

Kurs-Serie: Wordpress Themes erstellen

*Suchmaschinenoptimierung, WP als CMS, Navigations-Alternativen
und ein paar abschließende Tipps*

von **Vladimir Simovic**

akademie.de

akademie.de asp GmbH & Co. Betriebs- & Service KG

Lehrter Straße 16-17, 10557 Berlin

Tel: (030) 61655-0 Fax: (030) 61655-120

<http://www.akademie.de> E-Mail: info@akademie.de

Online auf [akademie.de](http://workshops.akademie.de):

<http://workshops.akademie.de/direkt?pid=68631>

Inhaltsverzeichnis

Suchmaschinenoptimierung, WP als CMS, Navigations-Alternativen und ein paar abschließende Tipps	3
Die Suchmaschinenoptimierung Ihres Projektes	3
wpSEO: komfortabel Ihre WordPress-Installation für Suchmaschinen optimieren	16
WordPress als "klassisches" CMS	18
Weitere Möglichkeiten um die Navigation umzusetzen	22
Freie bzw. öffentliche Themes erstellen	27
Über akademie.de	30

Suchmaschinenoptimierung, WP als CMS, Navigations-Alternativen und ein paar abschließende Tipps

Sie haben nun in den ersten Kurs-Teilen der Wordpress Themes Kurs-Serie ein HTML-Template an Hand einer grafischen Vorlage erstellt, ein einfaches WordPress-Theme realisiert und dieses erweitert.

Nun geht es darum, einige grundlegende Sachen über die Suchmaschinenoptimierung für WordPress-Installationen zu lernen. Danach werden wir uns mit den Unterschieden "WordPress für Weblogs" und "WordPress als 'klassisches' CMS" auseinandersetzen.

Im vorletzten Abschnitt werde ich Ihnen zeigen, wie Sie alternative Navigationsmenüs umsetzen können und wo hier WordPress seine Grenzen hat.

Zum Schluss gebe ich Ihnen einige Tipps wenn Sie sich entscheiden WordPress-Themes für ein breites Publikum zu entwickeln.

Die Suchmaschinenoptimierung Ihres Projektes

Der Unterschied zwischen der On-Page- und der Off-Page-Optimierung

Prinzipiell gibt es zwei Arten von Suchmaschinenoptimierung: **On-Page-** und **Off-Page-Optimierung**. Die erste beschäftigt sich mit Maßnahmen, die die Website bzw. deren Aufbau und Struktur betreffen. Die zweite Art der Optimierung beschäftigt sich mit Maßnahmen jenseits der Website. Hierbei geht es hauptsächlich um die Erhöhung der Linkpopularität bzw. die Erhöhung der Anzahl der eingehenden Links. Wir werden uns im folgenden Abschnitt mit der On-Page-Optimierung beschäftigen.

"Sprechende" URLs und die XML-Sitemap

Auf noch erstaunlich vielen dynamischen Websites (egal ob Blogs, Foren oder Wikis) sieht man nicht-sprechende URLs im Einsatz, z. B. www.domain.de/?p=123&blubb=457⁵. Das ist sowohl für den Nutzer als auch für die Suchmaschine nicht sonderlich aussagekräftig. Besser ist stattdessen:

5) <http://www.domain.de/?p=123&blubb=457>

www.domain.de/garten/blumen/⁶. Hier erkennt der Besucher sofort worum es geht und die Suchmaschine hat schon zwei Keywords zum "Futtern" bekommen.

Daher empfehle ich Ihnen auf jeden Fall unter "Einstellungen" » "Permalinks", die sprechenden URLs zu aktivieren. Diese Funktion lässt sich sehr einfach einrichten, entfaltet aber eine große Wirkung.

Die zweite Maßnahme - die Erstellung einer XML-Sitemap - erleichtert dem Suchrobot die Indexierung der Seite und kann dadurch indirekt der Positionierung der Website auf die Sprünge helfen. Das passende Plugin dazu haben wir bereits im dritten Kurs-Teil dieser Serie beschrieben.

Den Seitentitel optimieren

Den Seitentitel zu optimieren ist eine sehr wichtige Maßnahme, wenn es darum geht, die Website für die Suchmaschinen zu optimieren, und daneben hilft diese Maßnahme auch den Besuchern. Es wird sich immer wieder zeigen, dass sehr viele Maßnahmen in Bezug auf die Suchmaschinenoptimierung auch in direkter oder indirekter Weise den Besuchern der Website zur Gute kommen.

Nun geht es darum zu klären, was genau unter dem Seitentitel verstanden wird.



Optimaler Seitentitel

Mit der oberen Abbildung möchte ich Ihnen zwei Dinge zeigen. Zum einem sehen Sie noch einmal genau, was mit Seitentitel gemeint ist (siehe Cursor)

6) www.domain.de/garten/blumen/

und zum anderen sehen Sie auch direkt ein gutes Beispiel ... gönnen Sie mir ruhig das bisschen Eigenlob :-).

! Tipp: Warum ist der Seitentitel aus der vorherigen Abbildung optimal?

Die Suchmaschinen bewerten im Seitentitel lediglich die ersten 60-65 Zeichen. Daher ist es wichtig, die Überschrift bzw. den Titel des einzelnen Blog-Artikels an den Anfang zu rücken, damit auch die relevanten Begriffe des Artikels auf jeden Fall in die Bewertung kommen.

Und wie erreicht man dieses Ergebnis? Eine Möglichkeit ist es sich mit einem Text- bzw. Code-Editor zu bewaffnen und folgendes in der *header.php* einzufügen:

```
1 <?php if (is_home()) { ?>
2     <title><?php bloginfo('name'); ?> - <?php
   bloginfo('description'); ?></title>
3 <?php } else { ?>
4     <title><?php wp_title('&raquo;', true, 'right');
   bloginfo('name'); ?></title>
5 <?php } ?>
```

Was erreicht dieser Code? Dieser Code besteht aus einem if-else-Konstrukt. Es wird abgefragt, ob man sich auf der Startseite befindet. Wenn ja, dann wird der Name des Blogs (<?php bloginfo('name'); ?>) gefolgt von dem Blog-Slogan (<?php bloginfo('description'); ?>) angezeigt, hier die passende Abbildung:



Seitentitel auf der Startseite

Befindet man sich allerdings nicht auf der Startseite, sondern auf einer Unterseite, soll zuerst der Titel (*wp_title()*), gefolgt von einem Trennzeichen und dem Blog-Titel erscheinen:



Seitentitel einer Unterseite

Und hier zur Verdeutlichung der Seitentitel der Kategorie "WordPress" in meinem Weblog:



Seitentitel einer Kategorie

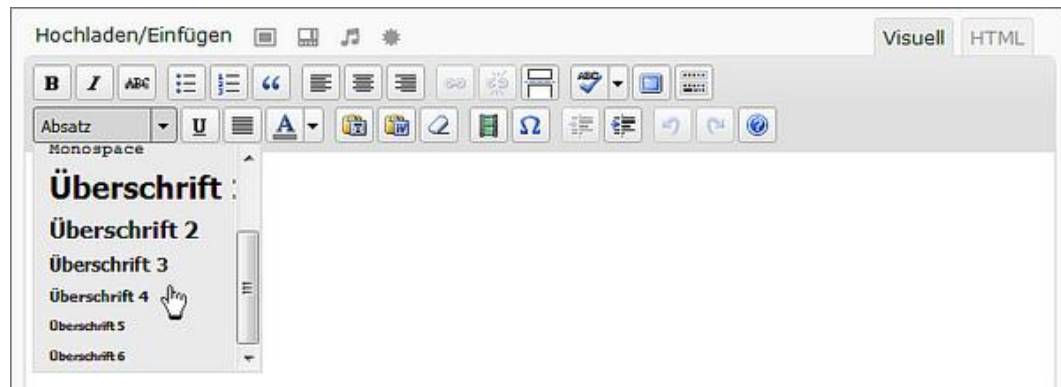
Durch die Anpassung des Seitentitels in der *header.php* wird der Seitentitel nach dem gleichen Prinzip auch auf anderen Unterseiten (Pages, Archiv etc.) so angezeigt wie in den letzten beiden Abbildungen.

Kommen wir noch einmal kurz zu dem neuen Code, speziell dem in der Zeile 10: `wp_title('»', true, 'right');`. Der Template-Tag `wp_title();`⁷ hat seit der Version 2.5 drei Parameter. Der erste ist für das Trennzeichen, der zweite Parameter bestimmt, ob der Titel ausgegeben wird und mit dem dritten Parameter positioniert man das Trennzeichen (rechts oder links vom Titel).

7) http://codex.wordpress.org/Template_Tags/wp_title

Überschriften und Zwischenüberschriften einsetzen

So wie man in einem Office-Dokument die einzelnen Bereiche durch Überschriften einleitet bzw. einen Text strukturiert, so kann man dies auch in einem (X)HTML-Dokument tun ... und WordPress nutzt XHTML-Dokumente. Es nimmt die Inhalte - welche Sie verfasst haben - aus der Datenbank und generiert dynamisch eine (X)HTML-Seite.



Auswahl von Überschriften

Für die Überschriften sind in (X)HTML die Elemente `h1`-`h6` zuständig. Wobei `h1` die Überschrift mit höchster und `h6` mit der niedrigsten Ordnung ist. Verwenden Sie die Überschriften semantisch richtig. Das heißt, Sie sollen eine bestimmte Überschrift genau dort verwenden, wo sie hingehört, also nicht `h2` wählen, wo `h3` richtig wäre, weil Sie die große und fette Schrift von `h2` schön finden.

(X)HTML ist entgegen anders lautender Gerüchte **keine** Programmiersprache (das ist z. B. C++), keine Seitenbeschreibungssprache (das ist z. B. PDF) und auch keine Textverarbeitung (dies versucht MS Word zu sein). (X)HTML ist eine **Auszeichnungssprache**: Sie zeichnen einzelne Bereiche aus und geben somit dem gesamten Dokument eine Struktur und dem Inhalt die Bedeutung. Wie das dann aussieht ist (X)HTML schnurz-piep-egal. Um das visuelle - Layout und Design - kümmert sich CSS. (X)HTML und CSS sind gute Teamplayer und erledigen ihre Bereiche sehr gut, daher ist es nicht sinnvoll (X)HTML die Aufgaben von CSS zu übertragen.

Wenn Ihnen also die Schriftgröße von `h3` zu klein ist, zeichnen Sie bitte die entsprechende Überschrift nicht mit `h2` aus. Lassen Sie `h3` dort stehen und notieren Sie in der CSS-Datei z. B. folgendes: `h3 {font-size: 120%;}`.

Noch einmal zurück zu den Überschriften. Bei längeren Texten - egal ob in einem Blog-Beitrag oder auf einer Page (Seite) - ist es sinnvoll, den Text mit Zwischenüberschriften zu gliedern. Das hilft dem Leser und ist gut für die Suchmaschinenoptimierung.

Allerdings gilt es zu beachten, dass z. B. das Default-Theme die Überschrift erster Ordnung (h1) bereits für den Schriftzug im Kopfbereich und die Überschrift zweiter Ordnung (h2) für die Überschriften bzw. Titel der Blog-Beiträge und der Pages "reserviert" hat. Somit stehen Ihnen h3-h6 zur Verfügung, aber die vier Arten von Überschriften sind mehr als ausreichend um den Inhalt eines Beitrages oder einer Page zu strukturieren.

Diese Struktur kann allerdings von Theme zu Theme abweichen, daher lohnt es sich, in den Template-Dateien nachzusehen, welche Überschriften bereits "reserviert" sind.

Es gab vor knapp zwei Jahren Anzeichen dafür, dass Google der Überschrift dritter Ordnung (h3) einen recht hohen Wert beigemessen hat, weil diese Überschrift in sehr vielen Blog-Templates die Zwischenüberschriften im Inhalt sind (also relevant für den Inhalt) - und weil sehr viele Webmaster es übertrieben haben, und die ersten beiden Überschriften (h1 und h2) mit relevanten Schlüsselwörtern vollgestopft hatten.

Schlüsselwörter in der Verlinkung unterbringen

Viele Webmaster verlinken innerhalb der eigenen Website (interne Links) zu wenig und/oder nicht optimal. Verlinken Sie innerhalb Ihrer Website überall dort, wo Sie denken, dass es den Besuchern helfen könnte.

Betreiben Sie ein Weblog, senden Sie doch einen Pingback vom neuen auf den alten Beitrag, wenn Sie den alten Beitrag erwähnen. Der Leser des neuen Beitrags weiß dann, dass es einen älteren Beitrag zu dem Thema gibt und der Leser des alten Beitrages sieht in den Kommentaren den Pingback und weiß, dass es zu dem Thema auch einen aktuellen Beitrag gibt.

So wie Sie mit dieser Maßnahme dem Besucher helfen, helfen Sie auch der Suchmaschine, diesen Beitrag zu finden.

Achten Sie bei der Verlinkung außerdem darauf, wichtige Schlüsselbegriffe direkt in den Linktext aufzunehmen. Hier ein sehr unglückliches Beispiel, welches man leider viel zu häufig sieht:



Beispiel:

Um den neuen Artikel zur gesetzlichen Versicherung zu lesen, klicken Sie [hier](#)⁸.

8) -

Viel besser ist folgendes:

Beispiel:

Lesen Sie den [neuen Artikel zur gesetzlichen Versicherung](#)⁹.

Diese Verlinkung ist für den Leser viel logischer und in punkto Suchmaschinenoptimierung ist es viel besser, da die Suchmaschinen es sehr hoch bewerten, wenn die relevanten Schlüsselwörter (Keywords) im Link-Text auftauchen. Allerdings sollten Sie hier, wie in allen Bereichen der Suchmaschinenoptimierung, maßvoll vorgehen. Wenn Sie übertreiben und in den Verlinkungen zu viele Keywords unterbringen, verärgern Sie nicht nur die Besucher sondern unter Umständen auch [Tante Guugel](#)¹⁰.

Benennung von Dateien

Auch Dateien - sowohl Bilder als auch z. B. PDF-Dateien - kann man mit sprechenden Namen beglücken. Folgende Namen:

- wordpress-workshop.pdf
- wordpress-admin-ansicht.png

sind auf jeden Fall leserlicher und Aussagekräftiger als:

- neue-datei1.pdf
- neues-bild.png

und zwar nicht nur für den Besucher, sondern auch für die Suchmaschinen. Bedenken Sie, dass zumindest Google die PDF-Dateien indexiert und dass alle gängigen Suchmaschinen auch die Möglichkeit anbieten, nach Bildern (z. B. die Bilder-Suche von Google) zu suchen. Dafür sind aussagekräftige Namen sehr wichtig.

Das alt-Attribut bei den Bildern

Ich würde Ihnen hier gerne einen kleinen zusätzlichen Hinweis bezüglich der Bilder geben. An vielen Stellen ist da etwas von einem alt-Tag zu lesen. Das ist einfach **falsch!** Es gibt weder in HTML noch in XHTML einen alt-Tag, es gibt lediglich ein alt-Attribut. So ist ein Bild-Element aufgebaut:

9) -

10) <http://www.google.de>

```
1 
```

Das ganze im oberen Beispiel ist ein (X)HTML-Element, `img` ist der Tag, `alt` und `title` sind Attribute und innerhalb der Anführungszeichen befinden sich die Attributwerte. Schematisch ist ein (X)HTML-Element folgendermaßen aufgebaut:

```
1 <Element-Name Attribut="Wert"> Inhalt des Elements</Element-Name>
```

`<Element-Name Attribut="Wert">` ist hierbei der Start-Tag und `</Element-Name>` ist der End-Tag. Wobei zu erwähnen ist, dass manche Elemente - `
` für Zeilenumbruch, `` für Bilder, `<hr />` für die horizontale Trennungslinie etc. - nur aus einem Tag bestehen.

Dies nur am Rande, damit Sie mit tiefer gehendem Insider-Wissen glänzen können :-).

Und bevor ich es vergesse, der Inhalt bzw. der Wert des `alt`-Attributs ist sehr wichtig. Dieser erscheint, wenn das Bild aus diversen Gründen nicht geladen wird und bei "Programmen", die keine Bilder anzeigen können: Textbrowser, Screenreader und Suchmaschinen. Daher ist der Inhalt dieses Attributs sehr wichtig. Allerdings würde ich Ihnen nicht empfehlen, dort Kurzgeschichten unterzubringen oder mit Keywords zu spamen ... das kann nach hinten losgehen. Ein kurzer, aussagekräftiger Satz über das Bild ist ausreichend.

🔗 Aufgabe: Fragen zur Selbstprüfung

1. Wie sieht die Struktur eines optimalen Seitentitels aus? Und warum?
2. Was ist der Unterschied zwischen dem `alt`-Attribut und dem `alt`-Tag? Ich bin zuversichtlich, dass Sie gegen eine Fangfrage nichts einzuwenden haben. :-)
3. Nennen Sie Beispiele für einen "guten" und einen "schlechten" Seitentitel und Dateinamen.

Duplicate Content verhindern bzw. eindämmen

Unter [Duplicate Content](#)¹¹ versteht man, wenn identischer Inhalt unter verschiedenen URLs erscheint. Das kann z. B. passieren, wenn man als Autor

11) <http://googlewebmastercentral-de.blogspot.com/2008/09/die-duplicate-content-penalty.html>

den gleichen Inhalt auf zwei verschiedenen Websites veröffentlicht, wenn jemand den Inhalt Ihrer Website klaut und als eigenen veröffentlicht oder auch einfach innerhalb des eigenen Weblogs.

Dies passiert in Weblogs und anderen dynamischen Websites sehr schnell. Zum Beispiel taucht ein Blog-Artikel sowohl auf der Startseite, als auch unter dem eigenen Permalink, als auch in weiteren Übersichten auf: Kategorie-Übersicht, Suchergebnisse etc.

Das Problem an der ganzen Sache ist, dass wenn ein identischer Inhalt auf mehreren Adressen auftaucht, dann müssen die Suchmaschinen entscheiden welche URL sie indizieren und welche Adressen sie als Dubletten ignorieren sollen. Daher könnte es z. B. sein, dass ein Artikel von Ihnen nicht unter seinem Permalink, sondern mit der URL der Kategorieübersicht indiziert wird.

Was kann man also gegen das Problem von Duplicate Content (DC) unternehmen? Hier gibt es mehrere unterschiedliche Vorgehensweisen. Zum einen steht dem WordPress-Nutzer **eine Reihe von SEO-Plugins** zur Verfügung, wie z. B. wpSEO oder izioSEO. Diese Plugins werden einmal konfiguriert und verrichten dann ihre Arbeit automatisch. Sie teilen u. a. den Suchmaschinen mit welche URL eines Inhaltes ist die relevante Adresse.

Allerdings kann man auch mit ganz einfachen Mittel das Problem entschärfen. Speziell bei längeren Artikeln empfiehlt es sich den Beitrag zu trennen und zwar in einen einleitenden Teil und den Rest. Auf den Übersichtsseiten wird lediglich der einleitende Teil gezeigt und zum Rest gelangt man durch das Anklicken des **"weiterlesen..."-Links**.

Mit dieser einfachen Maßnahme haben Sie zwei Sachen erreicht. Zum einen ist die DC-Problematik etwas entschärft - unterschiedlicher Inhalt auf den Übersichten und der Einzelansicht - und zum anderen kommt das den Lesern zu Gute wenn nicht der komplette Artikel von z. B. neun DIN-A4-Seiten in der Übersicht auftaucht.

Eine weitere einfache Maßnahme kann man mit Hilfe von Conditional Tags erreichen. Hier der Code, welchen man im Kopfbereich (<head></head>) einfügen muss. In den allermeisten WordPress-Themes wäre dies in der *header.php*.

```
1 <?php
2
3 if (is_archive() or is_search()) {
4
5     echo "<meta name=\"robots\" content=\"noindex, follow\" />\n";
6 }
```

```

7 }
8
9 ?>

```

Befindet man sich auf einer Übersichts- oder Archivseite - Kategorie-Übersicht, Jahresarchiv oder Suchergebnis-Seite - dann wird folgender **Meta-Tag** eingefügt:

```

1 <meta name="robots" content="noindex, follow" />

```

Das signalisiert den Suchmaschinen: *folge bitte allen Links auf dieser Seite, indiziere aber bitte nicht den Inhalt dieser Seite.*

Eine weitere Maßnahme ist das Erstellen von neuen Template-Dateien. Die Template-Dateien *category.php* und *search.php* sind zuständig für die Ausgabe der jeweiligen Übersichtsseite. Als Grundlage für die beiden Dateien nutze ich die *index.php*, welche ich an wenigen Stellen modifiziert habe. Folgendermaßen könnte die *category.php* ausschauen:

```

1 <?php get_header(); ?>
2
3     <div id="inhalt">
4
5         <h2>Die Kategorie "<?php single_cat_title(); ?>"</h2>
6
7         <div class="kategorie-info">
8             <p>Hier sind alle Beiträge aufgelistet, die in die
9             Kategorie <strong><?php single_cat_title(); ?></strong> einsortiert
10            wurden.</p>
11
12            <p>Du hast die Möglichkeit den <a href="<?php echo
13            get_category_link($cat);?>feed/" class="fett">RSS-Feed</a> speziell
14            nur für diese Kategorie zu abonnieren.</p>
15        </div>
16
17    <?php if (have_posts()) : while (have_posts()) : the_post(); ?>
18
19        <h2><a href="<?php the_permalink() ?>" rel="bookmark"
20        title="Der Verweis (Permalink) zu: <?php the_title(); ?>"><?php
21        the_title(); ?></a></h2>
22
23        <div class="beitrags-info">Von <?php the_author() ?> am
24        <?php the_time('d. F Y'); ?> um <?php the_time('H:i'); ?> <?php
25        edit_post_link(); ?></div>

```

```

16
17         <div class="der-beitrag">
18             <?php the_excerpt(); ?>
19         </div> <!-- Ende des jeweiligen Beitrags -->
20
21         <div class="feedback">
22             <?php comments_popup_link('Kommentare (0)',
23 'Kommentare (1)', 'Kommentare (%)'); ?>
24         </div>
25         <!-- Ende des jeweiligen Beitrags -->
26 <?php comments_template(); /* Der Kommentarbereich */ ?>
27
28 <?php endwhile; else: ?>
29     <p>Tut mir leid, es wurde kein passender Beitrag
30 gefunden.</p>
31 <?php endif; ?>
32
33     <p><?php posts_nav_link(' - ', '« Zurückblättern',
34 'Weiterblättern »'); ?></p>
35
36 </div><!-- /#inhalt -->
37
38 <?php get_sidebar(); ?>
39 <?php get_footer(); ?>

```

Gegenüber der *index.php* gibt es zwei konkrete Änderungen. Zum einen habe ich eine Informationsbox vor der Auflistung der Beiträge eingebunden. In der Info-Box wird zuerst der Name der Kategorie (*single_cat_title();*) ausgegeben und darunter folgt der Link zu dem Newsfeed der jeweiligen Kategorie: `echo get_category_link($cat);?>feed/` ergibt dann z. B. www.domain.de/kategorie/allgemein/feed/.

Die zweite Änderung betrifft die Ausgabe der einzelnen Artikel. Diese werden nicht mehr in voller Länge ausgegeben, sondern lediglich ein Auszug (*the_excerpt();*). Der Auszug besteht standardmäßig aus maximal 55 Wörtern des Original-Artikels und er wird ohne Formatierungen und Auszeichnungen (engl.: plain text) ausgegeben.

Sie können aber natürlich, im Backend auch einen individuellen Auszug verfassen. Das Modul dafür befindet sich direkt unterhalb des Editorfeldes für den Inhalt eines Artikels.

So schaut das ganze dann aus:



Die neue Kategorie-Übersicht

Damit das ganze auch visuell umgesetzt wird, müssen weitere Anpassungen an der *style.css* stattfinden:

```

1  .fett {
2      font-weight: bold;
3  }
4  ...
5  #inhalt .kategorie-info {
6      background: #dcdecf no-repeat right bottom;
7      padding: 10px 12px; border: 1px solid #bbc0a2; margin: 7px 0
8      17px 0;
9  }
10 #inhalt .kategorie-info p {
11     margin: 3px 0;
12 }

```

Nach dem gleichen Prinzip erstellt man auch das Such-Template (*search.php*). Als Vorlage dient uns hierbei die soeben erstellte *category.php*.

```

1  <?php get_header(); ?>
2

```

```

3         <div id="inhalt">
4
5             <h2>Deine Suche nach "<?php the_search_query(); ?>"</h2>
6             <div class="kategorie-info"><p>Falls deine Suche nach
<strong><?php the_search_query(); ?></strong> kein oder kein
zufriedenstellendes Ergebnis geliefert hat, dann versuche es im
<strong><a href="/archiv/">Archiv</a></strong>. Dort findest du
eine Tag-Wolke, einen Monats-Archiv und eine Auflistung der 25
letzten Beiträge.</p></div>
7
8 <?php if (have_posts()) : while (have_posts()) : the_post(); ?>
9
10         <h2><a href="<?php the_permalink() ?>" rel="bookmark"
title="Der Verweis (Permalink) zu: <?php the_title(); ?>"><?php
the_title(); ?></a></h2>
11         <div class="beitrags-info">Von <?php the_author() ?> am
<?php the_time('d. F Y'); ?> um <?php the_time('H:i'); ?> <?php
edit_post_link(); ?></div>
12
13         <div class="der-beitrag">
14             <?php the_excerpt(); ?>
15         </div> <!-- Ende des jeweiligen Beitrags -->
16
17         <div class="feedback">
18             <?php comments_popup_link('Kommentare (0)',
'Kommentare (1)', 'Kommentare (%)'); ?>
19         </div>
20         <!-- Ende des jeweiligen Beitrags -->
21
22 <?php comments_template(); /* Der Kommentarbereich */ ?>
23
24 <?php endwhile; else: ?>
25         <p>Tut mir leid, es wurde kein passender Beitrag
gefunden.</p>
26 <?php endif; ?>
27
28         <p><?php posts_nav_link(' - ', '« Zurückblättern',
'Weiterblättern »'); ?></p>
29
30     </div><!-- #inhalt -->
31
32 <?php get_sidebar(); ?>
33 <?php get_footer(); ?>

```


Mit Hilfe des Template-Tags `the_search_query()` wird das eingesetzte Suchwort ausgegeben. Falls die Suche für den Besucher kein zufrieden stellendes Ergebnis liefert, kann man ihn z. B. auf eine Archiv-Seite hinweisen. Diesen Hinweis setze ich auch auf www.perun.net¹² ein.

Mit den zwei neuen Template-Dateien haben Sie, wie schon weiter oben erwähnt, zwei Effekte erreicht. Zum einen haben Sie die DC-Problematik entschärft und zum anderen dem Besucher zusätzlichen Komfort verschafft: Hinweis auf die Kategorie und Hinweis auf das Archiv.

An diesem Beispiel sieht man deutlich, dass die Optimierungen für die Suchmaschinen und für die Besucher Hand in Hand gehen.

Kanonische URLs

Eine neue Möglichkeit um die DC-Problematik in den Griff zu bekommen verspricht der neue Attributs-Wert für das `link`-Element:

```
1 <link rel="canonical" href="http://www.perun.net" />
```

`rel="canonical"` ist [recht jung](http://www.perun.net)¹³ und damit kann man den gängigen Suchmaschinen (Google, MSN und Yahoo) "sagen" welche Seite bzw. welche URL zu bevorzugen ist, falls es im System DC geben sollte. Sofern Sie mindestens WordPress 2.9 einsetzen, müssen Sie sich zu diesem Thema keine Sorgen machen, weil WordPress automatisch im Kopfbereich der einzelnen Artikel das passende `rel="canonical"` einfügt.

wpSEO: komfortabel Ihre WordPress-Installation für Suchmaschinen optimieren

Bei der Erweiterung [wpSEO](http://www.wpseo.de/)¹⁴ von Sergej Müller handelt es sich um ein sehr nützliches Plugin, welches Ihnen hilft Ihre WordPress-Installation für Suchmaschinen zu optimieren.

Anmerkung:

Bei wpSEO handelt es sich um ein kostenpflichtiges Plugin, welches je nach Lizenz zwischen knapp 20 und knapp 100 Euro kostet. Sie können aber das Plugin zehn Tage lang kostenlos testen.

12) <http://www.perun.net>

13) <http://googlewebmastercentral-de.blogspot.com/2009/02/bestimmt-eure-kanonische-url.html>

14) <http://www.wpseo.de/>

Dieses Plugin ist nicht im offiziellen Verzeichnis hinterlegt und daher muss man es sich von der Plugin-Website herunterladen und das entpackte Paket nach `/wp-content/plugins/` hochladen: entweder über das FTP-Programm oder über die WordPress-Funktion.



Seriennummer eingeben

Nach der Aktivierung haben Sie die Möglichkeit den Lizenzschlüssel einzugeben, wenn Sie keinen Schlüssel eingeben funktioniert die Erweiterung dennoch: es startet die zehntägige Testphase.

Auf der Optionen-Seite der Erweiterung ("Einstellungen » wpSEO") haben Sie die Möglichkeit sehr viele Einstellungen zu tätigen. Sie haben die Möglichkeit den Aufbau des Seitentitels für Artikel, Seiten, Archive etc. zu steuern, Sie haben die Möglichkeit die Seiten-Beschreibungen und Seiten-Keywords zu steuern, Sie haben die Möglichkeit zu steuern welche Seiten des Weblogs indiziert werden sollen und welche nicht, um so genannten *Duplicate Content* (identischer Inhalt auf mehreren URLs) zu verhindern und vieles mehr.



"Doppelten Inhalt" vermeiden

Die Funktionalität des Plugins ist enorm und auf alle Punkte einzugehen würde den Rahmen dieses Abschnitts sprengen zumal es auf der Website des Autors unter [Dokumentation](#)¹⁵ eine sehr ausführliche Anleitung gibt.

15) <http://www.wpseo.de/manual/>

! Tipp: Weitere SEO-Plugins

Wer es gerne komplett kostenlos haben will, der kann sich das Plugin "[All in One SEO Pack](#)¹⁶" anschauen. Es handelt sich hierbei um ein Plugin, welches einen ähnlichen Funktionsumfang wie wpSEO zu bieten hat.

Auf der Website [TalkPress.de](#)¹⁷ gibt es einen recht ausführlichen Artikel, welcher die drei WordPress-Plugins wpSEO, All in One SEO Pack und izioSEO mit einander vergleicht.

WordPress als "klassisches" CMS

Dass man WordPress nicht nur für Weblogs, sondern auch für Webprojekte ohne Blog-Charakter einsetzen kann ist ein alter Hut.

Schon vor fast vier Jahren habe ich angefangen, mich mit dem Thema WordPress als CMS zu beschäftigen, und habe im [Juni](#)¹⁸ und [Juli](#)¹⁹ 2005 zwei Artikel zu diesem Thema in meinem Weblog verfasst. Mittlerweile habe ich bei mehr als einem Dutzend Kunden-Websites WordPress als CMS eingesetzt und weder die Kunden noch ich haben diese Entscheidung bereut ... im Gegenteil.

Aber warum bzw. wann sollte man WordPress als "klassisches" CMS einsetzen? Dafür gibt es viele Gründe. WordPress hat niedrige Anforderungen an einen Server, ist leicht zu installieren und zu pflegen und man kann auch anspruchsvolle Layouts problemlos umsetzen. Wenn Bedarf herrscht, kann man auf eine Fülle frei verfügbarer Plugins zurückgreifen, um benötigte Funktionen einzubauen. Zu guter Letzt haben Sie die Möglichkeit, noch zusätzlich auf ein sehr gutes Blog-System zurückzugreifen, wenn Sie sich entscheiden, Ihr Webprojekt noch um ein Weblog zu erweitern.

WordPress ist trotz seiner Schlankheit ein recht leistungsfähiges System und ist in der Bedienung, im Vergleich zu manchen Schwergewichten auf dem Markt, auch recht schnell zu erlernen.

Die Funktionsweise von WordPress als "klassisches" CMS

Die Funktionsweise von WordPress als "klassisches" CMS ist schnell erklärt. Alles was zu dem Blogkreislauf gehört - Artikel, Kategorien, Feed etc. - rückt

16) <http://wordpress.org/extend/plugins/all-in-one-seo-pack/>

17) <http://talkpress.de/artikel/seo-plugin-wordpress-vergleich-all-in-one-izioseo-wpseo>

18) <http://www.perun.net/2005/06/03/wordpress-als-cms-ein-beispiel/>

19) <http://www.perun.net/2005/07/09/wordpress-als-cms-zweites-beispiel/>

in den Hintergrund. Dafür rücken die Seiten und Seiten-Templates in den Vordergrund.

Unter "Einstellungen" » "Lesen" haben Sie die Möglichkeit, die Startseite und die Artikel-Seite festzulegen: sehr wichtige Funktionen wenn Sie WordPress für Projekte einsetzen möchten, welche keinen Blog-Charakter aufweisen.

Im Folgenden werde ich Ihnen zwei von sehr vielen Plugins vorstellen, die Ihnen den Umgang mit Seiten (Pages) erleichtern.

Simple Page Ordering: viele Seiten im Griff behalten

Die Erweiterung [Simple Page Ordering](http://wordpress.org/extend/plugins/simple-page-ordering/)²⁰ haben Sie bereits im dritten Teil dieser Kurs-Serie kennen gelernt. Damit können Sie einfach Seiten verschieben, verschachteln und neu anordnen. Aus Gründen der Vollständigkeit, erwähne ich sie hier noch einmal.

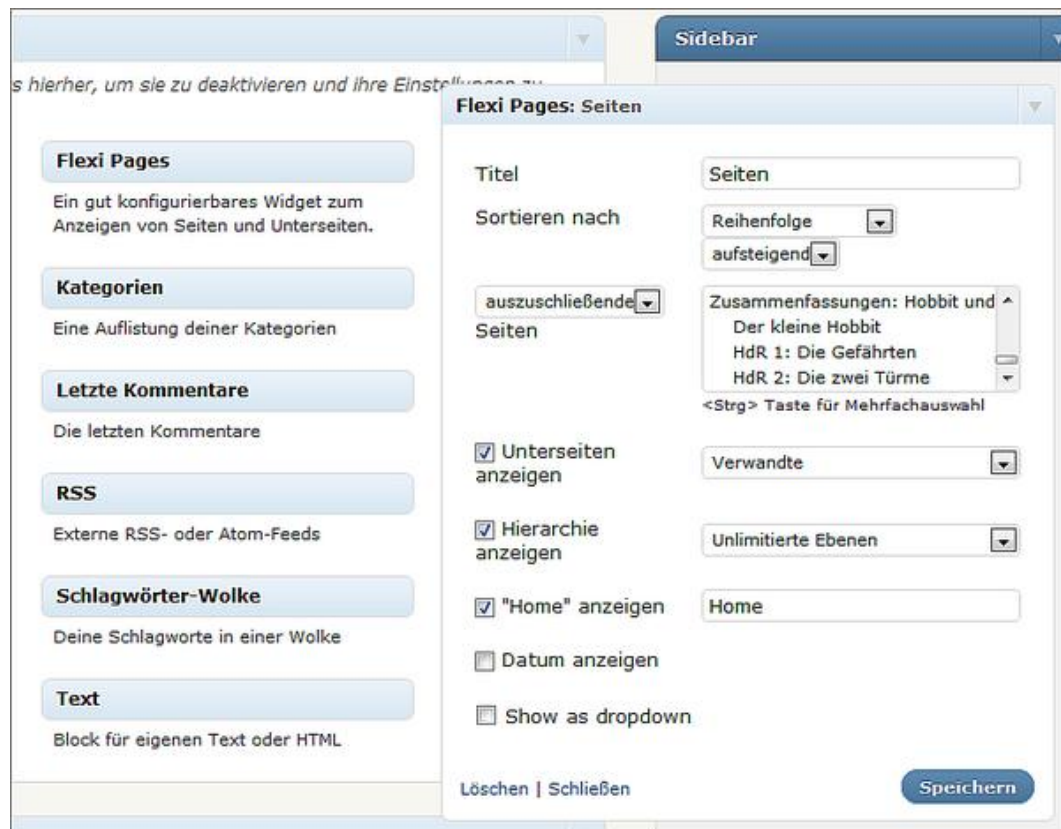
Flexi Pages Widget: Unterseiten verbergen oder auflisten

Mit der Erweiterung [Flexi Pages Widget](http://wordpress.org/extend/plugins/flexi-pages-widget/)²¹ haben Sie die Möglichkeit, die Unterseiten erst dann in der Seitenleiste auflisten zu lassen, wenn man sich auf der zugehörigen Hauptseite ("Eltern") oder auf einer benachbarten Unterseite ("Geschwister") befindet.

Für die automatische Installation geben Sie am besten "Flexi" als Suchbegriff ein um das Plugin im offiziellen Verzeichnis zu finden. Bei einer manuellen Installation müssen Sie den kompletten entpackten Ordner *flexi-pages-widget* nach */wp-content/plugins/* hochladen. Nach dem Aktivieren des Plugins wechseln Sie direkt nach "Design » Widgets" (dieses Plugin hat keine Optionen-Seite):

20) <http://wordpress.org/extend/plugins/simple-page-ordering/>

21) <http://wordpress.org/extend/plugins/flexi-pages-widget/>



Unterseiten ausblenden

Auf der linken Seite der Abbildung sehen Sie die verfügbaren Widgets (Erweiterungen für die dynamische Sidebar). Um ein Widget zu aktivieren, z. B. das gerade installierte und aktivierte Flexi Pages ziehen Sie es einfach mit der Maus in den rechten Bereich auf die Sidebar. Wenn Sie ein Widget erstmalig in diesen Bereich ziehen und dort "fallen lassen" öffnet es sich automatisch im Bearbeitungsmodus, so wie es in dem Screenshot abgebildet ist. Sie können dann die nötigen Einstellungen vornehmen und anschließend Speichern und das Widget-Fenster schließen. Möchten Sie zu einem späteren Zeitpunkt Einstellungen verändern, müssen Sie auf den kleinen Pfeil in der rechten oberen Ecke des jeweiligen aktiven Widgets klicken. Der Bearbeitungsmodus öffnet sich dann.

Seit der Version 2.8 ist es zudem möglich, Widgets, die man nicht mehr in der Sidebar haben möchte, zu deaktivieren ohne direkt auch die getätigten Einstellungen zu löschen. Dazu ziehen Sie das Widget aus der Sidebar hinaus in den unteren Bereich der Seite zu den "Inaktiven Widgets". Möchten Sie auch die Einstellungen löschen, ziehen Sie das Widget einfach in den Bereich "Verfügbare Widgets" oder klicken im Bearbeitungsmodus auf den Link "Entfernen".

Aber nun zurück zu dem frisch installierten Plugin *Flexi Pages*, das Ihnen beim Handling von Unterseiten behilflich sein soll.

Als erstes können Sie sich entscheiden, wie Sie die Navigation bzw. die Auflistung der Websites benennen wollen ("Titel"). Sie können dieses Feld auch leer lassen - dann gibt es logischerweise auch keinen Titel.

Darunter können Sie sich für einen Sortierungsmechanismus entscheiden. Zur Auswahl stehen unter anderem Sortierung nach der ID-Nummer der Seite, alphabetische Reihenfolge, dem Erstellungszeitpunkt, oder die Reihenfolge-Nummerierung. Daneben gibt es eine zweite Auswahl für absteigende und aufsteigende Reihenfolge.

Darunter haben Sie die Möglichkeit, bestimmte Seiten aus der Auflistung herauszunehmen oder nur bestimmte Pages einzufügen. Mit gedrückter Strg- bzw. Ctrl-Taste können Sie eine Mehrfachauswahl vornehmen.

Darunter gibt es fünf Checkboxes:

1. Unterseiten anzeigen: Hier können Sie wählen, ob und wenn ja welche Unterseiten angezeigt werden sollen: alle Unterseiten ("Alle"), auch die Unterseiten der gerade gewählten Hauptseite ("Verwandte") oder nur die Unterseiten der gerade ausgewählten Hauptseite, aber nicht deren Geschwister ("Nur direkt verwandte").
2. Hierarchie anzeigen: Hier kann man die Verschachtelungstiefe bestimmen bzw. bis zu welcher Tiefe es eine Auflistung geben soll.
3. "Home" anzeigen: Sie haben hier die Möglichkeit, einen Link zur Startseite zu generieren und diesen auch hier zu benennen. In unserem Beispiel ist das nicht notwendig, da wir bereits einen Link zur Startseite haben.
4. Datum anzeigen: Hier können Sie auswählen, ob Sie das Erstellungsdatum der Haupt- und Unterseiten angezeigt bekommen haben möchten.
5. Show as dropdown: Aktiviert man diese Funktion, so wird das komplette Menü in Form einer Drop-Down-Liste ausgegeben.

Anmerkung:

Bitte bedenken Sie, dass die meisten Themes-Autoren die dynamischen Sidebars so eingerichtet haben, dass mit der Aktivierung des Widgets der vorherige Inhalt der Sidebar deaktiviert wird. Deshalb sehen Sie in der oberen Abbildung nur den Inhalt des Flexi-Pages-Widgets.

Weitere Möglichkeiten um die Navigation umzusetzen

Im folgenden Abschnitt werde ich Ihnen einige Möglichkeiten aufzeigen, wie Sie ein Navigationsmenü aufbauen können, welches von dem Standard abweicht.

Plugins

Wie Sie mit Hilfe von Plugins in den Aufbau der Navigation eingreifen können, haben Sie bereits im vorherigen Abschnitt erfahren. Mit Hilfe des Plugins "[Flexi Pages Widget](#)²²" können Sie bequem bestimmen, welche Pages und vor allem wann sie im Menü auftauchen sollen.

Ein weiteres Plugin welches Sie verwenden können ist [Subpages Extended](#)²³. Die Bedienung dieser Erweiterung ist sehr einfach, wer dennoch nicht weiter kommt, kann sich den [folgenden Artikel](#)²⁴ und/oder das [folgende Video](#)²⁵ anschauen.

Code-Lösungen (1)

Recht häufig wird von den Kunden der Wunsch geäußert an einer Stelle, die Top-Level-Seiten ("Eltern" bzw. "Vorfahren" 1. Grades) anzuzeigen und anderswo auf der Seite, die untergeordneten Seiten ("Kinder" und "Enkel"). Mit folgendem Code zeigt man nur die Top-Level-Seiten an:

Code: Nur die Top-Level-Seiten anzeigen

```
1 <ul id="hauptnavi">
2     <?php
        wp_list_pages('sort_column=menu_order&title_li=&depth=1'); ?>
3 </ul>
```

Diesen Code würde man klassischerweise oben im Kopfbereich unterbringen. Der Wert 1 bei dem Parameter `depth` ist dafür verantwortlich, dass **nur** die Seiten der obersten Ebene aufgelistet werden.

Jetzt geht es darum an der gewünschten Stelle, zum Beispiel in der Sidebar, die untergeordneten Seiten anzuzeigen:

```
1 <?php
2 if($post->post_parent) { /*Die Seite ist eine "Kind-Seite"*/
3     wp_list_pages('sort_column=menu_order&title_li=&child_of=' .
        $post->post_parent);
```

22) [#_Nützliche_Plugins_für](#)

23) <http://wordpress.org/extend/plugins/subpages-extended/>

24) <http://www.perun.net/2012/02/19/wordpress-unterseiten-kontextabhaengig-anzeigen-lassen/>

25) <http://youtu.be/IK6sK-pOVYQ>


```

4 }
5 elseif(wp_list_pages("child_of=".$post->ID."&echo=0")) { /*Die
    Seite hat "Kinder"*/
6     wp_list_pages('sort_column=menu_order&title_li=&child_of='.$
    $post->ID);
7 }
8 ?>

```

Das obere Code-Beispiel ist dafür zuständig, auf den Seiten (Pages), die Unterseiten haben, diese auch als Listen auszugeben. Zuerst prüft der Code, ob die Seite, auf der man sich aktuell befindet, eine Unterseite ("Kind-Seite") ist. Wenn ja, dann werden hier alle Unterseiten der jeweiligen Eltern-Seite aufgelistet.

Anschließend wird geprüft, ob die aktuelle Seite selbst auch Unterseiten hat, wenn ja, dann werden auch diese Unterseiten aufgelistet.

Man könnte das auch in Klarsprache ausdrücken: "Wenn du ein Kind bist, dann zeige mir deine Geschwister. Wenn du Elternteil bist, dann zeige mir deine Kinder."

Code-Lösungen (2)

Um das folgende Beispiel nachzuvollziehen sind gute CSS-Kenntnisse notwendig.

Die Lösung basiert nicht auf dem uns bereits bekannten Template-Tag `wp_list_pages()`, sondern auf der Menüfunktion von WordPress: `wp_nav_menu()`.²⁶ Hat man diese Funktion aktiviert, dann erscheint im Adminbereich unter dem Menüpunkt *Design* ein weiterer Unterpunkt mit dem Namen *Menüs*. Dort kann man dann individuelle Navigationsmenüs per Klicken und Ziehen erstellen.

Das folgende Beispiel stammt von meiner [Herr-der-Ringe-Website](http://www.faszination-tolkien.de/)²⁷ und dort kann man das Live und in Farbe begutachten.

Zuerst aktivieren und definieren wir diese Funktion in der *functions.php*:

```

1 // Menü-Funktion aktivieren
2 function die_hauptnavi() {
3     register_nav_menus(
4         array(
5             'haupt_navi' => 'Die Hauptnavigation',
6         )
7     );

```

26) http://codex.wordpress.org/Function_Reference/wp_nav_menu

27) <http://www.faszination-tolkien.de/>

```

8 }
9
10 add_action( 'init', 'die_hauptnavi' );
11
12 // Abfragen ob ein Menü erstellt wurde, wenn nicht...
13 function meine_hauptnavi() {
14     if ( function_exists( 'wp_nav_menu' ) )
15         wp_nav_menu( 'menu=haupt_navi&container_class=pagemenu&fallback_cb=die_ersatznavi' );
16     else
17         die_ersatznavi();
18 }
19
20 // Die Definition der Ersatzlösung
21 function die_ersatznavi() {
22     wp_list_pages('title_li=&depth=1');
23 }

```

Damit haben Sie sowohl die Menüfunktion aktiviert als auch eine Ausweichlösung definiert, falls kein Menü definiert wurde. Im übrigen müssen Sie dies, wenn Sie das Standard-Theme von WordPress nutzen nicht machen. Dort wurde diese Funktionalität bereits aktiviert.

Im Theme selber, zum Beispiel in der *header.php*, wird das individuelle Menü, dann durch `<?php meine_hauptnavi(); ?>` eingebunden.

Im HTML-Quelltext schaut das ganze dann folgendermaßen aus:

```

1 <div class="pagemenu">
2     <ul id="menu-hauptnavi" class="menu">
3         <li id="menu-item-575" class="menu-item menu-item-type-
post_type menu-item-object-page menu-item-575"><a title="Bebilderte
Zusammenfassungen von Der Hobbit und Herr der Ringe" href="http://
www.faszination-tolkien.de/zusammenfassung-hobbit-herr-der-
ringe/">Zusammenfassungen</a>
4             <ul class="sub-menu">
5                 <li id="menu-item-612" class="menu-item menu-item-type-
post_type menu-item-object-page menu-item-612"><a href="http://
www.faszination-tolkien.de/zusammenfassung-hobbit-herr-der-ringe/
einstieg-in-tolkiens-mittelerde/">Einstieg in Tolkiens Mittelerde</
a></li>
6                 <li id="menu-item-582" class="menu-item menu-item-type-
post_type menu-item-object-page menu-item-582"><a href="http://
www.faszination-tolkien.de/zusammenfassung-hobbit-herr-der-ringe/
hobbit/">Der kleine Hobbit</a></li>
7                 ...

```

```

8         </ul>
9     </li>
10    <li id="menu-item-576" class="menu-item menu-item-type-
post_type menu-item-object-page menu-item-576"><a title="Die Ringe
der Macht" href="http://www.faszination-tolkien.de/die-ringe-der-
macht/">Die Ringe</a></li>
11    <li id="menu-item-577" class="menu-item menu-item-type-
post_type menu-item-object-page menu-item-577"><a title="Die
Geschichte von Mittelerde" href="http://www.faszination-tolkien.de/
geschichte-arda-mittelerde/">Geschichte</a>
12        <ul class="sub-menu">
13            ...
14        </ul>
15    </li>
16    ...
17 </ul>
18 </div>

```

Damit ganze noch übersichtlich bleibt, habe ich das Beispiel auf das Notwendige gekürzt.

Wie Sie sehen, liefert uns auch hierbei WordPress automatisch Werte für id- und class-Attribute und zwar für das Menü, für die Liste und für die Untermenüs. Damit haben wir Punkte, an denen wir mit CSS ansetzen können.

Und hier die CSS-Regeln:

```

1 #menu-hauptnavi {
2     list-style: none; margin: 0; padding: 0;
3     position: absolute; top: -86px; left: 0;
4 }
5
6 .pagemenu li {float: left; position: relative;}
7
8 .pagemenu a {
9     color: #eee; text-decoration: none;
10    float: left;
11    line-height: 40px;
12    padding: 0 8px; margin: 0 1px;
13 }
14
15 .pagemenu a:first-child {margin-left: none;}
16
17 .pagemenu a:hover, .pagemenu .current-menu-item
a, .pagemenu .current-menu-parent a {

```

```

18 background: #8b9b87 url(img/hover.png) repeat-x;
19 color: #2B3A27; font-weight: normal;
20 }
21
22 .pagemenu li:hover {background: #8b9b87 url(img/hover.png) repeat-
23 x;}
24
25 .pagemenu .sub-menu {
26 padding: 0 0 2px 0; margin: 0; list-style: none;
27 position: absolute; top: -999px;
28 background: #alad9e;
29 opacity: 0;
30 }
31
32 .sub-menu li {float: none; display: block; position: static;
33 padding-bottom: 1px;}
34
35 .pagemenu li:hover .sub-menu {
36 background: #8b9b87;
37 top: 40px;
38 opacity: 1;
39 -webkit-transition: opacity .4s ease-in-out; transition:
40 opacity .4s ease-in-out;
41 }
42
43 .pagemenu .sub-menu a {
44 padding: 3px 10px; margin: 0;
45 display: block; float: none;
46 color: #2b3a27; background: none;
47 line-height: 1.5; white-space: nowrap;
48 }
49
50 .pagemenu li ul a:hover {color: #eee; background: #425c3b;}

```

Das ganze schaut dann auf der HdR-Seite dann so aus:



Dropdownmenü im Einsatz

Um so ein Dropdownmenü zu realisieren, müssen Sie nicht mit `wp_nav_menu()` arbeiten, Sie können auch auf den alten Bekannten `wp_list_pages()` zurückgreifen. Sie müssen lediglich per CSS die Untermenüs in der Standardansicht verstecken und die sie erst beim `hover`, also beim Mouseover, sichtbar werden lassen.

Freie bzw. öffentliche Themes erstellen

Sie werden sich jetzt sicherlich fragen warum man sich die Mühe machen sollte ein WordPress-Theme zu erstellen und dieses anschließend zum kostenlosen Download anzubieten. Die Frage ist berechtigt, aber auch schnell zu beantworten.

Warum überhaupt ein freies WordPress-Theme erstellen?

Mit einem kostenlosen WordPress-Theme können Sie recht schnell ein breites Zielpublikum erreichen und beweisen, dass Sie das Thema beherrschen. Sie haben sich eine gute Referenz und eine zusätzliche Übungsgelegenheit geschaffen: also zwei "Fliegen" mit einer Klappe erledigt.

Das Erstellen von Themes für ein breites Publikum unterscheidet sich in einigen Punkten von dem Erstellen eines Themes für einen konkreten Kunden. Im folgenden Abschnitt werde ich Ihnen einige meiner Erfahrungswerte weitergeben.

Internationalisierung (Lokalisierung)

Da Sie ein breites Publikum erreichen wollen, sollten Sie das Theme international machen. Schauen Sie sich am besten dafür das Classic-Theme genauer an. Dort finden Sie an mehreren Stellen folgende Angaben, welche der Lokalisierung dienen:

Code: Erste Methode der Lokalisierung

```
1 <?php _e('Leave a comment'); ?>
```

oder:

Code: Zweite Methode der Lokalisierung

```
1 <?php comments_number(__('No Comments'), __('1 Comment'), __('%  
Comments')); ?>
```

Die zwei Methoden der Lokalisierung - `_e()` und `__()` - unterscheiden sich dadurch, dass die erste Methode eingesetzt wird, wenn der Begriff einfach und ohne Umwege übersetzt werden soll. Die zweite Methode wird eingesetzt, wenn der zu übersetzende Begriff Argument einer Funktion ist. Zum Beispiel als Parameter-Wert eines Template-Tags.

Diese beiden Arten der Lokalisierung dienen den Sprachdateien als "Anker" um zu wissen wo ein Begriff oder Satz übersetzt werden soll. Vorausgesetzt, die Sprachdatei wurde korrekt eingebunden und vorausgesetzt, für das entsprechende Wort oder den entsprechenden Satz gibt es überhaupt eine passende Übersetzung.

Aber das ist dann die Sorge der einzelnen Nutzer bzw. der Übersetzer. Sie als Themes-Autor haben Ihre Schuldigkeit getan und das Theme internationalisiert.

An alles denken

Im Gegensatz zu einem konkreten Projekt müssen Sie bei einem Theme für ein breites Publikum an alles denken. Sie sollten daher das Theme und vor allem die CSS-Datei für so viele Eventualitäten wie nur möglich vorbereiten. Zum Beispiel: nicht nur die aller nötigsten Elemente stylen sondern auch die, die recht selten verwendet werden, wie z. B. die Definitionsliste.

"Entfernen Sie mein Bild von Ihrer Homepage ... sofort!"

Wie es üblich ist, werden Sie bei einem WordPress-Theme einen Link auf Ihre Website unterbringen wollen. Das ist nur legitim, Sie bieten das Theme kostenlos an und die Leute bzw. potentielle Kunden sollen wissen, dass Sie der Urheber des Werkes sind: das ist auch der Sinn an der ganzen Geschichte.

Nun ist es leider so, dass viele Blogger ihre Kontaktdaten nicht veröffentlichen **und/oder** manche Besucher die Menüpunkte "Kontakt" oder "Impressum" ignorieren. So dass es Ihnen passieren kann, dass Sie z. B. am Telefon aufgefordert werden auf "Ihrer" Website den Text, das Foto, das Video etc. zu entfernen.

In den allermeisten Fällen lässt sich das sehr schnell klären und die Anrufer wissen dann, dass Sie nicht der Betreiber der besagten Website sind, sondern nur das Theme zum kostenlosen Download angeboten haben. Am besten verwenden Sie bei solchen Gesprächen nicht das Wort Theme, sondern "kostenloses Design". Damit können auch Leute die weder Weblogs noch WordPress kennen eher etwas anfangen.

Ich will Sie nicht abschrecken, aber es soll nicht nachher heißen, dass ich diese Info unterschlagen habe.

Anforderungen der Verzeichnisse

Wenn Sie ein WordPress-Theme erstellt haben und zur Verfügung stellen möchten, sollten Sie dies (auch) möglichst auf den offiziellen Seiten von WordPress tun. Die Verzeichnisse stellen dafür allerdings einige Bedingungen.

Da momentan das [deutschsprachige Themes-Verzeichnis](#)²⁸ noch immer einen Aufnahmestopp hat, beziehen sich folgende Anforderungen lediglich auf das [offizielle Themes-Verzeichnis](#)²⁹.

1. Erstelle eine zip-Datei mit allen benötigten Themes-Dateien.
2. Erstelle eine CSS-Datei, die folgendes im einleitenden CSS-Kommentar beinhaltet: einen einmaligen Themes-Namen, Tags, Version und folgenden Klassen: `.alignright`, `.alignleft`, `.aligncenter`.
3. Füge dem Theme einen Screenshot bei (*screenshot.png*).
4. Das Theme muss OpenSource (GPL) sein.
5. Keine versteckten und bezahlten Links im Theme.
6. Die RSS-Feeds bzw. die Links zu den Feeds einbauen.
7. Die [Gravatar](#)³⁰-Funktion einbinden.
8. Die Widget-Funktionalität einbinden.
9. Die Kategorien und Tags müssen ausgegeben werden.
10. Der Blog-Titel (`bloginfo('name');`) und der Slogan (`bloginfo('description');`) müssen angezeigt werden.
11. Du bist der Urheber des Themes.
12. Themes mit 18+ Inhalt (Porno & Gewalt) werden nicht aufgenommen.

Original: <http://wordpress.org/extend/themes/about/>³¹

Die Anforderungen sind recht hoch und die Betreiber sind schon teilweise recht kleinlich bei der Kontrolle der eingehenden Themes. Aber es lohnt sich dennoch das Theme für das offizielle Verzeichnis aufzubereiten, da das offizielle Verzeichnis die erste Anlaufstelle ist, wenn die Leute nach einem kostenlosen Theme suchen.

28) <http://themes.wordpress-deutschland.org/>

29) <http://wordpress.org/extend/themes/>

30) http://codex.wordpress.org/Using_Gravatars

31) <http://wordpress.org/extend/themes/about/>

Über akademie.de

akademie.de bietet [Praxis-Wissen](#)¹ in Form von Artikeln, ausführlichen Ratgebern, verständlichen Anleitungen und praxiserprobten Mustervorlagen.

Bei uns finden Sie Beiträge zu über 2.400 Einzelthemen: von Existenzgründung bis Umsatzsteuervoranmeldung, von Marketing bis Forderungsmanagement, von Soft Skills bis digitale Bildbearbeitung - und noch viel mehr.

An unseren [Online-Workshops](#)² können Sie bequem von jedem Rechner mit Internetzugang aus teilnehmen: von Experten angeleitet, mit persönlichem Feedback, bei freier Zeiteinteilung.

Testen Sie akademie.de zwei Wochen lang [kostenlos](#)³!

Nutzungshinweise und Copyright: Was darf ich mit diesem Text machen?

Sie dürfen unsere Texte, Bilder, Programmcodes und Musterdateien speichern und für Ihren eigenen Gebrauch ausdrucken und nutzen.

Dagegen dürfen Sie unsere Inhalte nicht vervielfältigen, veröffentlichen oder als Unterrichtsmaterial o.ä. nutzen. Verstöße gegen das Urheberrecht können teuer werden. Besser: einfach fragen - per E-Mail: service@akademie.de⁴ oder Telefon: (030) 616 55 - 0.

1) <https://www.akademie.de/>

2) <https://www.akademie.de/online-workshops>

3) <https://www.akademie.de/user/mitglied-werden>

4) <mailto:service@akademie.de>