

Wie erstellst Du eigene Shortcodes in WordPress?

Wie erstellst Du eigene Shortcodes in WordPress?

Der Kern eines Shortcodes ist der PHP-Code, der ausgeführt werden soll, wenn Wordpress in einem Blogbeitrag darauf stößt. In diesem Workshop gehen wir nicht auf die Programmierung der jeweiligen Shortcode-Funktionalität ein, denn das würde zu weit führen und wäre letztlich ein Programmierkurs in PHP. Aber wir zeigen, wie Du einen eigenen Shortcode in Wordpress zum Laufen bekommst. Wir beginnen mit einem einfachen Beispiel und bauen im weiteren Verlauf dann darauf auf, um komplexere Einsatzmöglichkeiten von Shortcodes zu demonstrieren.

Alle PHP-Code-Schnipsel aus diesem Workshop gehören in die *functions.php* Deines Wordpress-Themes beziehungsweise sinnvollerweise in die *functions.php* eines Child Themes – oder noch besser in ein individuelles Plugin. Der Einfachheit halber sprechen wir im Workshop dennoch von der *functions.php*.

Inhalt:

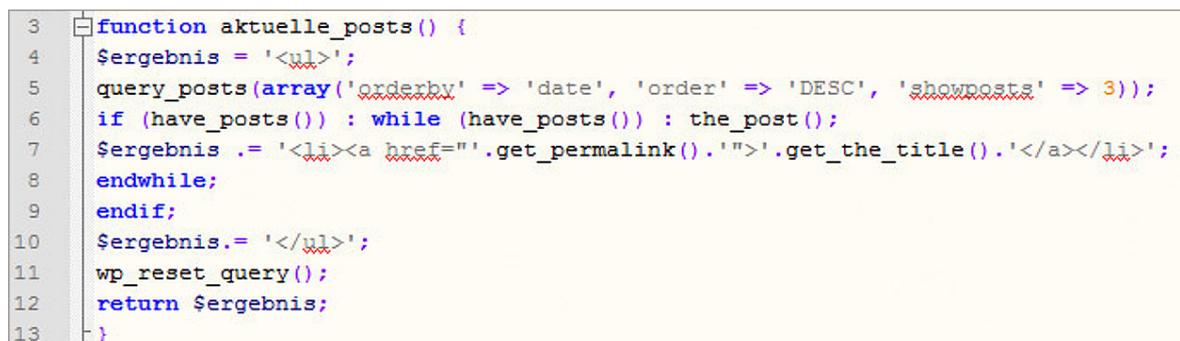
| | |
|--|-----------|
| 1. Die Callback-Funktion | 3 |
| 2. Quellen für fertige Shortcodes | 4 |
| 3. Shortcode-Funktion in Wordpress registrieren | 4 |
| 4. Flexiblere Inhalte mit Parametern | 5 |
| 5. Shortcodes auf ganze Textpassagen anwenden | 7 |
| 6. Shortcodes in der Sidebar und im Beitrags-Teaser verwenden | 8 |
| 7. Shortcodes im PHP-Code eines Themes verwenden | 9 |
| 8. Was passiert mit alten, nicht mehr benötigten Shortcodes? | 10 |
| 9. Nützliche und weniger nützliche Shortcode-Plugins | 11 |

1. Die Callback-Funktion

Du beginnst mit der Programmierung der eigentlichen PHP-Funktion, die ausgeführt wird, wenn Wordpress in einem Beitrag den zugehörigen Shortcode findet – die sogenannte Call-back-Funktion.

Für unser Beispiel soll ein Shortcode den Titel der aktuellsten drei Beiträge Deines Blogs inklusive Link anzeigen:

```
function aktuelle_posts() {
    $ergebnis = '<ul>';
    query_posts(array('orderby' => 'date', 'order' => 'DESC', 'showposts' => 3));
    if (have_posts()) : while (have_posts()) : the_post();
    $ergebnis.='<li><a href="'.get_permalink().'">'.get_the_title().'</a></li>';
    endwhile;
    endif;
    $ergebnis.= '</ul>';
    wp_reset_query();
    return $ergebnis;
}
```



```
3 function aktuelle_posts() {
4     $ergebnis = '<ul>';
5     query_posts(array('orderby' => 'date', 'order' => 'DESC', 'showposts' => 3));
6     if (have_posts()) : while (have_posts()) : the_post();
7     $ergebnis .= '<li><a href="'.get_permalink().'">'.get_the_title().'</a></li>';
8     endwhile;
9     endif;
10    $ergebnis.= '</ul>';
11    wp_reset_query();
12    return $ergebnis;
13 }
```

Die Variable *\$ergebnis* enthält nun den HTML-Code, den Wordpress dazu benutzt, den Shortcode im Beitrag zu ersetzen:

```
<ul>
  <li><a href="http://www...de/titel1.html">Titel 01</a></li>
  <li><a href="http://www...de/titel2.html">Titel 02</a></li>
  <li><a href="http://www...de/titel3.html">Titel 03</a></li>
</ul>
```

Wichtig ist bei Shortcodes immer, dass die Variable mit dem Inhalt, der in Wordpress ausgegeben werden soll, per *return* zurückgegeben wird und nicht etwa per *echo*-Befehl.

Nun steht also unsere Funktion, die den auszugebenden Inhalt erzeugt.

2. Quellen für fertige Shortcodes

Statt die Callback-Funktion für Shortcodes selbst zu programmieren, kannst Du natürlich auch fertigen Code nehmen, den Du im Internet findest – beispielsweise bei [WPdev Snippets](#) oder [Snipplr](#).

Bei Code-Schnipseln aus dem Internet solltest Du immer auf eine seriöse Quelle achten, denn schlechter PHP-Code kann die Sicherheit Deines Blogs gefährden. Lies immer alle Kommentare zu einem Code-Schnipsel bis zum Ende, denn dort findest Du meist einen Hinweis darauf und oft auch entsprechende Verbesserungsvorschläge.

3. Shortcode-Funktion in Wordpress registrieren

Jetzt fehlt noch die Integration, damit Wordpress diese Funktion als relevant für die Verarbeitung von Inhalten erkennt. Dazu musst Du die eben erstellte Funktion *aktuelle_posts()* in Wordpress registrieren. Dies geschieht mit einem Codeschnipsel ebenfalls in der *functions.php*:

```
add_shortcode("recent-posts", "aktuelle_posts");
```

```
15 | add_shortcode("recent-posts", "aktuelle_posts");
```

Callback-Funktion in Wordpress für den Shortcode *[recent-posts]* registrieren.

Mit *add_shortcode* definierst Du also einen neuen Shortcode, weist ihm mit dem ersten Parameter (hier: *recent-posts*) seinen Namen zu. Mit dem zweiten Parameter gibst Du an, welche PHP-Funktion ausgeführt werden soll, wenn Wordpress auf diesen Shortcode stößt, hier also *aktuelle_posts()*.

>>

Wenn Wordpress nun in einem Beitrag den Shortcode `[recent-posts]` findet, führt es automatisch die Funktion `aktuelle_posts()` aus und ersetzt den Shortcode durch die von der Funktion erzeugten Inhalte.

Beispiel-Beitrag für Shortcodes

📅 Januar 13, 2016 👤 fneumeier 💬 0 Comment ✎ Edit

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis.

At vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet.

- Lorem ipsum dolor sit amet
- At vero eos et accusam
- Beispiel-Beitrag für Shortcodes

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum.

Das sieht der Leser Deines Blogs an Stelle des Shortcodes `[recent-posts]`.

Wichtig ist dabei, dass Dein Shortcode einen einzigartigen Namen bekommt, damit verschiedene Shortcodes sich nicht gegenseitig stören. Die in unserem Beispiel verwendete Bezeichnung „recent-posts“ ist daher eher problematisch, weil eine gewisse Wahrscheinlichkeit besteht, dass ein anderer Shortcode diese recht naheliegende Bezeichnung bereits verwendet. Für Deine eigenen Shortcodes wählst Du daher am besten ein persönliches Präfix, das Du vor den Shortcode-Namen setzt, etwa Deine Initialen – in meinem Fall nach dem Muster `fn-shortcode`.

4. Flexiblere Inhalte mit Parametern

Im Beispiel oben haben wir mit einem statischen Shortcode begonnen, der immer nur die jeweils drei aktuellsten Blogbeiträge ausgibt. Das bauen wir nun etwas aus und fügen einen Parameter hinzu. Der erlaubt es, im Shortcode anzugeben, wie viele aktuelle Beiträge angezeigt werden sollen, beispielsweise also fünf:

```
[recent-posts posts="5"]
```

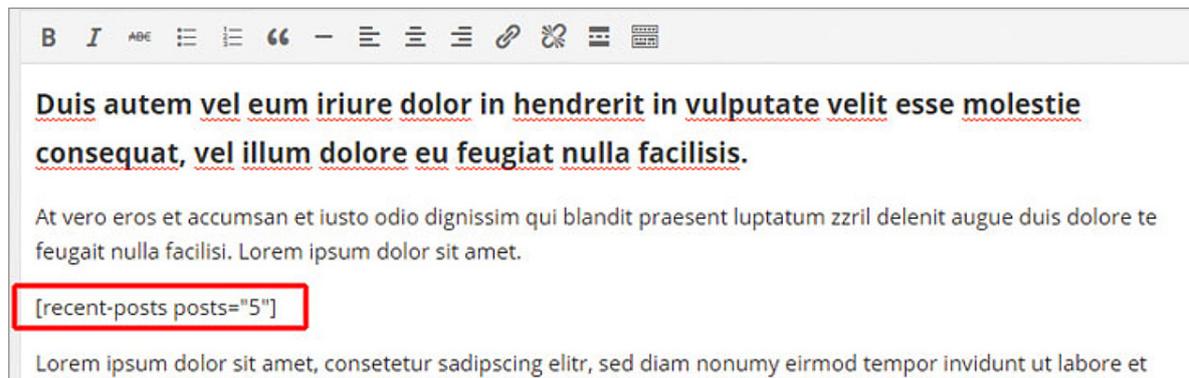
Im PHP-Code unserer Funktion lesen wir den oder die übergebenen Parameter aus und packen sie zur Weiterverarbeitung in ein Array. Wir definieren zusätzlich den Standardwert 3 für den Fall, dass ein Shortcode den Parameter nicht enthält:

```
extract(shortcode_atts(array('posts' => 3), $atts));
```

>>

Damit Wordpress die Liste der aktuellsten Beiträge als Liste darstellt, fügen wir noch den nötigen Code dazu ein. Die erweiterte Callback-Funktion aus unserem Beispiel sieht jetzt so aus (Veränderungen zum vorigen Code in Rot):

```
function aktuelle_posts($atts) {  
    extract(shortcode_atts(array('posts'=> 3), $atts));  
    $ergebnis = '<ul>';  
    query_posts(array('orderby'=>'date', 'order'=>'DESC', 'showposts'=> $posts));  
    if (have_posts()) : while (have_posts()) : the_post();  
    $ergebnis.='<li><a href="'.get_permalink().'">'.get_the_title().'</a></li>';  
    endwhile;  
    endif;  
    $ergebnis.='</ul>';  
    wp_reset_query();  
    return $ergebnis;  
}
```



B I ABC ☰ ☷ “ — ☰ ☷ ☷ 🔗 🌀 ☰ ☷

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis.

At vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue dui dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet.

[recent-posts posts="5"]

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et

Über den Parameter im Shortcode steuerst Du, ...



Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis.

At vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue dui dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet.

- Lorem ipsum dolor sit amet
- At vero eos et accusam
- Beispiel-Beitrag für Shortcodes
- Lorem ipsum dolor sit amet
- Hello world!

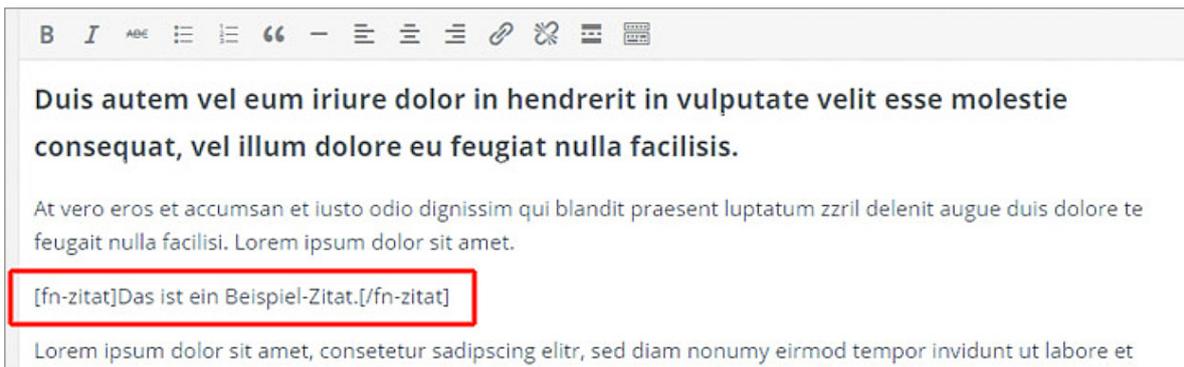
Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore

... wie viele Beiträge dem Leser an dieser Stelle angezeigt werden.

5. Shortcodes auf ganze Textpassagen anwenden

Aber was tun, wenn Du eine ganze Textpassage aus Deinem Blogbeitrag mit der PHP-Funktion verarbeiten willst? Ein Beispiel wäre, ein Zitat optisch hervorzuheben. Dafür gibt es die Shortcode-Variante mit Anfangs- und End-Tag, die wie auch bei dem einfachen Shortcode mit Parametern ergänzbar ist:

```
[fn-zitat]Das ist ein Beispiel-Zitat.[/fn-zitat]
```



So sieht der zweiteilige Shortcode im Editor aus.

Dieser Shortcode besteht also aus einem öffnenden `[fn-zitat]` und einem schließenden Tag `[/fn-zitat]` und bezieht sich auf das, was dazwischen steht. In der Callback-Funktion übernimmst Du den Inhalt zwischen den Tags als Parameter in die Funktion. Im Beispiel nennen wir die Variable `$text`, die wir zusätzlich zu eventuell vorhandenen Parametern übertragen:

```
function zitat($atts, $text = null) {...}
```

Im PHP-Code kannst Du dann die Inhalte der Variable `$text` entsprechend verarbeiten. Fügen wir in unserem Beispiel also einfach ein wenig Formatierung für das Zitat hinzu. Und weil wir damit einen neuen Shortcode definieren, registrieren wir ihn auch gleich noch bei Wordpress:

```
function zitat($atts, $text = null) {  
    $ergebnis = '<div style="font-size:2em; color:red; float:right;">&quot;' .  
        . $text . '&quot;</div>';  
    return $ergebnis;  
}  
add_shortcode("fn-zitat", "zitat");
```

>>

Beispiel-Beitrag für Shortcodes

📅 Januar 13, 2016 👤 fneumeier 💬 0 Comment ✎ Edit

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis.

At vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue dui dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet.

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam

"Das ist ein Beispiel-Zitat."

So sieht der zweiteilige Shortcode im Editor aus.

6. Shortcodes in der Sidebar und im Beitrags-Teaser verwenden

Es gibt Fälle, in denen Du Shortcodes nicht nur innerhalb eines Beitrags verwenden willst, sondern auch in Sidebar-Widgets. Standardmäßig funktioniert das nicht, aber eine Code-Zeile in der `functions.php` aktiviert Shortcodes auch für Widgets:

```
add_filter('widget_text', 'do_shortcode');
```

In den meisten Fällen ist es gut, wenn Shortcodes im Teaser der Beiträge nicht ausgeführt werden – also die ersten paar Zeilen des Beitrags, die auf der Startseite Deines Blogs angezeigt werden. Oft steht am Anfang eines Beitrags nämlich ein Bild inklusive `[caption]`-Shorttag und der würde im Teaser nur stören.

Sollen Shortcodes aber auch im Teaser – im Wordpress-Sprachgebrauch „excerpt“ – ausgeführt werden, aktivierst Du das Feature mit der folgenden Zeile in der `functions.php`:

```
add_filter('the_excerpt', 'do_shortcode');
```

>>

Wordpress Bloggertricks

SAMPLE PAGE SOCIAL STREAM TEST

Uncategorized

Beispiel-Beitrag für Shortcodes

Januar 13, 2016 fneumeier 0 Comment Edit

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla

Read more

Search



Shortcode-Test

"Das Beispiel-Zitat, diesmal in einem Widget."

Letzte Beiträge

Eine Codezeile genügt, um Shortcodes auch in Widgets in der Randspalte verwenden zu können.

7. Shortcodes im PHP-Code eines Themes verwenden

Shortcodes kannst Du übrigens nicht nur beim Schreiben von Beiträgen verwenden, sondern auch in Deine Theme-Dateien einfügen. Dort funktioniert die Ausführung von Shortcodes wie im folgenden Beispiel:

```
<?php echo do_shortcode("[beispiel_shortcode]"); ?>
```

Hier kannst Du natürlich genau das gleiche mit Shortcodes machen wie auch in Beiträgen oder in der Sidebar, nur dass Du die Shortcode-Funktionalität eben auch in Bereichen des Blogs nutzen kannst, die außerhalb eines Beitragstextes liegen.

8. Was passiert mit alten, nicht mehr benötigten Shortcodes?

Manchmal wird ein Shortcode obsolet, den Du lange Zeit verwendet hast. Das könnte zum Beispiel passieren, wenn Du per Shortcode Anzeigen in Deine Beiträge eingebaut hast, die Du dort aber nicht mehr haben willst.

Würdest Du nur die Shortcode-Funktion aus der *functions.php* entfernen, blieben die Shortcodes in Deinen Beiträgen stehen. Weil Wordpress sie nicht mehr als funktionierende Shortcodes erkennt, würden sie im Klartext ausgegeben, also *[beispiel-shortcode]* mitten in Deinem Beitrag.



The screenshot shows a WordPress post titled "Beispiel-Beitrag für Shortcodes" with a date of January 13, 2016, by user fneumeier. The main content area contains a paragraph of Lorem Ipsum text with a broken shortcode `[fn-zitat]Das ist ein Beispiel-Zitat.[/fn-zitat]` highlighted in a red box. To the right, there is a widget titled "Shortcode-Test" containing another broken shortcode `[fn-zitat]Das Beispiel-Zitat, diesmal in einem Widget.[/fn-zitat]`, also highlighted in a red box. Below the widget is a "Letzte Beiträge" section with a link to the current post.

So würden Deine Leser einen Shortcode sehen, zu dem Du lediglich die Callback-Funktion gelöscht hast.

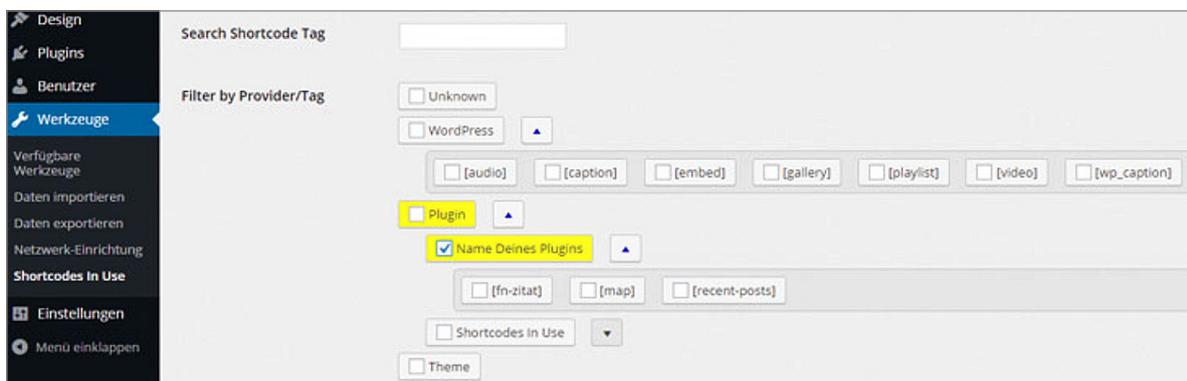
Der sauberste Weg wäre, die Shortcodes aus jedem einzelnen dieser Beiträge zu entfernen. Sind das zu viele und der Aufwand daher zu hoch, kannst Du auch die bisherige Callback-Funktion durch eine Anweisung ersetzen, die „nichts“ zurückgibt. Dann ignoriert Wordpress den Shortcode einfach:

```
add_shortcode( 'beispiel-shortcode', '__return_false' );
```

9. Nützliche und weniger nützliche Shortcode-Plugins

Auf der Wordpress-Website findest Du zahlreiche Plugins, die Shortcode-Sammlungen enthalten. Tendenziell blähen die aber vor allem Deine Installation auf und enthalten viel mehr Shortcodes, als Du je verwenden wirst. Entsprechend zurückhaltend solltest Du solche Plugins einsetzen.

Interessant ist das Plugin **Shortcode Reference**. Es gibt Dir beim Schreiben eines Beitrags einen Überblick zu allen in Deiner Wordpress-Installation verfügbaren Shortcodes inklusive Benutzungshinweisen, soweit die Shortcodes diese Informationen enthalten.



Zeigt verfügbare Shortcodes an: Plugin „Shortcodes in Use“

Shortcodes in Use gibt Dir einen Überblick, in welchen Beiträgen Du welche Shortcodes benutzt hast. Außerdem zeigt es Dir an, wo Shortcodes definiert sind, also beispielsweise in welchem der installierten Plugins.