

WordPress ist eine beliebte Software für Blogs, die viele Plugins, Themes und Einstellmöglichkeiten bietet.

Sobald mehrere Plugins installiert und Artikel mit Bildern geschrieben worden sind, lässt die Performance von WordPress zu wünschen übrig. Die nachfolgende Anleitung soll Benutzern helfen, WordPress schneller zu machen und mit einfachen Schritten die Ladegeschwindigkeit und Performance zu steigern.

WordPress schneller machen - So gehts

Die Ladezeit einer Internetseite ist heutzutage ein nicht zu verachtender Faktor, wenn es darum geht, gute Rankings bei Google zu erzielen und Besucher auf die eigene Internetseite zu bringen.

Google selbst hat eine tolle Software veröffentlicht, die Betreibern von Internetseiten aufzeigen soll, wo die Schwächen liegen und wie man diese beheben kann.

Installation von Google Page Speed

Google Page Speed können Sie hier (https://developers.google.com/speed/pagespeed/insights_extensions) kostenlos herunterladen und installieren. Sobald die Installation abgeschlossen ist, rufen Sie Ihre Internetseite auf, wählen mit einem Rechtsklick „Element untersuchen“ und wählen „Page Speed“ aus.

Lassen Sie nun Ihre Seite analysieren. Sie werden eine Punktzahl angezeigt bekommen und eine Liste mit Dingen, die Optimierungsbedarf haben.

Übersicht

Die Seite Tutova.de - Tutorials, Anleitungen un... hat einen Gesamt-PageSpeed Score von 93 (von 100 Punkten) erhalten. [Weitere Informationen](#)

Zusammenfassung

Klicken Sie auf die Namen der Regel, um Verbesserungsvorschläge anzuzeigen.

- Hohe Priorität** Diese Vorschläge stellen die größten potenziellen Leistungsgewinne für den geringsten Entwicklungsaufwand dar. Für diese Website gibt es jedoch keine Vorschläge hoher Priorität. Sehr gut!
- Mittlere Priorität** Diese Vorschläge können kleinere Verbesserungen oder viel mehr Implementierungsarbeit beinhalten. Sie sollten sich als Nächstes um diese Elemente kümmern:
CSS (klein) inline einfügen, JavaScript später parsen, Skalierte Bilder bereitstellen
- Niedrige Priorität** Diese Vorschläge stellen die kleinsten Verbesserungen dar. Sie sollten sich nur um diese Elemente kümmern, nachdem Sie die Elemente mit einer höheren Priorität bearbeitet haben:
Bilder optimieren, Reihenfolge der Formate und Skripts optimieren, Browser-Caching nutzen, Umleitungen minimieren, JavaScript reduzieren, Cache-Validierer angeben, HTML reduzieren, Bildabmessungen festlegen, Abfragemarkierungen aus statischen Ressourcen entfernen, Header des Typs "Vary: Accept-Encoding" angeben
- Schon fertig!** Für diese Regeln gibt es keine Vorschläge, da auf dieser Seite die bewährten Vorgehensweisen bereits befolgt werden. Sehr gut!

Bevor wir uns allerdings weiter mit Google Page Speed beschäftigen, sollten Sie bei WordPress alle nicht benötigten Plugins löschen, denn diese sind meist recht performancelastig.

Bilder optimieren mit Google Page Speed

Wie Sie im obigen Bild erkennen können, schlägt Google eine Optimierung der Bilder vor und zeigt auch an, um wie viel Kilobyte die Bilder ohne Qualitätsverlust optimiert werden können. Lassen Sie

sich die optimierte Version anzeigen und tauschen Sie diese mit der alten Version auf Ihrem FTP-Server aus. Da es bei vielen Bildern relativ aufwändig sein kann, gibt es ein Plugin dafür. Dieses Plugin heißt <http://optimus.io/> und bewirkt, dass alle Bilder vor dem hochladen entsprechend verlustfrei optimiert werden. So können Sie leicht wichtige Ladezeit einsparen.

Gzip und Cachingfunktion nutzen

Um die Ladezeit von WordPress deutlich zu beschleunigen, sollten Sie ein Plugin wie W3 Total Cache nutzen. Dieses sorgt dafür, dass Dateien nicht jedes Mal neu geladen werden müssen, sondern im Cache abgelegt werden. Weiterhin können größere Dateien wie .css oder .js Dateien komprimiert ausgegeben werden. so wird aus einer 100 KB großen .css-Datei eine 20 KB Datei.

Ist das Plugin installiert, können Sie die Page Speed Analyse nochmals durchlaufen lassen um zu sehen, ob die Änderungen übernommen worden sind.

Falls es Probleme gibt, sollten Sie einmal Ihre .htaccess Datei aufrufen und ihre Einträge überprüfen.

```
# BEGIN WordPress
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteRule ^index\.php$ - [L] RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [L] </IfModule>
# END WordPress
# BEGIN Gzip Compression
AddEncoding gzip .gz
<filesmatch "\.php\.gz$">
AddType "text/php" .gz
</filesmatch>
<filesmatch "\.js\.gz$">
AddType "text/javascript" .gz
</filesmatch>
<filesmatch "\.css\.gz$">
AddType "text/css" .gz
</filesmatch>
#
<ifmodule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteCond %{HTTP:Accept-encoding} gzip
RewriteCond %{REQUEST_FILENAME} \.(js|css|php)$
RewriteCond %{REQUEST_FILENAME}.gz -f
RewriteRule ^(.*)$ $1.gz [QSA,L] </ifmodule>
# END Gzip Compression
#
# BEGIN W3TC Browser Cache
<IfModule mod_mime.c>
AddType text/css .css
AddType text/x-component .htc
```

AddType application/x-javascript .js
AddType application/javascript .js2
AddType text/javascript .js3
AddType text/x-js .js4
AddType text/html .html .htm
AddType text/richtext .rtf .rtx
AddType image/svg+xml .svg .svgz
AddType text/plain .txt
AddType text/xsd .xsd
AddType text/xsl .xsl
AddType text/xml .xml
AddType video/asf .asf .asx .wax .wmv .wmx
AddType video/avi .avi
AddType image/bmp .bmp
AddType application/java .class
AddType video/divx .divx
AddType application/msword .doc .docx
AddType application/vnd.ms-fontobject .eot
AddType application/x-msdownload .exe
AddType image/gif .gif
AddType application/x-gzip .gz .gzip
AddType image/x-icon .ico
AddType image/jpeg .jpg .jpeg .jpe
AddType application/json .json
AddType application/vnd.ms-access .mdb
AddType audio/midi .mid .midi
AddType video/quicktime .mov .qt
AddType audio/mpeg .mp3 .m4a
AddType video/mp4 .mp4 .m4v
AddType video/mpeg .mpeg .mpg .mpe
AddType application/vnd.ms-project .mpp
AddType application/x-font-otf .otf
AddType application/vnd.oasis.opendocument.database .odb
AddType application/vnd.oasis.opendocument.chart .odc
AddType application/vnd.oasis.opendocument.formula .odf
AddType application/vnd.oasis.opendocument.graphics .odg
AddType application/vnd.oasis.opendocument.presentation .odp
AddType application/vnd.oasis.opendocument.spreadsheet .ods
AddType application/vnd.oasis.opendocument.text .odt
AddType audio/ogg .ogg
AddType application/pdf .pdf
AddType image/png .png
AddType application/vnd.ms-powerpoint .pot .pps .ppt .pptx
AddType audio/x-realaudio .ra .ram
AddType application/x-shockwave-flash .swf
AddType application/x-tar .tar
AddType image/tiff .tif .tiff
AddType application/x-font-ttf .ttf .ttc
AddType audio/wav .wav
AddType audio/wma .wma
AddType application/vnd.ms-write .wri

```
AddType application/vnd.ms-excel .xla .xls .xlsx .xlt .xlw
AddType application/zip .zip
</IfModule>
<IfModule mod_expires.c>
ExpiresActive On
ExpiresByType text/css A31536000
ExpiresByType text/x-component A31536000
ExpiresByType application/x-javascript A31536000
ExpiresByType application/javascript A31536000
ExpiresByType text/javascript A31536000
ExpiresByType text/x-js A31536000
ExpiresByType text/html A3600
ExpiresByType text/richtext A3600
ExpiresByType image/svg+xml A3600
ExpiresByType text/plain A3600
ExpiresByType text/xsd A3600
ExpiresByType text/xsl A3600
ExpiresByType text/xml A3600
ExpiresByType video/asf A31536000
ExpiresByType video/avi A31536000
ExpiresByType image/bmp A31536000
ExpiresByType application/java A31536000
ExpiresByType video/divx A31536000
ExpiresByType application/msword A31536000
ExpiresByType application/vnd.ms-fontobject A31536000
ExpiresByType application/x-msdownload A31536000
ExpiresByType image/gif A31536000
ExpiresByType application/x-gzip A31536000
ExpiresByType image/x-icon A31536000
ExpiresByType image/jpeg A31536000
ExpiresByType application/json A31536000
ExpiresByType application/vnd.ms-access A31536000
ExpiresByType audio/midi A31536000
ExpiresByType video/quicktime A31536000
ExpiresByType audio/mpeg A31536000
ExpiresByType video/mp4 A31536000
ExpiresByType video/mpeg A31536000
ExpiresByType application/vnd.ms-project A31536000
ExpiresByType application/x-font-otf A31536000
ExpiresByType application/vnd.oasis.opendocument.database A31536000
ExpiresByType application/vnd.oasis.opendocument.chart A31536000
ExpiresByType application/vnd.oasis.opendocument.formula A31536000
ExpiresByType application/vnd.oasis.opendocument.graphics A31536000
ExpiresByType application/vnd.oasis.opendocument.presentation A31536000
ExpiresByType application/vnd.oasis.opendocument.spreadsheet A31536000
ExpiresByType application/vnd.oasis.opendocument.text A31536000
ExpiresByType audio/ogg A31536000
ExpiresByType application/pdf A31536000
ExpiresByType image/png A31536000
ExpiresByType application/vnd.ms-powerpoint A31536000
ExpiresByType audio/x-realaudio A31536000
```

```
ExpiresByType image/svg+xml A31536000
ExpiresByType application/x-shockwave-flash A31536000
ExpiresByType application/x-tar A31536000
ExpiresByType image/tiff A31536000
ExpiresByType application/x-font-ttf A31536000
ExpiresByType audio/wav A31536000
ExpiresByType audio/wma A31536000
ExpiresByType application/vnd.ms-write A31536000
ExpiresByType application/vnd.ms-excel A31536000
ExpiresByType application/zip A31536000
</IfModule>
#
<IfModule mod_deflate.c>
<IfModule mod_headers.c>
Header append Vary User-Agent env=!dont-vary
</IfModule>
AddOutputFilterByType DEFLATE text/css text/x-component application/x-
javascript application/javascript text/javascript text/x-js text/html
text/richtext image/svg+xml text/plain text/xsd text/xsl text/xml image/x-
icon application/json
<IfModule mod_mime.c>
# DEFLATE by extension
AddOutputFilter DEFLATE js css htm html xml
</IfModule>
</IfModule>
<FilesMatch "\.(css|htc|js|js2|js3|js4|CSS|HTC|JS|JS2|JS3|JS4)$">
FileETag MTime Size
<IfModule mod_headers.c>
Header set Pragma "public"
Header append Cache-Control "public"
</IfModule>
</FilesMatch>
<FilesMatch
"\.(html|htm|rtf|rtx|svg|svgz|txt|xsd|xsl+xml|HTML|HTM|RTF|RTX|SVG|SVGZ|TXT|X
SD|XSL|XML)$">
FileETag MTime Size
<IfModule mod_headers.c>
Header set Pragma "public"
Header append Cache-Control "public"
</IfModule>
</FilesMatch>
<FilesMatch
"\.(asf|asx|wax|wmv|wmx|avi|bmp|class|divx|doc|docx|eot|exe|gif|gz|gzip|ico|j
pg|jpeg|jpe|json|mdb|mid|midi|mov|qt|mp3|m4a|mp4|m4v|mpeg|mpg|mpe|mpp|otf|odb
|odc|odf|odg|odp|ods|odt|ogg|pdf|png|pot|pps|ppt|pptx|ra|ram|svg|svgz|swf|tar
|tif|tiff|ttf|ttc|wav|wma|wri|xla|xls|xlsx|xlt|xlw|zip|ASF|ASX|WAX|WMV|WMX|AV
I|BMP|CLASS|DIVX|DOC|DOCX|EOT|EXE|GIF|GZ|GZIP|ICO|JPG|JPEG|JPE|JSON|MDB|MID|M
IDI|MOV|QT|MP3|M4A|MP4|M4V|MPEG|MPG|MPE|MPP|OTF|ODB|ODC|ODF|ODG|ODP|ODS|ODT|O
GG|PDF|PNG|POT|PPS|PPT|PPTX|RA|RAM|SVG|SVGZ|SWF|TAR|TIF|TIFF|TTF|TTC|WAV|WMA|
WRI|XLA|XLS|XLSX|XLT|XLW|ZIP)$">
FileETag MTime Size
```

```

<IfModule mod_headers.c>
Header set Pragma "public"
Header append Cache-Control "public"
</IfModule>
</FilesMatch>
# END W3TC Browser Cache
# BEGIN W3TC Page Cache core
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteCond %{HTTP:Accept-Encoding} gzip
RewriteRule .* - [E=W3TC_ENC:gzip] RewriteCond %{REQUEST_METHOD} !=POST
RewriteCond %{QUERY_STRING} =""
RewriteCond %{REQUEST_URI} \/$
RewriteCond %{HTTP_COOKIE} !(comment_author|wp\ -
postpass|w3tc_logged_out|wordpress_logged_in|wptouch_switch_toggle) [NC]
RewriteCond "%{DOCUMENT_ROOT}/wp-
content/cache/page_enhanced/%{HTTP_HOST}/%{REQUEST_URI}/_index.html%{ENV:W3TC
_ENC}" -f
RewriteRule .* "/wp-
content/cache/page_enhanced/%{HTTP_HOST}/%{REQUEST_URI}/_index.html%{ENV:W3TC
_ENC}" [L] </IfModule>
# END W3TC Page Cache core

```

Diese .htaccess-Datei sorgt dafür, dass das Komprimierungsverfahren gzip angewendet wird, Browsercaching genutzt wird und den Dateien ein Ablaufdatum gegeben wird, sodass diese nur neu geladen werden, wenn eine Änderung vorhanden ist. Diese genannten Verfahren können die Ladezeit um mehrere Sekunden reduzieren und den Seitenaufbau deutlich beschleunigt werden.

Sie können die obige .htaccess einmal probeweise in Ihrem WordPressverzeichnis einfügen (Vorher natürlich die alte sichern) und Page Speed nochmals durchlaufen lassen. Jetzt sollten Sie einen deutlichen Anstieg Ihrer Punkte erkennen und einige Optimierungsvorschläge sind erledigt.

Page Speed zeigt auch Dinge an, die man zwar optimieren kann, aber in Hinsicht auf den Seitenaufbau und die Ladegeschwindigkeit, keine wirklichen Unterschied machen. Dazu gehört beispielsweise der Punkt „Bildabmessungen festlegen“. Dieses kann man zwar ändern, aber je nach Template ist dies für Neulinge mit einem relativ großen Aufwand verbunden.

Sollten Sie nun weitere Änderungen vornehmen, empfiehlt es sich, den Cache bei WordPress zu löschen. Das Plugin W3 Total Cache hat hierfür den Button „Performance -> Empty all Caches“. Die Browserdaten sollten dann ebenfalls gelöscht werden.

Nutzen Sie Page Speed mit Firefox, so können Sie auch einmal die Variante unter Google Chrome nutzen, denn diese zeigt erfahrungsgemäß mehr zu optimierende Dinge an, als bei Firefox.

Um Ihre Ladegeschwindigkeit zu messen, bietet die Seite Pagespeed.de eine einfache Möglichkeit hierzu. Geben Sie dort Ihre Internetadresse ein und starten Sie den Test.

Im Grunde genommen sind alle Optimierungen schnell erledigt und bieten eine gute Möglichkeit die Performance der eigenen Seite mit wenigen Schritten enorm zu steigern und WordPress somit schneller zu machen.

Viele Dinge sind eher als Spielerei zu betrachten, aber Gzip-Komprimierung, Optimierung der Bilder und Caching sollten genutzt werden.

Für bestehende, ältere Blogs empfiehlt es sich, einmal das Plugin „WP-Optimize“ anzusehen. Dieses löscht unnötige Daten und Sicherungskopien von Artikeln aus Ihrer Datenbank und entschlannt diese auf beachtliche Weise.