

iranphp articles

عنوان مقاله : نکاتی مفید در مورد تنظیم `php.ini`
نگارنده : فرشاد پایدار, جابر صادقی
آدرس پست الکترونیک : `www.rahacomputer.g3z.com, farshadcdeb@yahoo.com`
تاریخ نگارش :

نکاتی مفید در مورد تنظیم php.ini :

فایل پیکربندی PHP

قبل از شروع مطالب نگاهی اجمالی به ویژگیهای فایل پیکربندی PHP منطقی به نظر می رسد. نام این فایل `php.ini` می باشد و به دلایلی از ساختار کلی فایل های `ini` که بیشتر در برنامه های کاربردی ویندوز استفاده می شود، پیروی می کند. این فایل متنی به صورت `ASCII` می باشد که به بخشهایی (`session`) تقسیم می شود. هر بخش متغیرهایی را که مربوط به آن بخش می شوند، در بر می گیرد. هر بخش چیزی شبیه زیر می باشد :

```
[MySection]
variable="value"
anothervariable="anothervalue"
```

نام بخش در یک جفت براکت "[]" در بالا قرار می گیرد. و زیر آن در هر سطر یک متغیر با مقدارش قرار می گیرد. متغیرها به حروف حساس (`case sensitive`) هستند و نمی توانند شامل فاصله (`space`) باشند، در حالی که مقادیر متناظر آنها می توانند اعداد، رشته کاراکتری (`string`) و یا `Boolean` باشند.

سمی کالن ";" در ابتدای هر سطر به عنوان نشانگر درج توضیحات (`comment`) می باشد. این راهی ساده برای فعال یا غیرفعال کردن ویژگیهای PHP می باشد، به جای حذف کردن یک سطر می توانید به این روش آن را به توضیح تبدیل کرده که در پردازش فایل ها شرکت نخواهد کرد. در صورتی که فکر می کنید ممکن است بعد ها بخواهید از ویژگی که اکنون به کارتان نمی آید، استفاده کنید بهتر است آن را پاک نکنید .

برای اینکه PHP بتواند فایل `php.ini` را تشخیص بدهد باید آن را در دایرکتوری جاری، دایرکتوری که در متغیر `$PHPRC` تعریف شده، یا دایرکتوری که هنگام کامپایل مشخص شده است (برای ویندوز این همان دایرکتوری اصلی PHP است) نگهدارید .

بعد از اینکه شما تغییراتی در فایل `php.ini` دادید برای آنکه تغییرات اعمال شوند باید سرور خود را مجدداً راه اندازی نمایید (سرور را `restart` کنید). البته فرض بر این است که شما قبلاً در حال استفاده PHP بر سرور خود بوده اید. برای استفاده های `command-line` از PHP فایل پیکربندی `php.ini` در هر باری که برنامه PHP احضار می شود خوانده و چک می شود .

تنظیم ویژگیهای مفسر

اولین مرحله در این مقاله از مهمترین گامهای آن می باشد، ویژگیهایی که به مفسر زبان PHP مربوط می شود. اولین مورد در اینجا `engine` `variable` می باشد که چک می کند که `PHP engine` باید فعال `"on"` یا غیر فعال `"off"` باشد. `Off` کردن این متغیر به معنی این است که کدهای PHP توسط سرور تفسیر نشوند. معمولاً نیازی به این تغییر حس نمی شود، پس این متغیر را به صورت `on` رها کنید .

```
engine=On
```

متغیر `short_open_tag` کنترل می کند که آیا مفسر باید تگهای کوتاه را همانند تگ استاندارد تشخیص دهد یا نه. در صورتی که تشخیص می دهید این نمونه تگها باعث ناسازگاری با دیگر زبانها می شوند یا اینکه تصمیم به استفاده از قوانین `syntax` مختص PHP دارید می توانید آن را `off` کنید .

```
short_open_tag = On
```

معمولاً `session`، `cookie` و اطلاعات `HTTP header` در یک اسکریپت PHP باید قبل از تولید هرگونه خروجی توسط آن اسکریپت فرستاده شوند. در صورتی که این در برنامه شما امکان پذیر نیست می توانید هر چه را که PHP به نام `output buffering` می شناسد، توسط متغیرهای `output_buffering` فعال کنید .

با `on` بودن `output buffering`، زبان PHP خروجی اسکریپت شما را در یک بافر مخصوص نگه می دارد و هنگامی آنها را می فرستد که به طور مشخص به آن دستور داده شود. این به شما امکان می دهد که `cookie` ها یا اطلاعات `HTTP header` را از وسط یا انتهای کد اسکریپت خود بفرستید. هر چند این می تواند کارایی را در حاشیه کاهش دهد .

```
output_buffering = Off
```

شما می توانید با یک عدد به عنوان مقدار برای متغیر `output_buffering` اندازه بافر را تعیین کنید .

```
output_buffering = 2048
```

هنگامی که PHP شروع به کار می کند یک پیام حاوی شماره نسخه خود به هدر استاندارد سرور می افزاید. برای غیرفعال کردن آن می توانید متغیر `expose_php` را `off` کنید. این می تواند مفید باشد، مثلاً اگر بخواهید امکانات وب سرور خود را از دید هکرها پنهان کنید.

```
expose_php = On
```

تنظیم مسیر جستجوی

شما می توانید مسیر جستجویی را برای PHP با تنظیم متغیر `include_path` که لیستی از مسیرها را می پذیرد، مشخص کنید. هنگامی که با فایلی که مسیر آن مشخص نشده است، مواجه می شود به طور اتوماتیک این مسیرها را چک می کند. اگر شما کتابخانه ای از توابع یا مجموعه ای از کلاسها دارید که زیاد از آنها استفاده می کنید می توانید دایرکتوری موقعیت آنها را در این متغیر قرار دهید. همچنین این متغیر مکان مناسبی برای اضافه کردن دایرکتوری `PEAR` خود PHP می باشد، که مجموعه ای از کلاسهای که بسیار استفاده می شوند را شامل می شود.

```
include_path = ".:usr/local/lib/php/pear:"
```

کاربران ویندوز برای مشخص کردن چندین دایرکتوری باید از سمی کالن ":" بین آنها استفاده کنند و کاربران `unix` از کالن "." دو متغیر جالب در این زمینه `auto_prepend_file` و `auto_append_file` می باشند. این متغیرها فایلهایی را مشخص می کنند که PHP به ابتدا یا انتهای هر سند PHP اضافه خواهد کرد. این مورد بیشتر برای اضافه کردن `header` و `footer` به صفحاتی است که توسط PHP ایجاد می شوند، به طوری که باعث می شود تعدادی خط کد به هر سند PHP که شما می نویسید اضافه گردد. این فایلها هم می توانند اسکریپتهای PHP و هم `html` معمولی باشند، ولی کدهای PHP برای این امر باید در تگ استاندارد محصور شده باشند.

```
auto_prepend_file = /home/web/includes/header.php
auto_append_file = /home/web/includes/legal.php
```

چگونگی رفتار با خطاها

خطاها در PHP در چهار دسته می آیند `parsing errors`: یا خطای زمان تجزیه (کامپایل)، تذکرات درمورد خطاهای موجود در کد مثلاً مقدار دهی نکردن یک متغیر، `warning` یا هشدار و `fatal error`. معمولاً هنگامی که PHP با یکی از خطاهای `parsing error` یا `warning` یا `fatal error` مواجه می شود، خطاها را نمایش می دهد و اگر خطا از نوع `fatal error` باشد اجرای اسکریپت را در همان نقطه متوقف می کند. شما می توانید این رفتار را با متغیر `error_reporting` که چند فیلد خطا را می گیرد و تنها خطاهایی را که با آنها سازگار باشند نمایش می دهد، اصلاح کنید.

```
error_reporting = E_ALL
```

برای غیر فعال کردن نمایش خطاها مقدار متغیر `display_errors` را به نادرست تغییر دهید و به جای آن پیام خطاها را در ثبات خطاها از طریق متغیر `log_errors` بنویسید.

انجام این کار برای امنیت بیشتر مفید است، با غیرفعال کردن نمایش خطا شما اطلاعات ویژه سیستم خود را مخفی می کنید بدون توجه به اینکه آیا کاربران قصد خراب کردن سایت شما را دارند یا فقط استفاده می کنند. به جای آن شما باید خطاها را در یک فایل دلخواه یا ثبات خطای سیستم (`system logger`)، با تنظیم مقدار متغیر `error_log` به مسیر فایل انتخابی یا مقدار پیش فرض `"syslog"`، ثبت کنید. فقط به یاد داشته باشید که مرتباً به این فایل سرزنشید تا ببینید که در برنامه شما چه خطاهایی وجود دارد.

```
display_errors = Off
log_errors = On
error_log = "error.log"
```

فعال کردن توسعه ها

توسعه های متعددی برای PHP موجود است. در سیستمهای یونیکس لازم است که این توسعه ها در زمان کامپایل ساخته شوند. در ویندوز، dllهای باینری وجود دارند که باید ضمیمه شوند. متغیر `extension_dir` محتوی نام شاخه ای است که PHP در آن به دنبال این توسعه ها می گردد. مثلا:

```
extension_dir = "C:\Program Files\Internet Tools\Apache\bin\php4\extensions"
```

نسخه ویندوز PHP به همراه ۲۰ توسعه منتشر می شود که تمامی آنها در فایل `php.ini` لیست شده اند. برای فعال کردن هر توسعه، علامت سمی کالن جلوی آن را حذف کرده و سرور خود را از نو راه اندازی کنید. برای غیر فعال کردن یک توسعه نیز می توانید جلوی آن علامت سمی کالن را اضافه کنید. اگر توسعه مورد نظرتان در لیست توسعه ها در فایل `php.ini` نیست از متغیر `extension` استفاده کرده و نام `dll` مورد نظر را به آن نسبت دهید. مثلا:

```
extension=php_domxml.dll
extension=php_dbase.dll
```

تنظیم متغیرهای مربوط به توسعه ها

متغیرهای مربوط به توسعه ها، در بخش جداگانه ای از فایل پیکربندی (`php.ini`) ذخیره می شوند. برای مثال تمام متغیرهایی که به توسعه های MySQL مربوط می شوند باید در قسمت [MySQL] در فایل `php.ini`، قرار بگیرند.

اگر قصد استفاده از تابع `mail` در PHP را دارید، ۳ متغیر وجود دارد که باید آنها را تنظیم کنید. متغیرهای SMTP (و `sendmail_from` در ویندوز) یا `sendmail_path` در یونیکس) در هنگام ارسال e-mail از طریق تابع `mail` مورد استفاده قرار می گیرند. در ویندوز این متغیرها برای تنظیم SMTP Server و همچنین برای تعیین آدرس From در ساختار e-mail به کار می روند. در یونیکس متغیر `sendmail_path` مسیری MTA یا Mail Transfer Agent را برای ارسال `mail` تنظیم می کند.

```
SMTP = myserver.localnet.com
sendmail_from = me@localhost.com
sendmail_path = /usr/sbin/sendmail
```

متغیرهای `java.library.path`، `java.library`، `java.home`، `java.class.path` و همگی آدرس کلاس ها و کتابخانه های جاوا را تنظیم میکنند. این مقادیر به وسیله توسعه های جاوا مورد استفاده قرار می گیرند. بنابراین اگر می خواهید PHP بتواند ارتباط صحیحی با برنامه های جاوای شما برقرار کند از درستی و معتبر بودن مقادیر این متغیرها اطمینان حاصل کنید.

```
java.class.path = .\php_java.jar
java.home = c:\jdk
java.library = c:\jdk\jre\bin\hotspot\jvm.dll
java.library.path = .\
```

متغیر `session.save_path` مشخص کننده آدرس شاخه موقتی است که برای ذخیره اطلاعات `session` ها بکار می رود. معمولاً این متغیر به طور پیش فرض مقدار `/tmp` را دارد، اما از آنجا که این شاخه در ویندوز وجود ندارد باید آن را با آدرس درست شاخه موقت ویندوز تصحیح کنید در غیر این صورت هنگام فراخوانی دستور `session_start` با خطا مواجه می شوید. همچنین می توانید با تنظیم متغیر `session.cookie_lifetime` مدت اعتبار `cookie` های `session` را بر حسب ثانیه مشخص کنید.

```
session.save_path = c:\windows\temp
session.cookie_lifetime = 1800
```

تنظیمات مربوط به امنیت

تعدادی متغیر در `php.ini` وجود دارد که به امنیت PHP نصب شده بر روی سیستم شما مربوط می شوند. مهمترین آنها متغیر `safe_mode` می باشد که به طور مثال استفاده از آن برای محدود کردن آنچه کاربر می تواند از طریق PHP انجام دهد به ISPها توصیه میشود.

```
safe_mode = Off
```

اگر `safe mode` فعال باشد (`safe_mode = on`) می توانید با تنظیم متغیر `safe_mode_include_dir` مشخص کنید که کدام شاخه ها برای فایل ها مورد جستجو قرار بگیرند. همچنین میتوانید نوع برنامه هایی که کدهای PHP می توانند از طریق فرمان `exec` آنها را اجرا کنند، را محدود کنید. بدین منظور باید این برنامه های مجاز را در یک شاخه مخصوص قرار دهید و از طریق متغیر `safe_mode_include_dir` به PHP بگویید که تنها در آن شاخه به دنبال آنها بگردد. بدین ترتیب تنها برنامه های درون این شاخه توسط فرمان `exec` قابل دسترس خواهند بود.

```
safe_mode_include_dir = /usr/local/lib/php/safe-include
safe_mode_exec_dir = /usr/local/lib/php/safe-bin
```

شما می توانید اعمال مربوط به فایل ها را از طریق متغیر `open_basedir` محدود کنید. آدرس نسبت داده شده به این متغیر به عنوان ریشه اصلی (`root`) برای اعمال مربوط به فایل ها به کار می رود. وقتی که این متغیر مقدار می گیرد فایل هایی که در ساختار درختی این ریشه نباشند برای PHP غیر قابل دسترس خواهند بود. این روش مناسبی برای محدود کردن کاربران یک سیستم اشتراکی می باشد تا آنها تنها به آدرسهای مربوط به خودشان دسترسی داشته باشند.

```
open_basedir = /home/web/
```

متغیر `max_execution_time` تعیین میکند که PHP حداکثر چند ثانیه منتظر پایان یافتن یک کد باشد قبل از آنکه به اجبار آن را خاتمه دهد. وقتی کد شما گرفتار حلقه های بینهایت میشود این متغیر بکار می آید. اگر چه ممکن است این موضوع گاهی مزاحمت ایجاد کند مثلاً زمانی که کد نوشته شده نیاز به انجام اعمال زمانبری دارد، مثلاً `upload` کردن یک فایل حجیم. در این مواقع باید مواظب باشید که این مقدار را افزایش دهید تا مانع آن شوید که PHP اجرای کد شما را در وسط یک کار مهم خاتمه دهد.

```
max_execution_time = 90
```

تنظیمات مربوط به `upload` فایل ها و متغیر های فرمها

اگر پیکربندی های امنیتی که تاکنون ذکر شد کافی نیست باز هم می توانید با غیر فعال کردن قابلیت `upload` فایل از طریق متغیر `file_uploads`، یا با محدود کردن حداکثر حجم قابل `upload` از طریق متغیر `upload_max_filesize` سیستم خود را از این هم امن تر کنید. اغلب تمایل خواهید داشت که فضای کوچکی را برای `upload` فایلها اختصاص دهید مگر اینکه برنامه ای داشته باشید که قرار باشد فایل هایی نظیر یک گالری عکس یا یک `FTP Service` را پذیرا باشد.

```
file_uploads = On
upload_max_filesize = 2M
```

اگر تمایلی به `upload` کردن فایلها ندارید اما از تعداد زیادی فرم در صفحات PHP تان استفاده میکنید، ۲ متغیر دیگر وجود دارد که باید برایتان جالب باشد. اول متغیر `register_globals`، علت بیشترین سردردهای برنامه نویسان با سابقه PHP در `PHP 3.x` این متغیر به طور پیش فرض فعال بود (`on`) که موجب میشد وقتی یک فرم `submit` می شد متغیرهای آن بطور اتوماتیک به متغیرهای PHP تبدیل شوند. مسایل امنیتی موجب شد که در `PHP 4.x` این متغیر بطور پیش فرض غیر فعال باشد (`off`). در نتیجه متغیرهای فرم تنها به کمک آرایه های ویژه `$_GET` و `$_POST` قابل دسترس باشند. این امر خیلی از کدهای نوشته شده با `PHP 3.x` را با مشکل مواجه ساخت. مثلاً مقداری که در یک فیلد `text` وارد شده بود در `PHP 3.x` با `$email` قابل دسترس بود در حالی که در `PHP 4.x` با `$_POST[email]` یا `$_GET[email]` میتوان به آن رجوع کرد.

میتوانید این متغیر را غیر فعال کنید (`off`) تا امنیت بیشتری را در برابر حملاتی که از طریق فرمها صورت می گیرند برقرار کنید. به منظور سازگاری با کدهای نوشته شده با `PHP 3.x` آن را فعال کنید:

```
register_globals = on
```

متغیر دیگری که به کار با فرم ها مربوط میشود `post_max_size` است که کنترل می کند حداکثر چه حجم داده ای توسط متد `POST` یک فرم پذیرفته شود. به نظر نمی آید که زمانی لازم شود این مقدار را از `۸ MB` افزایش دهید. در عوض احتمال دارد بخواهید آن را به یک مقدار واقعی تر تغییر دهید. به هر حال اگر قصد دارید از امکانات `upload` فایل در PHP استفاده کنید این مقدار را بیشتر از مقدار متغیر `upload_max_size` قرار دهید.

```
post_max_size = 8M
```

یک متغیر جدید هم در PHP 5 معرفی شده است. `max_input_time`: که حداکثر زمان بر حسب ثانیه که عمل دریافت داده های ورودی از طریق POST، GET و PUT می تواند طول بکشد را مشخص می کند. اگر برنامه شما تحت یک اتصال کند اجرا می شود بهتر است این مقدار را افزایش دهید تا به برنامه اجازه دهید مدت بیشتری را به دریافت داده های ورودی اختصاص دهد.

```
max_input_time = 90
```

افزایش کارایی

هنوز مقادیر دیگری هم وجود دارد که با دستکاری آنها می توانید کارایی مفسر PHP را بالا ببرید. به منظور جلوگیری از اجرای کدهایی که ممکن است تمام حافظه موجود سیستم را اشغال کنند، PHP به شما اجازه می دهد که برای استفاده از حافظه محدودیتی معین کنید. این کار از طریق متغیر `memory_limit` قابل انجام است که حداکثر میزان حافظه قابل استفاده توسط یک برنامه واحد را مشخص می کند.

```
memory_limit = 8M
```

مقدار متغیر `memory_limit` اغلب باید از مقدار متغیر `post_max_size` بیشتر باشد. مساله دیگری که با دانستن آن می توانید کارایی را افزایش دهید غیر فعال کردن متغیرهای `$argc` و `$argv` است که تعداد و محتوای آرگومان های خط فرمان که به یک برنامه ارسال شده اند، را مشخص می کنند.

```
register_argc_argv = false
```

همینطور غیر فعال کردن آرایه های `$HTTP_GET_VARS`، `$HTTP_POST_VARS` چرا که به احتمال زیاد در دنیای جدید `$_GET`، `$_POST` دیگر نیازی به آنها نیست. غیر فعال کردن این اجزا باعث بهبود کارایی می شود البته فقط در PHP 5 و از طریق متغیر `register_long_arrays` قابل انجام است.

```
register_long_arrays = false
```

تابع ini_set

و در آخر نکاتی در مورد تابع `ini_set`. در حالی که PHP تمام تنظیماتش را در هنگام بالا آمدن، از فایل `php.ini` می خواند، این اجازه را به شما می دهد که این تنظیمات را به کمک تابع `ini_set` و در برنامه خود انجام دهید. البته این تغییرات فقط برای همان برنامه ای که در آن از تابع `ini_set` استفاده کرده اید اعمال می شوند. این تابع ۲ آرگومان می گیرد: نام متغیر پیکربندی ای که می خواهید مقدارش را تغییر دهید و مقدار جدید آن. این هم یک مثال که حداکثر زمان اجرای مجاز برای برنامه ای که این کد در داخل آن نوشته شده است را افزایش می دهد.

```
<?php
ini_set('max_execution_time', 600);
// more code
?>
```

باز هم یادآوری می شود که این تنظیمات در داخل هر برنامه ای که نوشته شود فقط همان برنامه را تحت الشعاع قرار می دهد. وقتی که برنامه خاتمه پیدا کند مقدار متغیر به طور اتوماتیک به مقدار اولیه تغییر خواهد کرد.

اگر برنامه های PHP شما بر روی یک سرور مشترک (Shared Server) اجرا می شوند احتمالا شما دسترسی به فایل پیکربندی `php.ini` نخواهید داشت. در این موارد تابع `ini_set` کمک شایانی می کند. چرا که اجازه می دهد تنظیمات مورد نظرتان را برای برنامه های خودتان اعمال کنید.