El siguiente paso fue la creación de lenguajes en el lado del servidor como PHP (PHP Hypertext Pre-processor) de Software libre, ASP (Active Server Pages) de Microsoft, JSP (JavaServer Pages) de JAVA, estos lenguajes se utilizan para generar páginas dinámicas, facilitando el acceso a las bases de datos. Para poder ejecutar programas en PHP, ASP o JSP es necesario descargarse aplicaciones que permiten ejecutar las extensiones de dichos lenguajes.

Si nuestro servidor soporta ejecución en PHP bastará con escribir el código en un editor de textos, guardarlo con la extensión .php en la carpeta del servidor donde podemos ejecutar PHP y accedemos al fichero escribiendo la dirección de la página en el explorador.

Integración con los lenguajes de marcas

Los lenguajes de marcas sirven para codificar documentos, la codificación se realiza incorporando etiquetas o marcas que tienen información adicional acerca de la estructura del texto.

El lenguaje de marcas más popular es el HTML, siglas de HyperText Markup Language (Lenguaje para el Formato de Documentos de Hipertexto), es el lenguaje usado por la World Wide Web. Otros ejemplos de lenguajes de marcas son XHTML (Extensible Hypertext Markup Language), RSS 2.0 (Really Simple Syndication), RTF (Rich Text Format), etc.

HTML es el lenguaje utilizado para construir páginas web, mediante una estructura se crea contenido en forma de texto, también se pueden incorporar otro tipo de elementos a ese contenido como imágenes, sonidos.

Los lenguajes de script de servidor como PHP generan HTML que puede ser utilizado junto con las hojas de estilo CSS.

Vamos a continuar explicando características del lenguaje PHP.

Sintaxis

Cuando PHP analiza un fichero, busca las etiquetas de apertura y cierre, que son `<?php` y `?>`, y que indican a PHP dónde empezar y finalizar la interpretación del código. Este mecanismo permite embeber a PHP en todo tipo de documentos, ya que todo lo que esté fuera de las etiquetas de apertura y cierre de PHP será ignorado por el analizador.

Herramientas de edición de código

Hay múltiples editores de código disponibles para programar en

PHP.

Elementos del lenguaje

- Tipos de datos básicos
- Operadores
- Estructuras de control
- Vectores y tablas
- Funciones
- Formularios y controles
- Clases
- Excepciones

Comentarios

En cualquier código de script de PHP se pueden poner comentarios (líneas de código que no se ejecutan). En PHP hay dos formas de escribir comentarios: En una sola línea y en múltiples líneas.

Funciones integradas y de usuario

Una función es una serie de instrucciones que tienen un determinado propósito y que se invocan mediante un nombre.

Librerías de funciones

En el caso de que la función sea de gran interés, sería conveniente definirlas en un archivo, por ejemplo, libreria.php, e incluirlo en cada script con la instrucción `include_once` o `require_once`, que ya conocemos.

Funciones recursivas

Una función se dice que es recursiva si dentro del código de dicha función se invoca a la misma función una o más veces.

Funciones para el manejo de variables

- Comprobación si una variable está vacía
 - `empty` – Determina si una variable está vacía
- Conversión a tipos básicos
 - `doubleval` – Alias de `floatval`
 - `floatval` – Obtener el valor flotante de una variable
 - `intval` – Obtiene el valor entero de una variable
 - `strval` – Obtiene el valor de cadena de una variable
- Conversión a tipos
 - `settype` – Establece el tipo de una variable
 - `gettype` – Obtener el tipo de una variable
- Comprobación de tipos
 - `is_array` – Comprueba si una variable es un array
 - `is_bool` – Comprueba si una variable es de tipo booleano
 - `is_callable` – Verificar que los contenidos de una variable puedan ser llamados como una función
 - `is_double` – Alias de `is_float`
 - `is_float` – Comprueba si el tipo de una variable es float
 - `is_int` – Comprueba si el tipo de una variable es integer
 - `is_integer` – Alias de `is_int`
 - `is_long` – Alias de `is_int`
 - `is_null` – Comprueba si una variable es NULL
 - `is_numeric` – Comprueba si una variable es un número o una cadena numérica
 - `is_object` – Comprueba si una variable es un objeto
 - `is_real` – Alias de `is_float`
 - `is_resource` – Comprueba si una variable es un recurso
 - `is_scalar` – Comprueba si una variable es escalar
 - `is_string` – Comprueba si una variable es de tipo string

- Existencia y eliminación de variables
 - `isset` – Determina si una variable está definida y no es NULL
 - `unset` – Destruye una variable especificada
- Mostrar tipo y contenidos de una variable
 - `print_r` – Imprime información legible para humanos sobre una variable
 - `var_dump` – Vuelca información sobre una variable
 - `var_export` – Imprime o devuelve una representación string de una variable analizable
- Variables definidas globales
 - `get_defined_vars` – Devuelve una matriz con todas las variables definidas
 - `get_resource_type` – Devuelve el tipo de recurso
 - `import_request_variables` – Importar variables GET/POST/Cookie en el ámbito global

Funciones numéricas

- Absoluto, potencia y raíz cuadrada
 - `abs` – Valor absoluto
 - `pow` – Expresión exponencial
 - `sqrt` – Raíz cuadrada
- Redondear
 - `ceil` – Redondear fracciones hacia arriba
 - `floor` – Redondear fracciones hacia abajo
 - `round` – Redondea un float
- Cambio de base
 - `base_convert` – Convertir un número entre bases arbitrarias
- Máximo y mínimo
 - `max` – Encontrar el valor más alto
 - `min` – Encontrar el valor más bajo
- Números aleatorios
 - `mt_getrandmax` – Mostrar el mayor valor aleatorio posible
 - `mt_rand` – Genera un mejor número entero aleatorio

- Comprobación de infinito y número no definido
 - `is_finite` – Encuentra si un valor es un número finito legal
 - `is_infinite` – Encuentra si un valor es infinito
 - `is_nan` – Encuentra si un valor no es un número
- Residuo y número pi
 - `fmod` – Devuelve el residuo de punto flotante (módulo) de la división de los argumentos
 - `pi` – Obtener valor de pi
- Formato numérico
 - `number_format` – Formatear un número con los miles agrupados

Funciones con caracteres

- Codificación de caracteres latin1 y unicode
 - `utf8_decode` – Convierte una cadena con los caracteres codificados ISO-8859-1 con UTF-8 a un sencillo byte ISO-8859-1
 - `utf8_encode` – Codifica una cadena ISO-8859-1 a UTF-8
- Unir y dividir
 - `str_split` – Convierte un string en un array
 - `join` – Alias de implode
 - `explode` – Divide una cadena en varias cadenas
 - `implode` – Une elementos de un array en una cadena
- Contar caracteres y palabras
 - `count_chars` – Devuelve información sobre los caracteres usados en una cadena
 - `str_word_count` – Devuelve información sobre las palabras utilizadas en un string

Funciones con cadenas

- Contar subcadenas y caracteres
 - `strlen` – Obtiene la longitud de una cadena
 - `substr_count` – Cuenta el número de apariciones del

substring

- Repetir
 - str_repeat – Repite un string
- Conversión ASCII-carácter
 - chr – Devuelve un caracter específico
 - ord – devuelve el valor ASCII de una caracter
- Comparaciones
 - strcasecmp – Comparación de string segura a nivel binario e insensible a mayúsculas y minúsculas
 - strcmp – Comparación de string segura a nivel binario
 - strnatcasecmp – Comparación de strings, insensible a mayúsculas y minúsculas, utilizando un algoritmo de «orden natural»
 - strnatcmp – Comparación de strings utilizando un algoritmo de «orden natural»
 - strncasecmp – Comparación de los primeros n caracteres de cadenas, segura con material binario e insensible a mayúsculas y minúsculas
 - strncmp – Comparación segura a nivel binario de los primeros n caracteres entre strings
 - substr_compare – Comparación segura a nivel binario de dos o más strings desde un desplazamiento, hasta una longitud de caractere
- Eliminación de caracteres iniciales-finales
 - trim – Elimina espacio en blanco (u otro tipo de caracteres) del inicio y el final de la cadena
 - ltrim – Retira espacios en blanco (u otros caracteres) del inicio de un string
 - rtrim – Retira los espacios en blanco (u otros caracteres) del final de un string
- Rellenar derecha-izquierda
 - str_pad – Rellena un string hasta una longitud determinada con otro string
- Invertir
 - strrev – Invierte una string
- Reemplazamiento

- `str_ireplace` – Versión insensible a mayúsculas y minúsculas de `str_replace`
- `str_replace` – Reemplaza todas las apariciones del string buscado con el string de reemplazo
- `substr_replace` – Reemplaza el texto dentro de una porción de un string
- Subcadenas
 - `stristr` – `strstr` insensible a mayúsculas y minúsculas
 - `strstr` – Encuentra la primera aparición de un string
 - `substr` – Devuelve parte de una cadena
 - `strpbrk` – Buscar una cadena por cualquiera de los elementos de un conjunto de caracteres
 - `strrchr` – Encuentra la última aparición de un caracter en un string
 - `strchr` – Alias de `strstr`
- Buscar
 - `strpos` – Encuentra la posición de la primera ocurrencia de un substring en un string
 - `strcspn` – Encuentra la longitud del segmento inicial que no coincida con la máscara
 - `stripos` – Encuentra la posición de la última aparición de un substring insensible a mayúsculas y minúsculas en un string
 - `strrpos` – Encuentra la posición de la última aparición de un substring en un string
 - `strspn` – Encuentra la longitud del segmento inicial de un string que consista únicamente en caracteres contenidos dentro de una máscara dada.
- Aleatorio
 - `str_shuffle` – Reordena aleatoriamente una cadena
- Ajustar
 - `wordwrap` – Ajusta un string hasta un número dado de caracteres
- Conversión binaria-hexadecimal de caracteres
 - `bin2hex` – Convierte datos binarios en su

- representación hexadecimal
- hex2bin – Convierte hexadecimal a binario
- Comparar
 - strcoll – Comparación de cadenas basada en la localidad
- Mayúsculas y minúsculas
 - lcfirst – Pasa a minúscula el primer carácter de un string
 - strtolower – Convierte una cadena a minúsculas
 - strtoupper – Convierte un string a mayúsculas
 - ucfirst – Convierte el primer carácter de una cadena a mayúsculas
 - ucwords – Convierte a mayúsculas el primer carácter de cada palabra en una cadena

Funciones sobre fechas

- Intervalos de tiempo
 - date_interval_create_from_date_string – Establece un objeto DateInterval desde las partes relativas de una cadena
 - date_interval_format – Formatea el intervalo
- Crear una fecha y hora
 - date_create – Devuelve un nuevo objeto DateTime
 - date_date_set – Establece la fecha
 - date_time_set – Establece la hora
 - date_create_from_format – Devuelve un nuevo objeto DateTime formateado según el formato especificado
- Dar formato a una fecha y hora
 - date_format – Devuelve la fecha formateada según el formato dado
- Modificar una fecha
 - date_add – Añade una cantidad de días, meses, años, horas, minutos y segundos al objeto DateTime
 - date_sub – Sustraer una cantidad de días, meses, años, horas, minutos y segundos de un objeto DateTime

- Restar dos fechas
 - `date_diff` – Devuelve la diferencia entre objetos `DateTime`.
- Operadores de fecha
 - `date1 == date2` Devuelve `TRUE` si ambas fechas son la misma, y `FALSE` en otro caso
 - `date1 < date2` Devuelve `TRUE` si `date1` es una fecha anterior a `date2`, y `FALSE` en otro caso
 - `date1 > date2` Devuelve `TRUE` si `date1` es una fecha posterior a `date2`, y `FALSE` en otro caso
 - `date1 != date2` Devuelve `TRUE` si `date1` es una fecha distinta a `date2`, y `FALSE` en otro caso
 - `date1 <= date2` Devuelve `TRUE` si `date1` es una fecha anterior o igual a `date2`, y `FALSE` en otro caso
 - `date1 >= date2` Devuelve `TRUE` si `date1` es una fecha posterior o igual a `date2`, y `FALSE` en otro caso

Funciones HTML

- Codificar caracteres en html
 - `htmlspecialchars` – Convierte caracteres especiales en entidades HTML
 - `htmlentities` – Convierte todos los caracteres aplicables a entidades HTML
- Decodificar caracteres en html
 - `html_entity_decode` – Convierte todas las entidades HTML a sus caracteres correspondientes
 - `htmlspecialchars_decode` – Convierte entidades HTML especiales de nuevo en caracteres

Gestión de errores

Estas funciones se ocupan del manejo y registro de errores. Le permiten definir sus propias reglas de manejo de errores, así como modificar la manera en que los errores pueden ser registrados. Esto le permite cambiar y mejorar la notificación de errores para acomodarla a sus necesidades.

Con las funciones de registro se pueden enviar mensajes directamente a otras máquinas, a un e-mail (¡o un e-mail a un busca!), al registro del sistema, etc., por lo que puede registrar y monitorizar selectivamente las partes más importantes de sus aplicaciones y sitios web.

Las funciones de notificación de errores le permiten personalizar qué niveles y tipos de errores se dan, abarcando desde simples avisos hasta funciones personalizadas devueltas al originarse un error.

- `debug_backtrace` – Genera un rastreo
- `debug_print_backtrace` – Muestra un rastreo
- `error_clear_last` – Limpiar el error más reciente
- `error_get_last` – Obtener el último error que ocurrió
- `error_log` – Enviar un mensaje de error a las rutinas de manejo de errores definidas
- `error_reporting` – Establece cuáles errores de PHP son notificados
- `restore_error_handler` – Recupera la función de gestión de errores previa
- `restore_exception_handler` – Restaura la función de gestión de excepciones previamente definida
- `set_error_handler` – Establecer una función de gestión de errores definida por el usuario
- `set_exception_handler` – Establece una función de gestión de excepciones definida por el usuario
- `trigger_error` – Generar un mensaje de error/advertencia/aviso de nivel de usuario
- `user_error` – Alias de `trigger_error`

Mecanismos de introducción de información: formularios

Todo formulario en XHTML está compuesto por una serie de controles definidos dentro de una etiqueta

.

En la etiqueta

se deben codificar los siguientes atributos:

- action
- method
- enctype

En el atributo action se indica cuál será la página que procesará los datos.

En el atributo method se indica cuál será el método que se utilizará para enviar los datos al servidor a la página definida en action. Hay dos posibles métodos: get y post.

El método get se pasa los datos añadiéndolos en la barra de direcciones del navegador. Es un método inseguro ya que se ven los datos que se pasan, y no permite pasar un número excesivo de datos ni subir archivos. La forma de recuperar los datos en la página definida en action es mediante la variable global \$_GET que ya veremos como se utiliza.

El método post se utiliza cuando se quiere ocultar la información y/o se suben archivos. Además, mediante dicho método se puede pasar mayor cantidad de datos. En la página que procesa los datos se accede a los mismos mediante la variable \$_POST, que se utiliza del mismo modo que la de \$_GET.

Autenticación de usuarios

Autenticación HTTP con PHP

Es posible usar la función header() para enviar un mensaje «Authentication Required» al navegador del cliente causando que se abra una ventana para ingresar usuario y password. Una vez se ha llenado el usuario y password, la URL contenida dentro del script PHP será llamada nuevamente con las

variables predefinidas `PHP_AUTH_USER`, `PHP_AUTH_PW`, y `AUTH_TYPE` puestas por el nombre del usuario, password y el tipo de autenticación respectivamente. Esas variables predefinidas son encontradas en los arrays `$_SERVER` y `$HTTP_SERVER_VARS`. Ambos métodos de autenticación «Basic» y «Digest» (desde PHP 5.1.0) son soportados.

Autenticación HTTP Basic

En el contexto de una transacción HTTP, la autenticación de acceso básica es un método diseñado para permitir a un navegador web, u otro programa cliente, proveer credenciales en la forma de usuario y contraseña cuando se le solicita una página al servidor.

Autenticación HTTP Digest

Digest access authentication o Autenticación de acceso con resumen es uno de los métodos usados en servidores web para negociar credenciales, tales como nombre de usuario y contraseña, desde el navegador web. El segundo mecanismo es Autenticación de acceso básica. El método «Digest access authentication» es usado para confirmar la identidad de un usuario antes de servir información sensible, como el historial de transacciones de un banco. Se aplica una función hash a la contraseña antes de ser enviada sobre la red, lo que resulta más seguro que enviarla en texto plano como en la autenticación básica.

Autenticación de usuario

Se puede realizar una autenticación de usuario para entrar en una determinada página del siguiente modo utilizando sesiones.

Control de accesos

El login con credenciales es el elemento para realizar un acceso.

Sesiones

Una sesión es un periodo de tiempo durante el cual una persona determinada ve un número de diferentes páginas Web de un determinado dominio.

Las sesiones y las variables de sesión permiten pasar datos entre páginas sin necesidad de utilizar formularios y es la mejor forma para ocultar datos comprometidos.

Ejercicios

- **Almacenar la posición GPS en una variable de sesión**
 - <https://www.jesusninoc.com/06/16/crear-un-sistema-con-jquery-que-permita-almacenar-en-un-fichero-de-un-servidor-la-hora-longitud-latitud-de-cualquier-dispositivo-utilizando-el-gps-del-navegador-mediante-una-peticion-a-un-fichero-ph/>
 - <https://www.jesusninoc.com/02/14/almacenar-la-posicion-gps-en-una-variable-de-sesion-en-php/>
-

Configuración del intérprete

Cada intérprete tiene sus propias configuraciones.