Que es el archivo php.ini

php.ini

El archivo php.ini es el que contiene nuestra configuración de PHP, con el que podemos controlar muchos aspectos de su funcionamiento. En esta página intentaremos explicar para que sirve cada una de sus instrucciones y cual es la mejor forma de configurarlo. La sistematica de la página sigue el mismo orden interior de php.ini, aunque puede que haya ligeras diferencias con tu copia, debidas a pequeños cambios entre versiones. La configuración aqui contemplada es la que corresponde a las versiones php 4.3.x.¿que es el archivo php.ini?

Este archivo sirve para indicar una serie de valores que determinan el comportamiento del intérprete PHP. Lo encontramos dentro de la distribución php en el directorio raiz bajo el nombre php.ini-recommended o php.ini-dist. Se trata de un archivo de texto, que podemos abrir con cualquier editor que trabaje con texto simple (*.txt). Como siempre, nos será mas cómodo trabajar con un editor como html-kit que coloree sintácticamente el archivo.

Lo primero que debemos hacer es en efecto editar una de las dos versiones disponibles, configurarla de acuerdo a nuestras necesidades, y guardarla con el nombre php.ini. ¿Cual escoger? las dos son el mismo archivo, con distintos valores por defecto.

Dentro de este archivo, todo lo que comienza con un punto y coma es un comentario, y es ignorado. El texto marcado con corchetes, como [PHP] indica una cabecera de sección.

Las instrucciones se llaman directivas, y estan formadas por una pareja compuesta por la clave y su valor, por ejemplo: asp_tags = Off. Y ten cuidado, porque diferencia entre mayusculas y minusculas. No es lo mismo asp_tags que Asp_tags. También verás que algunas directivas comienzan con ; lo que quiere decir que estan comentadas ya que no son necesarias por defecto. Debes desactivarlas sin necesitas esa funcionalidad.

Otro dato mas a tener en cuenta. En windows las rutas o paths se escriben con la barra invertida (c:\windows) mientras que unix utiliza la barra (/usr/local/bin/...). En php.ini deberas indicar algunas rutas. Los formatos admisibles son:C:\directorio\directorio
\directorio\directorio
/directorio/directorio/

Si no especificas la letra de la unidad, se presupone que es la unidad actual (esto es, donde tengas php.exe). ¿como trabaja el archivo php.ini?

Antes que nada aclarar que el proceso de instalación de PHP en tu ordenador NO crea el archivo php.ini. Una vez instalado PHP debes escoger uno de los archivos proporcionados como ejemplo y renombrarlos a php.ini

Si tenemos PHP como módulo del servidor, el archivo php.ini se lee cada vez que se reinicia. Por lo tanto tienes que reiniciar para que actualice los cambios. Si PHP está instalado como cgi (no recomendado) se leerá el php.ini en cada llamada a PHP. En ambos casos, lo primero a tener en cuenta será, pues, donde archivar php.ini. El servidor lo buscará sucesivamente -y por este orden- en el propio directorio php (c:/php si usas la instalacion por defecto). Si no lo encuentra alli lo buscará en la ruta definida como variable de entorno y finalmente en el directorio de sistema (c:/windows)

Lo aconsejado es mover php.ini a tu directorio de sistema (c:\windows si tienes W98). Cuida no dejar ninguna version antigua de php.ini en el directorio php, porque podría ser leida con preferencia a la que hayas movido a /windows/. Y *recuerda* que para que cualquier cambio que realices en el php.ini surta efecto, debes reiniciar tu servidor.

Los problemas mas comunes que encontrarás con PHP pasan casi siempre por una incorrecta configuración de php.ini, y en muchos casos, por tener el archivo mal ubicado o duplicado, leyendose un archivo distinto del que tu estas configurando. Si haces un cambio en php.ini y este no se refleja en el funcionamiento de PHP, comprueba la sintaxis que has usado; que has reiniciado el servidor correctamente y que este lee el php.ini deseado. Controla siempre tus copias de php.ini !!

Es altamente recomendable que tengas preparada una pagina con la función phpinfo() para ver como queda la configuración de tu php:

```
<?php
phpinfo();
?>
```

Guarda esta página como info.php o como se te ocurra, y tenla a mano para comprobar la configuración en cuanto

tengas tu php listo. Los dos php.ini

En la carpeta PHP verás que hay dos archivos php.ini: uno php.ini-recommended y otro php.ini-dist. Los dos tienen las mismas directivas, pero configuradas de distinta forma. La version recomendada es mas exigente en cuestiones de seguridad (esencialmente la directiva registrar globales esta off y mostrar errores tambien off) mientras que dist, aunque menos segura, posiblemente permitirá funcionar a la mayoria de los scripts que puedas bajarte de internet, sin necesidad de adaptaciones.Las directivas

Veremos a continuación cada una de las directivas y su significado, siguiendo el orden que podriamos ver en nuestro php.ini. Muchas directivas vienen con valores por defecto, o sin valor determinado, o comentadas (inactivas). Una buena política es dejarlas como estan, salvo que sepas exactamente que estas haciendo.

Los valores que indicamos en esta página son indicativos. Lo que pretendemos es explicar el valor de cada directiva (al menos las que conocemos), no proponer un php.ini modélico. Opciones de lenguaje

En esta primera sección encontramos algunas instrucciones generales sobre el funcionamiento de PHP:

engine = On activa la interpretación de scripts php (si php está cargado como módulo de apache). Esta directiva, en unión de httpd.conf, permite habilitar o deshabilitar php en directorios determinados.

short_open_tag = On Permite usar en tus scripts etiquetas php abreviadas <? ... ?>, y el atajo para imprimir variables <%= $valor %>. Si el valor es off, deberas usar la forma <?php ... ?> o <script>.
Se recomienda ponerlo a off para mayor portabilidad del código

asp_tags = Off Permite usar etiquetas estilo asp <% ... %>. Deshabilitado por defecto

precision = 14 número máximo de decimales visualizados

y2k_compliance = On Forzar compatibilidad con el año 2000.

output_buffering = Off permite enviar cabeceras http (cookies por ejemplo) desde puntos distintos al inicio del script. Además de valores on | off puedes fijar aqui el tamaño maximo (en bytes) de las lineas http permitidas, por ejemplo: output_buffering = 4096

Puedes deshabilitar esta función con carácter general aqui, y habilitarla en partes concretas de tus scripts utilizando las funciones de buffer correspondientes (por ejemplo ob_start()).

Cuando output buffering esta activado, PHP no lanza las cabeceras HTTP al inicio de la ejecución del script, sino que las almacena temporalmente en un buffer de memoria, lo que te permitirá modificar o añadir instrucciones HTTP durante la ejecución del script, que se enviarán solo cuando este finalice.

Esta posibilidad esta penalizada por una disminución del rendimiento.

output_handler = Con esta directiva puedes redirigir toda la salida de tus scripts a una función PHP. Es preferible no habilitar esta opción y establecerla si es preciso en cada uno de tus scripts.

zlib.output_compression = Off habilita la libreria zlib de forma que los datos de salida del script se envian comprimidos. Puedes indicar valores off|on o precisar el tamaño del buffer (por defecto es de 4 KB).

;zlib.output_handler = Si tienes activada la opción anterior, no puedes usar la directiva output_handler; con similar funcionalidad tienes zlib.output_handler.

implicit_flush = Off Intenta enviar al cliente el contenido de la memoria intermedia de salida. O dicho coloquialmente, "envia lo que tengas hasta ahora, en lugar de esperar a completarlo". Equivale a llamar la función flush() despues de cada llamada echo o print y cada segmento html. Es desaconsejable su activación, siendo preferido usar la función flush() cuando sea necesario.

unserialize_callback_func= relacionado con las funciones serialize(). Francamente no se mas sobre el tema.

allow_call_time_pass_reference = Off Uno mas de los cambios en PHP ... tradicionalmente podias construir una función y al usarla, decidir si pasabas o no el valor de una variable por referencia (&$var). Ahora esto es desaconsejado y se recomienda especificar que los valores serán pasados por referencia en la propia declaración de la funcion (function blah (&$var))

safe_mode = Off Para activar el modo seguro de PHP.

Si usas PHP como CGI, "debes" activar safe_mode y especificar el valor de safe_mode_exec_dir, con lo cual aseguras que el usuario solo pueda acceder a la información existente en las carpetas especificadas.

safe_mode_gid = Off Por defecto, con safe_mode On PHP hace un chequeo UID del fichero al abrirlo. Con esta directiva puedes especificar en su lugar un chequeo GID

safe_mode_include_dir = Los archivos que esten en este directorio podrán ser utilizados con include/require en safe_mode On sin necesidad de chequeos UID/GID

safe_mode_exec_dir = Si el PHP se utiliza en modo seguro, la función system() y el resto de funciones que ejecutan programas del sistema solo actuaran sobre archivos ejecutables que esten en el directorio indicado.

safe_mode_allowed_env_vars = PHP_ Puedes proporcionar aqui una serie de prefijos (separados por ;). Si indicas estos prefijos, en safe_mode los usuarios solo podrán alterar variables de entorno cuyo nombre comience con ese prefijo. Si esta directiva esta vacia, en safe_mode podrán modificarse todas las variables de entorno.

safe_mode_protected_env_vars = LD_LIBRARY_PATH una lista de variables de entorno (separadas por ;) que no pueden variarse via putenv() incluso aunque safe_mode_allowed_env_vars lo permita

open_basedir = Limita los archivos que se pueden abrir por PHP al árbol de directorios especificado.

Cuando un script intenta abrir un archivo con, por ejemplo, fopen, se comprueba su localización. Si el fichero está fuera del árbol de directorios especificado, PHP se negará a abrirlo. Todos los enlaces simbólicos son resueltos, de modo que no es posible evitar esta limitación usando uno de ellos.

El valor especial . indica que el directorio base será aquel en el que reside el script.

Bajo Windows, los directorios se separan mediante punto y coma. En el resto de sistemas, con dos puntos ":". Como módulo de Apache, los senderos para open_basedir de los directorios padre se heredan ahora automáticamente.

El valor por defecto es permitir abrir todos los archivos.

Esta directiva es independiente de Safe Mode.

disable_functions = Con esta directiva puedes inhabilitar con carácter general determinadas funciones PHP. Basta con incluirlas separadas por punto y coma (";"). Al igual que la anterior, es independiente de Safe Mode.

highlight... permite especificar los colores a utilizar por el coloreador de sintaxis interno de PHP

expose_php = On Permite controlar si PHP debe o no revelar su presencia en el servidor, por ejemplo incluyendose en las cabeceras http del servidor.Limites al empleo de recursos

max_execution_time = 30 Fija el tiempo máximo en segundos que se le permite usar a un script antes de ser finalizado por el intérprete. Así se evita que scripts mal escritos puedan bloquear el servidor.

max_input_time = 60 Tiempo máximo en segundos que el script puede invertir en analizar datos recibidos

memory_limit = 8M Fija el tamaño máximo de memoria en bytes que se permite reclamar a un script. Así se evita que script mal escritos se coman toda la memoria disponible de un servidor. Gestion y archivo de errores

error_reporting = E_ALL Fija el nivel (detalle) con el que PHP te informa de errores. Esta directiva vuelca el informe de errores en la pantalla, y su uso está desaconsejado en páginas en producción, ya que el error puede revelar información sensible. Lo recomendado es permitir mostrar errores, con el máximo detalle posible, mientras desarrollas el script PHP; y cuando está terminado y en producción, deshabilitar el mostrado de errores en pantalla y activar en su lugar el archivo de errores.

Como cada nivel de informe de error está representado por un número, puedes designar el nivel deseado sumando valores:
1 errores normales
2 avisos normales
4 errores del parser (error de sintaxis)
8 avisos de estilo no críticos
El valor por defecto para esta directiva es 7 (se muestran los errores normales, avisos normales y errores de parser).

Tambien puedes designar el nivel de error nominativamente:

Algunas combinaciones son:
error_reporting = E_ALL & ~E_NOTICE muestra todos los errores criticos, excluyendo advertencias que pueden indicar mal funcionamiento del código pero no impiden la ejecución del intérprete.

error_reporting = E_COMPILE_ERROR|E_ERROR|E_CORE_ERROR muestra solo errores.

error_reporting = E_ALL muestra todos los errores y advertencias.

display_errors = Off determina si los errores se visualizan en pantalla como parte de la salida en HTML o no.

Como queda dicho, es desaconsejado mostrar errores en pantalla en páginas visibles al público.

display_startup_errors = Off Incluso con display_errors on, por defecto PHP no muestra los errores que detecta en la secuencia de encendido. Con esta directiva puedes mostrar estos errores. Desaconsejado activarla.

log_errors = On Guarda los mensajes de error en un archivo. Normalmente el registro del servidor. Esta opción, por tanto, es específica del mismo.

log_errors_max_len = 1024 Especifica el tamaño del archivo error_log. Si tiene un valor 0 significa que no hay restricción de tamaño

ignore_repeated_errors = Off Si está activado, no archiva mensajes repetidos. No se consideran mensajes repetidos aquellos que no provienen de la misma linea.

ignore_repeated_source = Off Si está activado, considera repetidos los mensajes de error iguales, aunque provengan de distinta linea / script

report_memleaks = On Mostrar o no. memory leak se refiere a cuando (por error) el script no libera la memoria usada cuando ya no la necesita, y en consecuencia usa cada vez mas hasta llegar a agotarla.

track_errors = Off Si lo activamos, tendremos el último mensaje de error/advertencia almacenado en la variable $php_errormsg

html_errors = Off Si activo, no incluye etiquetas HTML en los mensajes de error.

docref_root = /phpmanual/ y docref_ext = .html Si tienes html_errors activado, PHP automaticamente incluye enlaces en el mensaje de error que te dirigen a la página del manual que explica la función implicada. Puedes bajarte una copia del manual y indicar su ubicación (y extensión del archivo) usando estas directivas.

error_prepend_string = "" Cadena a añadir antes de cada mensaje de error.

error_append_string = "" cadena a añadir despues del mensaje de error.

;error_log = filename Nombre del fichero para registrar los errores de un script. Si se utiliza el valor especial syslog, los errores se envían al registro de errores del sistema. Como verás, esta comentado (inhabilitado) por defecto. Gestion de datos

track_vars Esta directiva crea arrays $HTTP_GET_VARS, $HTTP_POST_VARS y $HTTP_COOKIE_VARS con los datos introducidos con los métodos GET, POST y con cookies. Desde PHP 4.0.3 está siempre activada.

;arg_separator.output = "&" El carácter que se empleará en las urls generadas por PHP para separar argumentos (valores pasados via url). & es el separador por defecto.

;arg_separator.input = ";&" separadores que usará PHP cuando analice una url suministrada para almacenarla en variables

variables_order = "GPCS" Esta directiva fija el orden (precedencia) en que PHP registrará y interpretará las variables de entorno (de izquierda a derecha en el orden indicado). Los valores posibles se toman con las iniciales del método usado para asignar el valor a la variable: Get, Post, Cookie, Enviroment y Server. Fijando por ejempo el valor a "GP", hará que el PHP ignore por completo las cookies y que sobreescriba las variables recibidas por GET con las que tengan el mismo nombre y vengan por POST.

En php.ini encontrarás una directiva semejante en desuso (no recomendada) que es gpc_order

register_globals = Off Permite registrar autómaticamente (o no) las variables EGPCS como globales. Por razones de seguridad se recomienda desactivar el registro.

register_argc_argv = Off Esta directiva instruye a PHP si debe declarar las variables argv&argc (arrays predefinidos que almacenan los parámetros pasados (argv) y su numero (argc).

post_max_size = 8M Tamaño máximo de los datos que PHP aceptará por el método POSTMagic quotes

magic_quotes_gpc = Off Fija el estado magic_quotes para operaciones GPC (Get/Post/Cookie). Si magic_quotes vale on, todas las ' (comilla sencilla), " (comilla doble), \ (barra invertida) y los NUL son automáticamente marcados con una barra invertida. Si además magic_quotes_sybase vale on, la comilla sencilla es marcada con otra comilla sencilla en lugar de la barra invertida.

magic_quotes_runtime = Off Si se habilita magic_quotes_runtime, muchas de las funciones que devuelven datos de algún tipo de fuente externa incluyendo bases de datos y archivos de texto devolverán las comillas marcadas con una barra invertida. Si también está activo magic_quotes_sybase, la comilla simple es marcada con una comilla simple en lugar de la barra invertida.

magic_quotes_sybase = Off Si magic_quotes_sybase está a on, la comilla simple es marcada con una comilla simple en lugar de la barra invertida cuando están habilitados magic_quotes_gpc o magic_quotes_runtime. Mas directivas de Gestion de datos

auto_prepend_file = y auto_append_file = permiten indicar la ruta y nombre de un archivo que se añadirán antes o despues (respectivamente) de todos los archivos php que se ejecuten.
El valor especial none desconecta la adición automática de archivos.
Si el script es terminado con exit(), no tendrá lugar la adición automática señalada con auto_append_file.
Los archivos indicados con estas directivas se incluirán como si fuesen llamados mediante la función include(), así que se utiliza include_path.

;default_charset = "iso-8859-1" Por defecto, el código de caracteres indicado por PHP en la cabecera de salida.

default_mimetype = "text/html" Por defecto, el tipo mime de salida de datos. Cada MIMETYPE define el formato de los datos (por ejemplo, texto/hmtl, jpg, gif ....)

;always_populate_raw_post_data = On PHP crea la variable $HTTP_RAW_POST_DATA cuando recibe datos via POST cuyo tipo MIME no reconoce (almacena los datos en esta variable sin analizarlos). Con esta directiva se ordena que se cree siempre la variable $HTTP_RAW_POST_DATA, aunque el tipo MIME sea conocido.

;allow_webdav_methods = On Permite manejar las peticiones http propias de webdav.Rutas y directorios

include_path = ".;c:\php\includes" Permite especificar una lista de directorios en los que las funciones require(), include() y fopen_with_path() buscaran los archivos requeridos. El formato es similar a la variable de entorno de sistema PATH: una lista de directorios separados por dos puntos en UNIX o por punto y coma en Windows. Ejemplo unix seria include_path=.:/home/httpd/php-lib y en windows include_path=".;c:\www\phplib".

El valor por defecto para esta directiva es . (sólo el directorio actual).

doc_root = Indica el "Directorio raiz" donde estan nuestras paginas php en el servidor. Sólo se usa si no está vacío. Si PHP se configura con safe mode, no se interpretaran las páginas php situadas fuera de este directorio. Ojo con los servidores virtuales que apuntan a zonas distintas del servidor.

user_dir = El directorio raiz para los archivos PHP bajo el directorio inicial de un usuario (/~usuario). Normalmente se deja vacio

extension_dir = ./ En qué directorio debe buscar el PHP las extensiones dinámicas a cargar. Bajo Windows, por defecto si no pones ningun valor en esta directiva, se buscaran en c:\php4\extensions\.

enable_dl = On Esta directiva sólo es útil en la versión del PHP como módulo del Apache. Puede habilitar o deshabilitar para un servidor virtual o para un directorio la carga dinámica de extensiones de PHP mediante dl().

La razón principal para desactivar la carga dinámica es la seguridad. Con la carga dinámica es posible ignorar las restricciones para abrir archivos establecidas con open_basedir.

El valor por defecto es permitir la carga dinámica, excepto cuando se usa safe_mode. En modo seguro, es imposible

usar dl().

cgi.force_redirect = 1 Por defecto se activa. Es una directiva importante de seguridad que "debes" activar si ejecutas en tu apache PHP como cgi (no es necesaria si tienes PHP como modulo, o si usas como servidor el IIS de microsoft).

; cgi.redirect_status_env = ; En conjunción con cgi.force_redirect y servidores distintos de Apache o iPlanet.

; fastcgi.impersonate = 1; En conjunción con IIS y FastCGISubir ficheros

file_uploads = On Permitir o no subir (upload) ficheros via HTTP.

upload_tmp_dir = Carpeta o directorio utilizable para guardar temporalmente archivos subidos por PHP. Si no se especifica, usará el designado por defecto por el servidor. El usuario que esté ejecutando el script debe tener permiso de escritura en ese directorio.

upload_max_filesize = 2M Tamaño máximo de archivos que pueden subirse. directivas relacionadas con fopen

allow_url_fopen = On Permite pasar urls (http, ftp) a la función fopen(), en lugar de la ubicacion fisica del archivo

;from="john@doe.com" define el email a usar como contraseña para ftp anonimo

;user_agent="PHP" define la "firma" que dejará PHP en el servidor remoto de donde coge los archivos

default_socket_timeout = 60 timeout en segundos para la apertura de sockets

; auto_detect_line_endings = Off Si activo, PHP detectara automaticamente el carácter que indica fin de linea (distinto en windows, linux y windows)Extensiones dinamicas

extension= Qué extensiones dinámicas debe cargar el PHP cuando arranca. Debes elegir el archivo que corresponde a tu sistema operativo: por ejemplo extension=msql.dll para windows, extension=msql.so para linux.

Ojo, aqui solo indicamos la extension de los archivos, no su ubicación. Los archivos DEBEN estar en el directorio especificado mas arriba con extension_dir.

Las versiones mas recientes de PHP traen "de serie" los modulos MYSQL, ODBC y GD por lo que NO tienes que cargar sus extensiones.Configuracion de modulos de PHP

define_syslog_variables = Off Permite definir variables del sistema. Recomendado Off.

;browscap = extra/browscap.ini El archivo browscap.ini es un archivo de texto que contiene información sobre las cadenas de identificación que usa cada navegador. Mediante esta directiva indicas a PHP donde tienes browscap.ini; se usa conjuntamente con la función get_browser().Directivas de Configuración de Correo

Si usas PHP bajo linux, puedes enviar correo usando tu propio PC con sendmail; con windows no tienes esa posibilidad, por lo que para enviar correos desde un script PHP con la funcion mail() tienes que delegar en tu configuración de correo ordinaria, la que usas por ejemplo con outlook para enviar y recibir correo.

Este seria un ejemplo bajo windows:

SMTP = mailhost@teleline.es Este seria el caso si tu conexion a internet te la proporciona telefonica. Especificamos la direccion del servidor smtp (correo saliente).

sendmail_from= webmaster@misitio.com La dirección del remitente ("De:") para los correos enviados desde PHP bajo Windows.