

WordPress-Templates mit Twig erstellen

Walter Ebert @wltrd

PHPUGFFM 21. Januar 2016

WordPress powers 1 in 5 websites

Posted by **Matthias Gelbmann** on 23 September 2013 in [News](#), [Content Management](#), [WordPress](#)

Summary:

The most popular content management system in the world keeps gaining popularity, and now powers 20% of the web.

Early in 2003 [Matt Mullenweg](#) had just turned 19 when he [faced a problem](#): the "logging software" of his blog, which already attracted a respectable 130 visitors a day, had been abandoned by its main developer. Convinced of the open source model, which - in his words - would enable others to continue should he fall off the face of the planet, he decided to create a fork and started a project which later he called [WordPress](#).

Rather than disappearing from the planet, Matt and WordPress quickly became a role model on how to build a successful business on the basis of an open source product. Matt received countless awards, such as making it into the list of [The 25 Most Influential People on the Web](#). In 2007, aged 23, he [reportedly](#) turned down a \$200 million acquisition offer for [Automattic](#), the company behind WordPress.

As of today, we can report another incredible milestone: 1 in 5 websites now are powered by WordPress.

WordPress powers 25% of all websites

Posted by **Matthias Gelbmann** on 9 November 2015 in [News](#), [Content Management](#), [WordPress](#)

Summary:

The most popular content management systems now runs a quarter of the web. We have a closer look at the detailed statistics behind that success.

Two years after WordPress usage [reached 20%](#), it achieved the next major milestone: 25% of all websites now use [WordPress](#). Or, as its founder Matt Mullenweg puts it: [Seventy-Five to Go](#).

The dominance of WordPress in the CMS market is amazing. The two closest competitors, [Joomla](#) and [Drupal](#) combined are used by 4.9%, less than a fifth of WordPress.

The market share among the 300+ content management system which we monitor is now at 58.7%



The Loop

```
<?php get_header(); ?>  
<?php if ( have_posts() ) : while ( have_posts() ) :  
the_post(); ?>  
  
<h2><a href="<?php the_permalink(); ?>">  
    <?php echo strtoupper( get_the_title() ); ?>  
</a></h2>  
  
<?php if ( has_post_thumbnail() ) the_post_thumbnail(); ?>  
<div class="content"><?php the_content(); ?></div>  
  
<?php comments_template(); ?>  
  
<?php endif; ?>  
  
<?php get_footer(); ?>
```

TWIG

The flexible, fast, and secure
template engine for PHP

[ABOUT](#) [DOCUMENTATION](#) [DEVELOPMENT](#) [CONTRIBUTORS](#)



Twig is a modern template engine for PHP

- **Fast:** Twig *compiles* templates down to plain optimized PHP code. The overhead compared to regular PHP code was reduced to the very minimum.
- **Secure:** Twig has a *sandbox* mode to evaluate untrusted template code. This allows Twig to be used as a template language for applications where users may modify the template design.
- **Flexible:** Twig is powered by a flexible *lexer* and *parser*. This allows the developer to define its own custom tags and filters, and create its own DSL.

Why yet another template engine?

When it comes to template engines in PHP, many people will tell you that PHP itself is a template engine. But even if PHP started its life as a template language, it did not evolve like one in the recent years. As a matter of fact, it doesn't support many features modern template engines should have nowadays:

[LEARN MORE →](#)

[INSTALL NOW ↓](#)

Stable: [1.23.1](#)

Vorteile von Twig

- Trennung von HTML und PHP
- Einfacher für Anfänger
- Mehr Sicherheit
- Vererbung von Templates
- Erfahrung aus anderen PHP-Projekten nutzen

Projekte die Twig nutzen

- Symfony
- Drupal 8
- EZ Publish
- Bolt
- Grav
- Craft
- Slim
- Sculpin
- Satis
- Wallabag
- (Laravel)
- ...

Nachteile von Twig

- Extra Template-Sprache
- Code ist (etwas) langsamer

Twig PHP-Erweiterung

```
git clone https://github.com/twigphp/Twig.git
```

```
cd Twig/ext/twig
```

```
phpize
```

```
./configure
```

```
make
```

```
sudo make install
```

```
php.ini
```

```
[twig]
```

```
extension=twig.so
```

composer.json

```
{  
    "require": {  
        "php": "^5.3.2 || ^7.0",  
        "twig/twig": "^1.23"  
    }  
}
```

composer.json

```
{  
    "require": {  
        "php": "^5.3.2 || ^7.0",  
        "twig/twig": "^1.23",  
        "johnpbloch/wordpress": "*"  
    },  
    "extra": {  
        "wordpress-install-dir": "web"  
    }  
}
```

Twig laden

```
require_once '/path/to/vendor/autoload.php';  
  
$loader = new Twig_Loader_Filesystem('/path/to/templates');  
$twig = new Twig_Environment($loader);  
  
// WordPress Voodoo → $data  
  
echo $twig->render('index.twig', $data);
```



Timber

Helps you create themes faster with sustainable code. With Timber, you write HTML using Mustache-like Templates
<http://timber.upstatement.com>

[Download Version 0.22.1](#)

[Description](#) [Installation](#) [FAQ](#) [Screenshots](#) [Other Notes](#) [Changelog](#) [Stats](#) [Support](#) [Reviews](#) [Developers](#)

Timber cleans-up your theme code so, for example, your php file can focus on being the data, while your twig/html file can focus 100% on the HTML and display.

Once Timber is installed and activated in your plugin directory, it gives any WordPress theme the ability to take advantage of the power of Twig and other Timber features.

Requires: 3.7 or higher

Compatible up to: 4.2.5

Last Updated: 3 hours ago

Active Installs: 8,000+

Ratings

4.5

[Watch 110](#)[Star 2,104](#)[Fork 202](#)

Home

Jared Novack edited this page Sep 12, 2015 · 54 revisions

[Pages 44](#)

New Docs

PHP file basics

Before sending WordPress data to the `.twig` file (where the fun happens), you must set up the page's data inside of your normal theme files (like `index.php`, `single.php`, `archive.php`, etc.). Instead of holding your HTML, these now become configuration files for corresponding `.twig` templates.

```
$data = Timber::get_context();
$data['posts'] = Timber::get_posts();
$data['foo'] = 'bar';
Timber::render('index.twig', $data);
```

Getting Started

- [Video Tutorials](#)
- [Getting Started Pt. 1: Setup](#)
- [Getting Started Pt. 2: Theming](#)
- [Tools for working with Twig](#)

Guides

- [Advanced Custom Fields](#)
- [Caching](#)
- [Configure Template Locations](#)
- [Custom Page Templates](#)
- [Debugging](#)
- [Extend Timber](#)
- [Image Manipulation](#)
- [Menus](#)
- [Pagination](#)
- [Performance & Caching](#)
- [Routing](#)
- [Sidebars](#)
- [Translation](#)

WordPress Packagist

```
{  
    "repositories": [  
        {"type": "composer", "url": "http://wpackagist.org"}  
],  
    "require": {  
        "php": "^5.3.2 || ^7.0",  
        "composer/installers": "~1.0",  
        "johnpbloch/wordpress": "*",  

```

web/wp-config.php

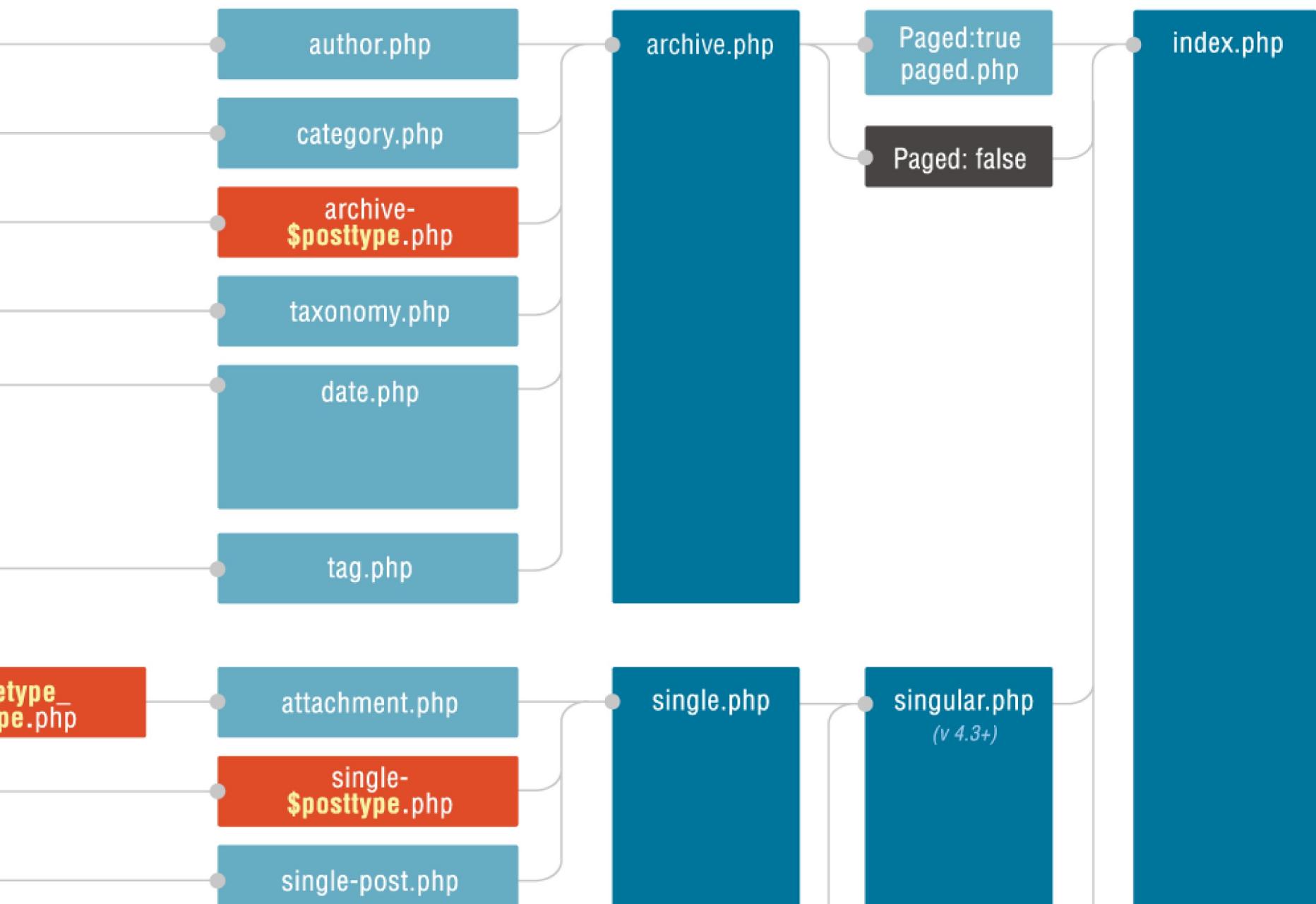
```
require __DIR__ . '/../../vendor/autoload.php';  
...  
  
define( 'WP_HOME', 'http://example.com' );  
define( 'WP_SITEURL', 'http://example.com/wp' );  
define( 'WP_CONTENT_URL', 'http://example.com/wp-content' );  
define( 'WP_CONTENT_DIR', __DIR__ . '/wp-content/' );
```

web/index.php

```
<?php  
define( 'WP_USE_THEMES', true );  
require __DIR__ . '/wp/wp-blog-header.php';
```

.../theme/single.php

```
<?php  
  
$context = Timber::get_context();  
  
$context['post'] = new TimberPost();  
  
Timber::render( 'single.twig', $context );
```



.../theme/functions.php

```
function my_context( $data ) {  
    $data['foo'] = 'bar';  
    $data['menu'] = new TimberMenu();  
  
    return $data;  
}  
add_filter( 'timber_context', 'my_context' );
```

.../theme/page.php

```
<?php  
  
$context = Timber::get_context();  
  
$context['foo'] = 'bar';  
  
$context['menu'] = new TimberMenu();  
  
$context['post'] = new TimberPost();  
  
Timber::render( 'page.twig', $context );
```

Twig-Syntax

`{{ }}` // Ausgabe

`{% %}` // Logik

`{# #}` // Kommentar

.../theme/views/single.twig

```
{# Dies ist ein Beitrag #}
{% extends "base.twig" %}
{% block content %}
<h2>{{ post.title|upper }}</h2>
{% if post.thumbnail %}

{% endif %}
<div class="content">{{ post.content }}</div>
{% include 'comments.twig' %}
{% endblock %}
```

.../theme/views/single.twig

```
{# Dies ist ein Beitrag #}
{% extends "base.twig" %}
{% block content %}
<h2>{{ post.title|upper }}</h2>
{% if post.thumbnail %}

{% endif %}
<div class="content">{{ post.content }}</div>
{% include 'comments.twig' %}
{% endblock %}
```

.../theme/views/single.twig

```
{# Dies ist ein Beitrag #}
{% extends "base.twig" %}
{% block content %}
<h2>{{ post.title|upper }}</h2>
{% if post.thumbnail %}

{% endif %}
<div class="content">{{ post.content }}</div>
{% include 'comments.twig' %}
{% endblock %}
```

.../theme/views/single.twig

```
{# Dies ist ein Beitrag #}
{% extends "base.twig" %}
{% block content %}
<h2>{{ post.title|upper }}</h2>
{% if post.thumbnail %}

{% endif %}
<div class="content">{{ post.content }}</div>
{% include 'comments.twig' %}
{% endblock %}
```

.../theme/views/single.twig

```
{# Dies ist ein Beitrag #}
{% extends "base.twig" %}
{% block content %}
<h2>{{ post.title|upper }}</h2>
{% if post.thumbnail %}

{% endif %}
<div class="content">{{ post.content }}</div>
{% include 'comments.twig' %}
{% endblock %}
```

Posts

```
// Posts  
$context['posts'] = Timber::get_posts();  
  
// WP_Query  
$args = [  
    'post_type'      => 'custom_post_type',  
    'post_status'    => 'publish',  
    'posts_per_page' => 5,  
];  
$context['posts'] = Timber::get_posts( $args );
```

Posts

```
{% for post in posts %}  
  <article id="post-{{ post.ID }}">  
    <h1>  
      <a href="{{ post.link }}>{{ post.title }}</a>  
    </h1>  
    <div class="date">  
      {{ post.date }}  
    </div>  
    <div class="excerpt">  
      {{ post.content|excerpt(55) }}  
    </div>  
  </article>  
{% endfor %}
```

Übersetzungen

```
<?php echo __( 'Sorry, no posts.', 'textdomain' ) ); ?>
```

```
<?php _e( 'Sorry, no posts.', 'textdomain' ); ?>
```

->

```
{{ __('Sorry, no posts.', 'textdomain') }}
```

WordPress-Funktionen

```
<?php get_search_form(); ?>
```

->

```
{{ fn('get_search_form') }}
```

Benutzerdefinierte Felder

```
<h3>{{ post.title }}</h3>
<div class="intro-text">
    {{ post.custom_field }}
</div>
```

Advanced Custom Fields

```
<h3>{{ post.title }}</h3>
<div class="intro-text">
    {{ post.get_field('meins_intro_text') }}
</div>
```

Timber-Klassen erweitern

```
class MySitePost extends TimberPost {  
    var $_issue;  
  
    public function issue() {  
        if (!$this->_issue) {  
            $issues = $this->get_terms('issues');  
            if (is_array($issues) && count($issues)) {  
                $this->_issue = $issues[0];  
            }  
        }  
        return $this->_issue;  
    }  
}
```

Eigene Twig-Filter erstellen

```
add_filter( 'get_twig', function( $twig ) {  
    $twig->addFilter(  
        new Twig_SimpleFilter(  
            'comment_text',  
            function( $text ) {  
                return apply_filters( 'comment_text', $text );  
            } ) );  
  
    return $twig;  
} );
```

Twig-Extensions nutzen

```
add_filter( 'get_twig', function( $twig ) {  
    $twig->addExtension(  
        new Twig_Extensions_Extension_Text()  
    );  
  
    return $twig;  
} );
```

◀ The Twig for Timber Cheatsheet

Posted: October 16, 2015, filed under: Tutorials+Tips, WordPress

[Timber](#) is a library for WordPress that allows you to work with object oriented templates using the Twig templating engine. I've [written about it before](#), and have used it on just about [every project I've done](#) in the last year and a half. I'm even [teaching a class on it!](#)

Anyhow, although Timber ultimately simplifies templating in WordPress, there's a bit of a barrier to entry. If you haven't worked in a templating language before, it takes some time to get a handle on what's possible. The following are some introductory Twig concepts and snippets I've found particularly useful when working with Timber and Advanced Custom Fields.

Conditionals

Let's say you have an ACF option for the display of the header that is either an image or text. You could check like this:



The "_s" for Timber: a dead-simple theme that you can build from

19 commits

2 branches

0 releases

2 contributors



Branch: master ▾

[timber-starter-theme](#) / +



 **jarednova** Merge pull request #7 from Upstatement/travis_tests ... Latest commit d16e6bf on Aug 11

 bin	Add test installation script	2 months ago
 static	renamed /js directory to /static	7 months ago
 templates	Added Travis tests	2 months ago
 tests	Added Travis tests	2 months ago
 .gitignore	Updated tests to (almost) execute properly	8 months ago
 .travis.yml	Add multisite to tests	2 months ago
 404.php	Added files from the jarednova/timber repo	8 months ago
 LICENSE	Added files from the jarednova/timber repo	8 months ago
 README.md	Add Travis badge	2 months ago
 archive.php	Added files from the jarednova/timber repo	8 months ago
 author.php	Added files from the jarednova/timber repo	8 months ago
 composer.json	Added Travis tests	2 months ago
 composer.lock	Added Travis tests	2 months ago
 footer.php	Added files from the jarednova/timber repo	8 months ago

Code

Issues 4

Pull requests 2

Pulse

Graphs

HTTPS clone URL

<https://github.com> 

You can clone with [HTTPS](#) or [Subversion](#). 

 Download ZIP

mini
#1 PHP 5.5.9-1ubuntu4.14

MySQL 5.5.46

Memory Usage 30.467.192 bytes



Request

Object Cache

Timber

index.twig

Called from /home/walter/webprojects/suchblog.info/web/wp-content/themes/suchblog2/index.php

Timber found template: /home/walter/webprojects/suchblog.info/web/wp-content/themes/suchblog2/views/index.twig. Here's the data that you sent:

```
Array
(
    [http_host] => http://dev.suchblog.info
    [wp_title] =>
    [wp_head] => TimberFunctionWrapper Object
        (
            [_function:TimberFunctionWrapper:private] => wp_head
            [_args:TimberFunctionWrapper:private] => Array
                (
                )
        )
        [_use_ob:TimberFunctionWrapper:private] =>

    [wp_footer] => TimberFunctionWrapper Object
        (
            [_function:TimberFunctionWrapper:private] => wp_footer
            [_args:TimberFunctionWrapper:private] => Array
                (
                )
        )
        [_use_ob:TimberFunctionWrapper:private] =>

    [body_class] => home blog logged-in admin-bar no-customize-support debug-bar-maximized
    [site] => TimberSite Object
        (
            [admin_email] => kontakt@suchblog.info
            [blogname] =>
            [charset] => UTF-8
            [description] => Das Blog über Usability und Suchtechnologie
            [id] =>
            [language] => de-DE
            [language_attributes] => TimberFunctionWrapper Object
                (
                    [_function:TimberFunctionWrapper:private] => language_attributes
                    [_args:TimberFunctionWrapper:private] => Array
                        (
                        )
                )
                [_use_ob:TimberFunctionWrapper:private] =>
        )
)
```

Praxisbeispiele

<https://github.com/jarednova/timber/wiki/Showcase>

<https://github.com/laras126/karenmcgrane>

<https://github.com/laras126/mtnmeister-theme>

<https://github.com/laras126/dijifi-theme>

<http://responsivewebdesign.com/toast/backend/>

WordPress-Projekte starten

```
composer create-project roots/bedrock
```

```
composer create-project org_heigl/wordpress_bootstrap
```

```
composer create-project wee/wordpress-project
```

WordPress-Termine

Jeden 2. Dienstag des Monats, WP-Meetup Frankfurt

<https://wpmeetup-frankfurt.de/>

16.-17. April 2016, WordCamp Nürnberg

<https://nuremberg.wordcamp.org/2016/>

24.-26. Juni 2016, WordCamp Europe, Wien

<https://2016.europe.wordcamp.org/>

September/Okttober 2016, WordCamp Frankfurt

<https://frankfurt.wordcamp.org/>

walter.ebert.engineering

@wltrd

walterebert.de

slideshare.net/walterebert