Clemens Gull

Know-how ist blau.

130 Hacks und Tweaks für WordPress 3!

WordPress-Tricks und -Tweaks

> So helfen Sie sich bei WordPress-Pannen selbst

> Bereichern Sie Ihre Website durch Social-Media-Funktionen für Twitter, Facebook & Co.

> So verbessern Sie die Usability Ihres Internetauftritts

Bohren Sie WordPress mit selbst programmierten Funktionen auf!

FRANZIS

Inhaltsübersicht

Vorwort

Worum geht es in diesem Buch?

Was wird für dieses Buch benötigt?

Gute Voraussetzungen

Herunterladen des Beispielcodes zum Buch

Berichtigungen

Danke!

Feedback

1 Tweaks

1.1 Der Aufbau der Tweaks

1.2 Konventionen

1.3 Diese Fehler können leicht auftreten

2 Einfache Tweaks

2.1 Hilfen für die Konfiguration von WordPress

2.2 Meta-Informationen anpassen und verändern

2.3 Formate anpassen oder erstellen

2.4 Zusätzliche Informationen für den Blog

2.5 ShortCodes verwenden

2.6 Soziale Netzwerke und externe Daten verwenden

2.7 Zusätzliche Funktionen ohne Plug-Ins

2.8 Die Sicherheit von WordPress verbessern

2.9 Das Back-End anpassen oder WordPress erweitern

3 Aufwendige Tweaks

3.1 Formate anpassen oder erstellen

3.2 ShortCodes verwenden

3.3 Zusätzliche Informationen für den Blog

3.4 Zusätzliche Funktionen ohne Plug-Ins

3.5 Das Back-End anpassen oder WordPress erweitern

4 Komplexe Tweaks

4.1 Formate anpassen oder erstellen

4.2 Die Usability verbessern

4.3 Das Back-End oder WordPress erweitern

5 Grundlegende Begriffe

5.1 Der Loop

5.2 Sidebars

5.3 Aufbau eines Themes

5.4 Der Aufbau eines Templates

Stichwortverzeichnis

Vorwort

Worum geht es in diesem Buch?

Dieses Buch ist kein klassisches Lehrbuch, wie wir es dutzendfach kennen. Aber es ist auch keine Referenz oder ein Kompendium zu WordPress. Es ist eine Sammlung von Tweaks[1] oder Hacks zu WordPress.

Diese Sammlung ist über Monate und Jahre in der praktischen Arbeit mit WordPress in den verschiedensten Einsatzgebieten entstanden. Wahrscheinlich sind die meisten Tipps, Tricks, Tweaks und Änderungen auch irgendwo in den Weiten des World Wide Webs zu finden. Aber einerseits ist es sicherlich sehr mühsam, sie für den einen oder anderen Zweck extra zu suchen. Andererseits ist es sehr praktisch, wenn diese Tweaks getestet, auch mit einer zusätzlichen Erklärung versehen und noch um zusätzliche Funktionen erweitert wurden.

Jeder einzelne Tweak ist in sich abgeschlossen und kann in der passenden WordPress-Installation eingesetzt werden. Natürlich sind die einzelnen Tweaks für sich getestet worden, aber trotzdem kann jeder Tweak in einer bestimmten Situation anders reagieren. Denn jedes Blog auf WordPress-Basis hat unterschiedliche Plug-Ins installiert. Leider ist es nicht möglich, alle Plug-In-Kombinationen mit den verschiedenen WordPress-Versionen und den Tweaks

zu testen. Daher ist es sinnvoll, die Tweaks nicht sofort in einer Produktivumgebung – also in einem echten Blog, der live im Internet steht – einzusetzen. Besser ist es, sie vorher in einem Testblog auszuprobieren.

Falls Sie hier eine Aufzählung aller Möglichkeiten, Tags und Befehle mit allen Eigenschaften von WordPress erwarten, muss ich Sie leider enttäuschen. Ich schreibe hier über die Kleinigkeiten, die bei der Arbeit mit WordPress unangenehm sind und – zum Teil mit überraschend geringem Aufwand – verbessert werden könnten.

In diesem Buch erwarten Sie drei Hauptkapitel. Ich habe mir lange überlegt, wie man die Tweaks zusammenfassen kann. Die beste Lösung war für mich, das Augenmerk auf den Schwierigkeitsgrad zu legen.

In den *einfachen Tweaks* sind kurze Programme oder einzeilige Einträge zusammengefasst. Diese Änderungen sind mit wenig Grundwissen schnell und einfach anwendbar.

Im Kapitel *Aufwendige Tweaks* sind Tipps und Tricks gesammelt, die bereits grundlegende Programmierkenntnisse voraussetzen. Sie sollten sich beim Einsatz dieser Tweaks ein wenig mit dem System von WordPress auskennen.

Das letzte Kapitel, *Komplexe Tweaks*, setzt bereits gefestigte Kenntnisse in PHP und/oder WordPress voraus. Natürlich können Sie diese Hacks auch als Nicht-Programmierer verwenden, wenn Sie die Anweisungen buchstabengetreu umsetzen. Aber mit dem notwendigen Wissen fällt es einfach leichter.

In den letzten beiden Abschnitten finden Sie einerseits ein *Glossar*, das verschiedene Fachbegriffe nochmals erklärt. Im *Index* sind natürlich die wichtigsten Begriffe zusammengefasst, aber Sie finden dort auch die einzelnen Tweaks schnell und einfach in alphabetischer Reihenfolge.

Was wird für dieses Buch benötigt?

Um die Tweaks anwenden zu können, setze ich eine funktionierende Installation von WordPress voraus. Ob es sich um ein System im Internet oder eine lokale Installation handelt, ist prinzipiell gleichgültig. Wichtig ist nur, dass Sie die einzelnen Dateien des aktiven Themas kennen und sie auch verändern können.

Welche Entwicklungsumgebung Sie verwenden, ist für die Verwendung dieses Buchs gleichgültig. Ich selbst bevorzuge Eclipse[2] mit einem Plug-In für die Arbeit mit PHP. Falls Sie eine lokale Installation von WordPress erstellen wollen, empfehle ich das Buch *PHP für WordPress* [3]. Dort finden Sie auch einen

guten und einfachen Einstieg in die Anpassung von WordPress-Themen und die Anwendung von PHP mit der Blog-Software.

Gute Voraussetzungen

Wo es für die Anwendung des Tweaks notwendig ist, werde ich natürlich ein wenig auf die Hintergründe von WordPress (die Tags, Befehle, Datenbanktabellen oder -felder, usw.) eingehen. Selbstverständlich werde ich auch, falls es notwendig ist, Elemente von HTML, Eigenschaften von CSS oder auch JavaScript-Befehle erklären.

Grundsätzlich sollten Sie HTML, CSS und JavaScript bereits sicher beherrschen. Wenn Sie sich in CSS noch nicht gut auskennen, ist *Webseiten-Layout mit CSS* [4] ein guter Einstieg. Für den Anfang in HTML, aber auch als Nachschlagewerk kann ich außerdem das *HTML5-Handbuch* [5] empfehlen. Außerdem schadet es nicht, wenn Sie bereits programmieren können. Hier wären Kenntnisse der Programmiersprache PHP sehr vorteilhaft. Falls Sie sich hier noch nicht so sicher sind, lassen sich einige Tweaks sicher schwer nachvollziehen. Für die Arbeit mit PHP und WordPress können Sie das oben empfohlene Buch *PHP für WordPress* durcharbeiten. Falls Sie noch etwas tiefer in die PHP-Programmierung einsteigen wollen, ist auch *Das Franzis-Lernpaket PHP/MySQL* [6] empfehlenswert.

Vorteilhaft, aber nicht notwendig ist das verwendete Template für WordPress. Ich habe mich für das Template »prettyNew« entschieden, das im Buch *PHP für WordPress* detailliert beschrieben wird. Natürlich können Sie alle darin beschriebenen CSS-Formatierungen und dargestellten HTML-Elemente auf Ihr persönliches WordPress-Thema anpassen.

Herunterladen des Beispielcodes zum Buch

Besuchen Sie unsere Website unter *http://www.buch.cd* und geben Sie dort die letzten sechs Ziffern der ISBN dieses Buches samt Bindestrich ein, um alle Beispielcodes und sonstigen Ressourcen zu diesem Buch herunterzuladen. Die verfügbaren Dateien werden nach der erfolgreichen Anmeldung angezeigt.

Berichtigungen

Obwohl alle Beteiligten mit größter Sorgfalt vorgehen, um die Richtigkeit der Inhalte sicherzustellen, passieren Fehler. Wenn Sie einen Fehler in diesem Buch entdecken, egal ob im Text oder im Quellcode, bin ich sehr dankbar, wenn ich eine Mitteilung erhalte. So können Sie anderen Lesern Ärger ersparen und mithelfen, die nachfolgende Version des Buches zu verbessern. Wenn Sie irgendeinen Druckfehler finden, teilen Sie ihn mir bitte per eMail an *buch@guru-20.info* mit. Ich werde alle Berichtigungen, Änderungen und Verbesserungen auf meinem Blog *http://www.guru-20.info* veröffentlichen.

Danke!

Herrn Franz Graser, meinem Lektor und Betreuer beim Franzis Verlag: Danke für die kompetente Betreuung bei der Umsetzung der Bücher, die Korrekturen und die Möglichkeit, als Autor zu arbeiten.

Dank auch an das komplette Team des Franzis Verlags. Ohne die vielen im Hintergrund arbeitenden helfenden Hände wäre die Erscheinung dieses Buches nicht gelungen.

Feedback

Ich würde mich über Reaktionen und Anregungen sehr freuen. Sie erreichen mich unter folgender Adresse: *gull@guru-20.info*

Ihr

Clemens Gull

Widmung

Für Raphaël

Ein (un)heimlicher IT-Berater

[1]
Ein Tweak verbessert ein komplexes System. Tweaks sind all die kleinen Änderungen, die ein System

verbessern, komfortabler oder fehlerärmer machen.

[2]
Im Internet *http://www.eclipse.org/downloads/packages/release/helios/sr1* unter der freien Lizenz Eclipse Public License verfügbar.

[3]
PHP für WordPress, Franzis Verlag, ISBN: 978-3-645-60011-8

[4]
Webseiten-Layout mit CSS, Franzis Verlag, ISBN: 978-3-7723-7568-2

[5]
HTML 5 Handbuch, Franzis Verlag, ISBN: 978-3-645-60079-8

[6]
Das Franzis-Lernpaket PHP/MySQL, Franzis Verlag, ISBN: 978-3-645-70047-4

1 Tweaks

Was sind eigentlich diese berühmten Tweaks oder oft als Hacks bezeichneten Dinger? Die Frage ist schnell beantwortet: Es sind mehr oder weniger kurze Programmschnipsel, die, in unserem Fall, mehr aus WordPress herausholen.

Oft ist es nur ein kurzer Programmcode, der eine versteckte Funktion aktiviert oder einfach nur mehr Komfort für die tägliche Arbeit bietet. Es kann aber auch ein umfangreicheres Codesegment sein, um zusätzliche Funktionen zu erhalten.

Aber ein Grundprinzip haben alle Tweaks gemeinsam: Sie greifen nie direkt in das System ein. Damit ist gemeint, dass die Core-Dateien [1] nicht verändert werden. In diesem Buch werden daher nur Eingriffe in die Konfigurationsdatei und in Dateien des aktiven Themas vorgenommen. Wenn Sie sich nicht sicher sind, wie sich WordPress wirklich zusammensetzt, werfen Sie einfach einen Blick in das Glossar am Ende des Buches.

1.1 Der Aufbau der Tweaks

Wie bereits im Vorwort beschrieben, ist das Buch in drei Hauptkapitel gegliedert, um die Tweaks je nach Schwierigkeit einzuteilen. Jedes Kapitel enthält zusammenfassende Themen. Allen Tweaks ist eine gewisse Struktur gemeinsam.

Änderungen an der Konfigurationsdatei

Soll die Konfigurationsdatei von WordPress, *wp-config.php* , verändert werden, können Sie den Eintrag direkt an der angegebenen Stelle vornehmen. Bevor Sie jedoch Änderungen an einem laufenden Blog, einer aktiven Website, vornehmen, sollten Sie die Konfigurationsdatei sichern. Falls etwas schiefgeht, können Sie dann auf die unveränderten Konfigurationsdaten zurückgreifen.

Den passenden Ort für jeden Tweak in der Datei *wp-config.php* finden Sie schnell. Am besten suchen Sie folgenden Quellcode:

```
if ( !defined('ABSPATH') )
define('ABSPATH', dirname(__FILE__) . '/');
require_once(ABSPATH . 'wp-settings.php');
```

Genau davor steht eine Bemerkungszeile. Sie erkennen sie durch `/*` am Beginn und `*/` am Ende.

Sie sieht ungefähr (je nach WordPress-Version) so aus:

```
/* That's all, stop editing! Happy blogging. */
```

Genau vor dieser Zeile fügen Sie den jeweiligen Quellcode ein. Damit kann eine angepasste Konfigurationsdatei so aussehen, wobei die geänderten Zeilen fett geschrieben sind:

```
 * in their development environments.
*/
define('WP_DEBUG', false);
/**************************

 * TWEAKS *

 **************************/

/***********************************

 * Funktion:  Mehrere URLs/Domaenen fuer

 * diesen Blog verwenden

 * WordPress: alle

 * Wirkung:   Front-End/Back-End

 * Aufruf:    Parameter-Definition

 ***********************************/

 define('WP_SITEURL',
'http://'.$_SERVER['SERVER_NAME']);
```

```
 define('WP_HOME',
'http://'.$_SERVER['SERVER_NAME']);

/* That's all, stop editing! Happy blogging. */
/** Absolute path to the WordPress directory. */
if (!defined('ABSPATH')) {
define('ABSPATH', dirname(__FILE__) . '/');
}
/** Sets up WordPress vars and included files. */
require_once(ABSPATH . 'wp-settings.php');
```

Änderungen an den Stilen des WordPress-Themas

Alle Formatierungen des aktiven Themas sind normalerweise in der Datei *style.css* abgelegt. Sie könnten alle Stile an das Ende der Datei anfügen. Aber um die Tweaks klarer zu trennen, habe ich mich für einen anderen Weg entschieden.

Wir legen für jeden Tweak, natürlich nur, wo es notwendig ist, eine neue CSS-Datei innerhalb des aktiven Themas an. Der Speicherort ist normalerweise das Basisverzeichnis des Templates. Aber wenn Sie sich nicht sicher sind, suchen Sie die Datei *style.css* im aktiven Thema, und Sie haben den Speicherort gefunden. Den Namen für das neue Stylesheet finden Sie immer am Anfang des Tweaks in einer eigenen Rubrik.

Danach müssen Sie nur noch den neuen Stil in die Header-Datei des Themas einbinden. Dazu öffnen Sie die Datei *header.php* und suchen nach folgender Zeile:

```
<link rel="stylesheet" type="text/css"
media="all" href="<?php bloginfo(
'stylesheet_url' ); ?>" />
```

Sie wird je nach Thema vielleicht ein wenig anders aussehen. Aber auf jeden Fall finden Sie den Begriff `bloginfo( 'stylesheet_url' );` in der Zeile. Nun haben Sie den richtigen Ort für die Änderung: Genau nach dieser Zeile binden Sie den neuen Stil mit folgendem Befehl ein:

```
<link rel="stylesheet" href="<?php
bloginfo('stylesheet_directory'); ?
>/microformat.css" type="text/css" media="screen"
/>
```

Auch wenn der Befehl hier in mehreren Zeilen erscheint, müssen Sie ihn in einer Zeile schreiben. Auch müssen Sie den Dateinamen, hier `microformat`, dem jeweiligen Tweak anpassen.

Änderungen an den Funktionen von WordPress

Prinzipiell gibt es für diese Anpassungen eine eigene Datei mit dem Namen *functions.php* im aktiven Thema. Falls sie bei Ihnen nicht vorhanden sein sollte, können Sie die Datei einfach anlegen.

In dieser Datei werden alle benutzerdefinierten Änderungen gespeichert. Sie wird bei einem Update von WordPress nicht angepasst, sondern hängt alleine vom jeweiligen Thema ab. Dadurch können Sie hier alle Tweaks einbauen und nach einem Update weiterverwenden, ohne von vorne beginnen zu müssen.

Selbstverständlich sollten Sie vor oder nach einem Update von WordPress die Tweaks überprüfen. Denn sie könnten für die neue Version nicht mehr verfügbar sein.

Auch bei den Funktionen habe ich mich für einen eigenen Weg entschieden, um die Struktur klarer zu gestalten. Nur einzeilige Tweaks programmieren wir direkt in dieser Datei. Hier fügen Sie die Zeile wie weiter unten beschrieben ein.

Die längeren Tweaks bauen wir nicht direkt in die Datei *functions.php* ein, sondern über sogenannte Includes. Das sind externe Dateien, die mit einem speziellen Befehl in eine andere PHP-Datei aufgenommen werden. Dazu gehen wir wie folgt vor:

Im aktiven Theme legen Sie ein Verzeichnis mit dem Namen *includes* an. In diesem werden wir später die einzelnen Tweaks in je einer PHP-Datei speichern. Nehmen wir an, wir haben einen Tweak erstellt und ihn in einer Datei mit dem Namen *boostMimeTypes.php* im Verzeichnis *includes* abgelegt. Nun müssen wir diesen Tweak auch in der Datei *functions.php* zur Verfügung stellen.

In der Datei *functions.php* finden Sie am Anfang ein paar Bemerkungszeilen , die wie folgt aussehen können:

```
<?php
/*********************
* Funktion:  Funktionen fuer das Thema prettyNew
* Dateiname: functions.php
* Autor:     Clemens Gull
* Version:   1.4
* Erstellt:  25. Juli 2008, 12:51
* Aenderung: 27. Juli 2008, Version 1.1
*                Hinzufuegen der 2. Seitenleiste
*            28. Juli 2008, Version 1.2
*                Internationalisierung
hinzugefuegt
*********************/
```

Danach kann noch zusätzlicher Quellcode kommen, der aber (noch) nicht von Interesse ist. Unsere Tweaks fügen wir entweder genau nach dem Ende des Bemerkungskopfes, also hinter der letzten Zeile im obigen Quellcode, ein. Falls keine Bemerkungen

vorhanden sind, fügen wir den Tweak direkt nach der öffnenden Zeile mit der Sequenz `<?php` ein.

Für unsere neue Funktion würde die geänderte Datei *functions.php* so aussehen, damit der Tweak aktiv ist:

```
<?php
/********************
* Funktion:  Funktionen fuer das Thema prettyNew
* Dateiname: functions.php
* Autor:     Clemens Gull
* Version:   1.4
* Erstellt:  25. Juli 2008, 12:51
* Aenderung: 27. Juli 2008, Version 1.1
*                Hinzufuegen der 2. Seitenleiste
*            28. Juli 2008, Version 1.2
*                Internationalisierung
hinzugefuegt
********************/
//MIME-Types hinzufuegen
require_once 'includes/boostMimeTypes.php';
...
?>
```

Auch diese Dateinamen sind am Anfang jedes Tweaks angeführt.

Änderungen an den Dateien des Themes

Bei manchen Tweaks muss eine bestimmte Datei des Themes verändert werden. Wenn sicher ist, welche das sein soll, finden Sie den exakten Dateinamen bei der Beschreibung des betreffenden Tweaks.

Es gibt aber auch Fälle, in denen es entweder mehrere verschiedene mögliche Dateien gibt oder Ihnen die Entscheidung überlassen wird. Der zweite Punkt ist leichter zu beantworten. In diesem Fall können Sie den Tweak entweder in die Seitenleiste oder in der Hauptseite oder auch bei einer Einzelseite einbauen. Das gilt zum Beispiel, wenn Sie Informationen zum Autor eines Artikels anzeigen lassen wollen. Dann kann das nach jedem Artikel auf der Hauptseite sein oder auch nur am Ende eines Artikels, falls er in der Einzelansicht angezeigt wird. Hier finden Sie zu Anfang der Beschreibung des Tweaks einen Vorschlag.

Ein typisches Beispiel für den ersten Punkt ist der Loop . Wenn dieser verändert werden oder auch zusätzliche Informationen anzeigen soll, gibt es mehrere Dateien, die angepasst werden können bzw. müssen. Hier ist prinzipiell Ihnen die Entscheidung darüber überlassen, ob Sie die Hauptseite, das Archiv, die Einzelansicht oder alle Dateien anpassen möchten. Wichtig ist nur, dass die Änderung innerhalb des Loops passiert. Auch hier finden Sie am Anfang der Beschreibung eines Tweaks die notwendige Information.

Für alle Tweaks

Für alle Tweaks gilt, dass diese Vorgehensweisen nicht jedes Mal aufs Neue beschrieben werden. Sie finden vielmehr am Anfang jedes Tweaks die jeweiligen Dateinamen, die Sie verwenden sollen, und auch Informationen, welche Dateien Sie ändern müssen.

Sie haben bereits gesehen, dass jeder Tweak oder jede Änderung einen sehr ausführlichen Bemerkungskopf hat. Natürlich können Sie sehr puristisch programmieren und nur die Befehle übernehmen. Trotzdem empfehle ich Ihnen diese Form. Denn so lässt sich einerseits auch nach Wochen und Monaten noch erkennen, was der Tweak wirklich macht. Und wie ich oben beschrieben habe, können WordPress-Updates die Tweaks stark beeinflussen. Daher ist es gut, gleich die Version Ihrer WordPress-Installation zu kennen.

Bei jedem Tweak finden Sie innerhalb des Sourcecodes immer wieder Bemerkungen. Diese dienen dem besseren Verständnis, wie die einzelnen Zeilen und Befehle funktionieren. Diese können Sie ebenfalls weglassen, wenn Sie sich gut mit der PHP-Programmierung auskennen. Aber falls Sie noch kein geübter Programmierer sind, würde ich sie ebenfalls stehen lassen. Denn so können Sie jederzeit

nachvollziehen, wie etwas funktioniert und wo Sie gefahrlos etwas verändern können.

Leider schaffe ich es hier nicht, zwei wichtige Dinge gleichzeitig zu berücksichtigen: einerseits den Komfort, dass die Tweaks für alle Benutzer funktionieren, und andererseits ein geschwindigkeitsoptimiertes WordPress. Ich habe mich als Schwerpunkt für das Erste entschieden. Die Tweaks sollen also möglichst gut und komfortabel für alle Leserinnen und Leser anwendbar sein.

Wenn Sie aber Geschwindigkeit für Ihren Blog als wichtiges Ziel definiert haben, müssen Sie bei allen Befehlen, die Informationen zu den Grundeinstellungen des Blogs abrufen, vorsichtig sein. Am einfachsten ist es, wenn Sie Befehle, die ein Verzeichnis oder den Namen des Blogs oder die eMail-Adresse des Administrators abfragen, durch feste Werte ersetzen. Beispielsweise ersetzen Sie den Befehl

```
  //eMail-Adresse des Administrators
$adminEmail = get_option('admin_email');
```

durch folgende Version:

```
  //eMail-Adresse des Administrators
$adminEmail = 'buch@guru-20.info';
```

Dies ist zwar nicht sehr elegant, aber es funktioniert. Es besteht aber auch eine Gefahr dabei. Falls sich

diese Daten einmal ändern, etwa, wenn Sie eine neue Mailadresse benutzen, kann es leicht passieren, dass Sie vergessen, dies in Ihrem Code zu ändern.

1.2 Konventionen

Die einzelnen unterschiedlichen Formatierungen bedürfen sicher keiner genaueren Erklärung, sie erschließen sich Ihnen beim Lesen. Die verwendeten Symbole spreche ich aber trotzdem kurz an.

Symbol	Verwendung
	Der Tweak kann für die WordPress-Version 2.7 verwendet werden.
	Der Tweak kann für die WordPress-Version 2.8 verwendet werden.
	Der Tweak kann für die WordPress-Version 2.9 verwendet werden.
	Der Tweak kann für die WordPress-Version 3.0 verwendet werden.

1.3 Diese Fehler können leicht auftreten

Fehlerhafte Header-Information

Einer der häufigsten Fehler ist die Meldung `Cannot modify header information`, wie sie in der folgenden Abbildung gezeigt ist.

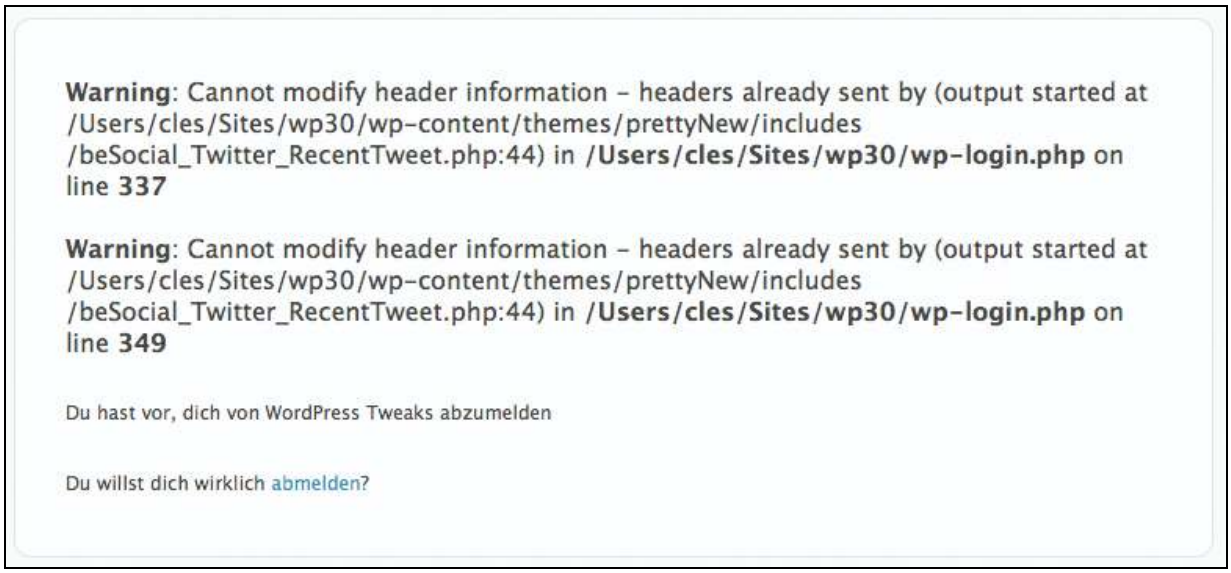

Bild 1.1 Dieser Fehler tritt häufig auf – seine Ursache ist aber eher banal.

Dieser Fehler hat große Auswirkungen, aber eine banale Ursache. Das Dumme daran ist, dass Sie die Ursache kaum sehen. Denn sie besteht meistens aus Leerzeichen, die am Anfang bzw. am Ende einer PHP-Datei stehen.

Um den Fehler zu beheben, müssen Sie zuerst die betroffene PHP-Datei identifizieren. Den Namen finden Sie in der Fehlermeldung; er steht in der runden Klammer. Man könnte eigentlich meinen, die fett angezeigte Datei enthalte den Fehler. Dem ist aber nicht so, diese Datei löst den Fehler nur aus.

Wenn Sie die betroffene Datei untersuchen, im oben gezeigten Fall ist es `beSocial_Twitter_RecentTweet.php`, überprüfen Sie, ob vor dem ersten `<?php` und nach dem letzten `?>` noch Leerzeichen oder irgendein anderer Text steht. Diese Zeichen müssen Sie löschen, da sie den Fehler erzeugen.

Fehlende Funktion

Oft kann es vorkommen, dass eine Fehlermeldung `Call to undefined function` erscheint, die Sie im folgenden Bild sehen.

Bild 1.2 Ein weiterer häufig auftretender Fehler – eine nicht definierte Funktion wird aufgerufen. Oft ist ein Tippfehler die Ursache.

Sie finden dort auch immer den Namen der Funktion und auch den Dateinamen, der den Fehler

verursacht. Dieser Fehler kann verschiedene Ursachen haben.

Es kann oft ein einfacher Tippfehler sein. Vielleicht haben Sie sich beim Aufruf der Funktion vertippt. Dann müssen Sie nur diesen Fehler korrigieren.

Der Fehler kann seine Ursache auch im Erstellen der Funktion haben. Nach dem Befehl `function` geben Sie ja den Namen der Funktion an. Hier können Sie sich auch vertippt haben. Oder Sie finden dort den wirklichen Namen der Funktion, so wie Sie ihn an anderer Stelle eingeben müssen, um sie zu verwenden.

Die letzte Ursache kann ein ganz einfaches Versehen sein. Vielleicht haben Sie die Funktion einfach nicht in der Datei *functions.php* eingebunden. Meistens fehlt der Aufruf der Datei mit dem Befehl `require_once 'dateiname.php'`.

Syntaxfehler

Dies ist einer der häufigsten Fehler, und auch hier gibt es einen Fehler, der sehr oft gemacht wird. Er zeigt sich in der unten dargestellten Fehlermeldung.

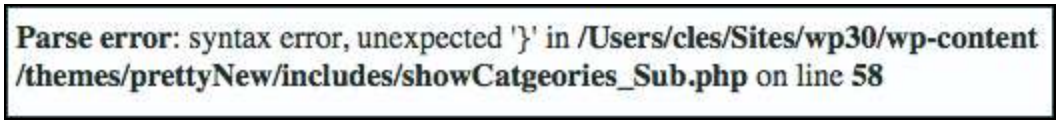

Bild 1.3 Ein syntax error – von allen Programmierern gleichermaßen gefürchtet

Sie sehen sofort am Anfang der Meldung `syntax error, unexpected`, dass es sich um einen Fehler in der Syntax von PHP handelt. Am Ende sind die Zeilen angegeben, in denen Sie den Fehler finden. Dies ist unter Umständen verwirrend! Denn meistens haben Sie in der vorhergehenden Zeile das Semikolon vergessen. So wie Sie es hier in Zeile 57 sehen. PHP ist hier sehr sensibel. Jede Zeile, außer die mit Blockoperatoren, muss mit einem Strichpunkt beendet werden.

```
...
55. } else {
56.     //Rueckgabe der Liste
57.     return $catList
58.   }
59. }
...
```

Vergessene Klammern

Oft vergisst man, eine geschwungene Klammer bzw. den Blockoperator von PHP zu schließen. Ich habe mir den Trick angewöhnt, den Blockoperator sofort wieder zu schließen, wenn ich ihn geöffnet habe, und danach zwischen den beiden weiterzuschreiben. Trotzdem kommt es auch bei mir manchmal zu folgender Fehlermeldung:

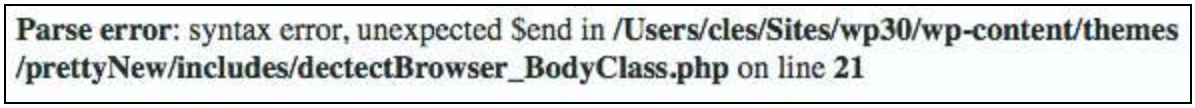

Bild 1.4 Diese Fehlermeldung beschwert sich über ein unerwartetes Zeilenende – meist ist eine vergessene Klammer der Grund.

PHP beschwert sich zwar auch hier über einen Syntax-Fehler, aber wenn Sie weiterlesen, sehen Sie die Formulierung `unexpexted $end` im Text. Die Zeilennummer bei so einer Fehlermeldung können Sie normalerweise ignorieren, da es immer die letzte Zeile des Codes ist. Denn PHP erkennt nicht, dass Sie die Klammer vergessen haben.

In diesem Fall müssen Sie wohl oder übel den Code durchsuchen und alle Klammern überprüfen.

Vergessene Punkte

Auch dieser Fehler passiert häufig! Punkte werden zum Verketten von Zeichenketten benutzt. Meistens werden damit Texte und Anführungszeichen und Variablen verbunden. Fehlt hier ein Punkt oder ist er zwischen die Anführungszeichen gerutscht, kommt es zu folgender Fehlermeldung. Sie sagt tröstlicherweise am Ende aber gleich, welche Zeile betroffen ist.

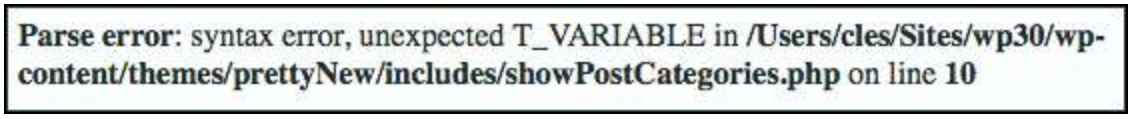

Bild 1.5 Diese Fehlermeldung wurde durch einen vergessenen Punkt herbeigeführt.

Im Quellcode kann das so aussehen:

```
...
$outPut .= '<a href="/category/'$cName.'
title="'.
$cName.'"><img src="'.
get_bloginfo('stylesheet_directory').
'/images/cat-'.$cID.'.jpg" alt="'.$cName.'" '.
' border="0" width="64" height="64" /></a>';
...
```

Sie sehen sofort, dass in der ersten Zeile bei der Variablen `$cName` ein Punkt vor dem Dollarzeichen fehlt. Es kann aber auch so aussehen: