



BURO **BJORN**

Björn Wijers / burobjorn.nl

burobjorn.nl/presentations/nluug-2016

Hi!

My name is Björn Wijers. Burobjorn is the name of my company.

Freelance software developer with an Interaction Design background

burobjorn.nl/presentations/nluug-2016

I'm a freelance software developer with a background in Interaction Design. I've worked as a self-employed professional for more than 10 years now. My clients range from small businesses, large corporations, non-profits, public broadcasters and the Dutch government.

Twelve years of WordPress experience

burobjorn.nl/presentations/nluug-2016

I've been working with WordPress for roughly twelve years.

I've written plugins, both opensource as well as custom in-house, send in patches, created themes, translated and in general try to contribute to the WordPress ecosystem as much as I'm able to.

**#1 WordPress achievement:
Lead-developer of Dutch Tax Services intranet**

burobjorn.nl/presentations/nluug-2016

My biggest achievement thus far with WordPress was being the lead-developer responsible for creating the intranet of The Dutch Tax Services (De Belastingdienst). Off course I didn't do this alone. We did this together as a small team of great people.

Currently:

- Migrating 3500 ASP sites → WordPress
- Voorjebuurt civic crowdfunding platform

And always interested in new cool projects...

burobjorn.nl/presentations/nluug-2016

Currently I'm consulting at a company migrating roughly 3500 websites classic ASP to WordPress, supporting an awesome civic crowdfunding platform called Voorjebuurt and I'm always looking for great new projects to work on.

Besides WordPress I like to tinker with mostly opensource software and a bit of hardware. I also organize a monthly tech/geeky meetup in Utrecht called BEEA. See BEEA.nl

Enough about me.

***“The Good, the Bad and the Ugly
of powering 25% of the Web”***

burobjorn.nl/presentations/nluug-2016


Today I'd like to talk about 'The Good, the Bad and The Ugly of powering 25% of the Web'. So what's with this percentage?

Ps: The title is a pun based on the title of a Western from the late sixties from the previous century featuring Clint Eastwood among others. See <http://www.imdb.com/title/tt0060196/>

'Lies, damned lies, and statistics'

Historical yearly trends in the usage of content management systems for websites

This report shows the historical trends in the usage of content management since January 2011.



	2011 1 Jan	2012 1 Jan	2013 1 Jan	2014 1 Jan	2015 1 Jan	2016 1 Jan	2016 16 Nov
None	76.4%	71.0%	68.2%	64.8%	61.7%	56.6%	53.7%
WordPress	13.1%	15.8%	17.4%	21.0%	23.3%	25.6%	27.1%
Joomla	2.6%	2.8%	2.8%	3.2%	3.0%	3.0%	3.3%
Drupal	1.4%	1.9%	2.3%	1.9%	2.0%	2.1%	2.2%
Magento				0.9%	1.1%	1.2%	1.3%
Blogger	0.6%	0.9%	1.1%	1.2%	1.1%	1.2%	1.1%
TYPO3	0.6%	0.6%	0.7%	0.6%	0.6%	0.7%	0.7%
Bitrix	0.2%	0.3%	0.3%	0.3%	0.4%	0.6%	0.7%
PrestaShop			0.3%	0.4%	0.5%	0.6%	0.6%
Shopify				0.1%	0.3%	0.4%	0.6%

burobjom.nl/presentations/nluug-2016

At the beginning of this year 25% of the top 10 million sites (popularity ranked by Alexa) is powered by WordPress according to W3Techs (<https://w3techs.com/technologies>). At this moment it supposedly has increased towards 27% already, as you can see in the figure on the slide. Personally I have some doubts surrounding this percentage and I'd welcome you to do your own research into these numbers, nonetheless WordPress is popular and you will or already have been in contact with WordPress. So what can you expect in this talk on WordPress?

Source statistics diagram (screenshot taken on 2016-11-16):

https://w3techs.com/technologies/history_overview/content_management/all

What to expect?

burobjorn.nl/presentations/nluug-2016

I'd like to talk to you about WordPress. I'll start with a bit of context on what WordPress is and where it came from. Starting from this historical perspective I'll continue to look into customizing WordPress and introduce topics I assume might be interesting to the audience here.

My goal is to make sure the people here at the NLUUG conference can start developing with / for WordPress and get enough information to seek and find solutions for your particular needs regardless if you are a developer, sysadmin or an interested layman.

I'll introduce to you the Good, The Bad and The Ugly of WordPress.

Oh, do not hesitate to interrupt me to ask questions!

“WordPress is web software you can use to create a beautiful website, blog, or app. We like to say that WordPress is both free and priceless at the same time”

burobjorn.nl/presentations/nluug-2016

So what's WordPress? And what's the background of it?

Nowadays WordPress is a web application which can be used in lots of situation, but it used to be a simple blog application. Let's start at the beginning.

WordPress 1.0

Released: January 3, 2004

burobjorn.nl/presentations/nluug-2016

Let's go back to the first 1.0 release of WordPress. Notable developers at that time are :
Matt Mullenweg & Mike Little.

The technology stack at that time consisted of

Programming Language: PHP4

Database: MySQL 3 - 4

Webserver: Apache 1.3 - 2.0

WordPress was made with a tiny bit of Javascript (mostly for the WYSIWYG editor), loads of html, oldskool CSS styling and plenty of PHP.

It's main purpose: providing you with a simple blog



burobjom.nl/presentations/nluug-2016

To give you an idea of WordPress at that this, here's the admin interface of WordPress 1.0.1 Pretty oldschool, if you'd ask me :)

Source image (downloaded on 2016-11-16):

<http://planetozh.com/blog/2008/12/a-journey-through-five-years-of-wordpress-interface/>

A historical (nerdy) perspective

WordPress 1.0 was launched before...

burobjorn.nl/presentations/nluug-2016

To give you an idea of the context and time in which the WordPress 1.0 was released, let's put things into a historical perspective

It's OK to stare.



burobjom

Apple iPhone combines two must-have items into one revolutionary device. iPhone includes the same iPod functions that you love and adds the technology of a smart phone. Boasting a touch-sensitive screen, full internet capabilities, and 3.5 inch wide screen display, this is the new must-have accessory for your life. Available in 4GB or 8GB models. For more information, visit apple.com/iphone.

iPhone

WordPress 1.0 was released **roughly 3 years before** Apple's first iPhone

Image Source (downloaded 2016-11-16):

<http://www.langleytoday.ca/guest-column-consumer-culture/>



WordPress 1.0 was released **roughly 3 years before** Asus' Eee PC / Netbook which might be considered the device evolved into the current tablets..

Image Source (downloaded 2016-11-16):

https://en.wikipedia.org/wiki/File:ASUS_Eee_White_Alt.jpg

This file is licensed under the Creative Commons Attribution 3.0 Unported license.
Attribution: Red at English Wikipedia

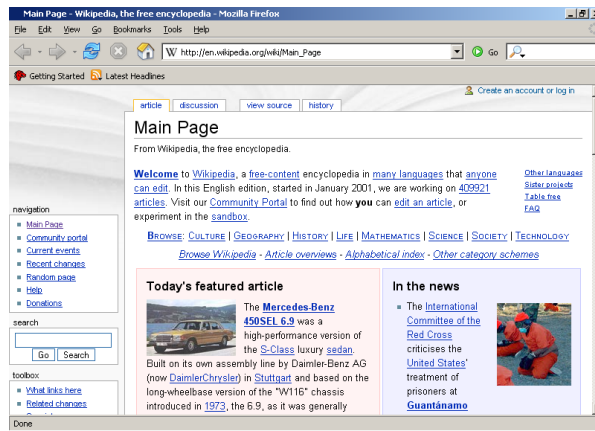


burobjom.nl/presentations/nluug-2016

WordPress 1.0 was released **roughly 2 years before** Microsoft's Internet Explorer 7!!

Image Source (downloaded 2016-11-16):

https://www.microsoft.com/latam/technet/downloads/itdowns/IE7/images/screen4_big.jpg

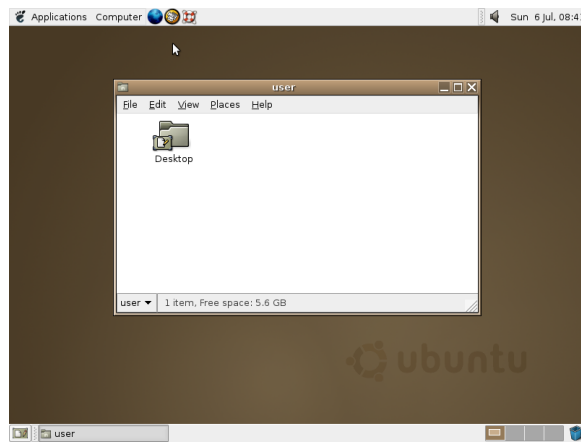


burobjom.nl/presentations/nluug-2016

WordPress 1.0 was released **roughly 10 months before** Mozilla's Firefox 1.0 release

Image Source (downloaded 2016-11-16):

By The original uploader was Minghong at English Wikipedia - Transferred from en.wikipedia to Commons by IngerAlHaosului using CommonsHelper., CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=8922606>



burobjom.nl/presentations/nluug-2016

WordPress 1.0 was released **roughly 9 months before** Ubuntu's first release known as 'Warty Warthog'

Image Source (downloaded 2016-11-16):

By The original uploader was Altonbr at English Wikipedia - Transferred from en.wikipedia to Commons by Shizhao using CommonsHelper., GPL, <https://commons.wikimedia.org/w/index.php?curid=7589578>



burobjom.nl/presentations/nluug-2016

WordPress 1.0 was released roughly **a month before** Facebook was launched..

Image Source (downloaded 2016-11-16):

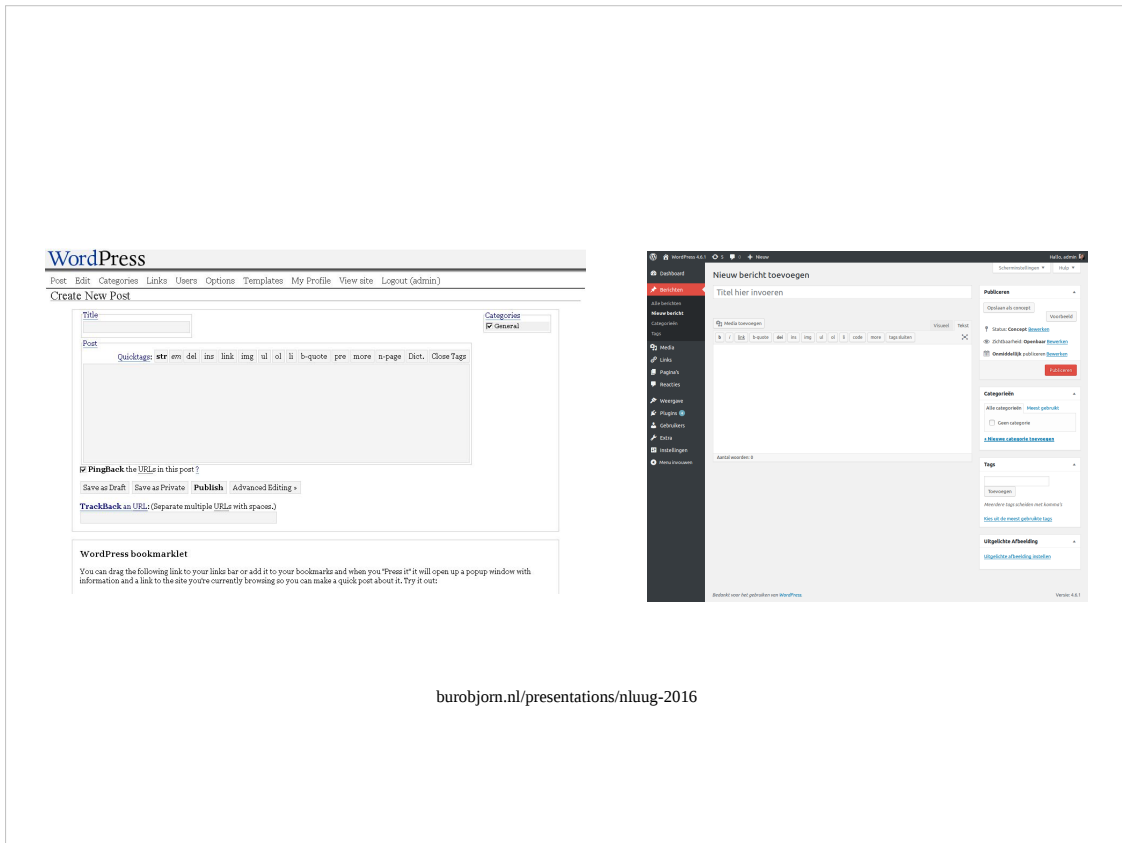
By Source, Fair use, <https://en.wikipedia.org/w/index.php?curid=21400927>.

Now, almost 13 years later...

27% of the Web is powered by it!*

burobjorn.nl/presentations/nluug-2016

So WordPress is pretty old in Internet terms, but still going strong as you can see.



burobjom.nl/presentations/nluug-2016

And lots of it has had the time to evolve as you can see for instance in the UI between 1.0 and the current stable 4.6.1 version.

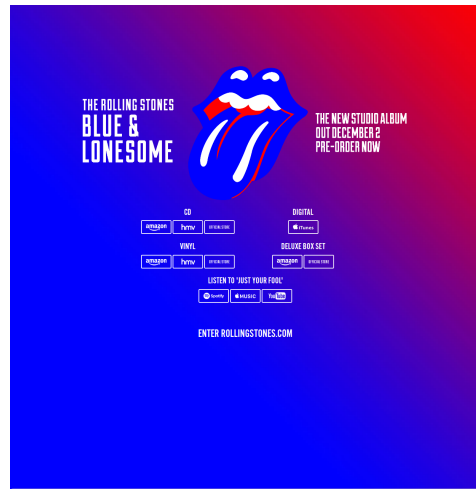
Image (left) Source: (downloaded on 2016-11-16):

<http://planetozh.com/blog/2008/12/a-journey-through-five-years-of-wordpress-interface/>

Examples of “Powered by WordPress”

burobjorn.nl/presentations/nluug-2016

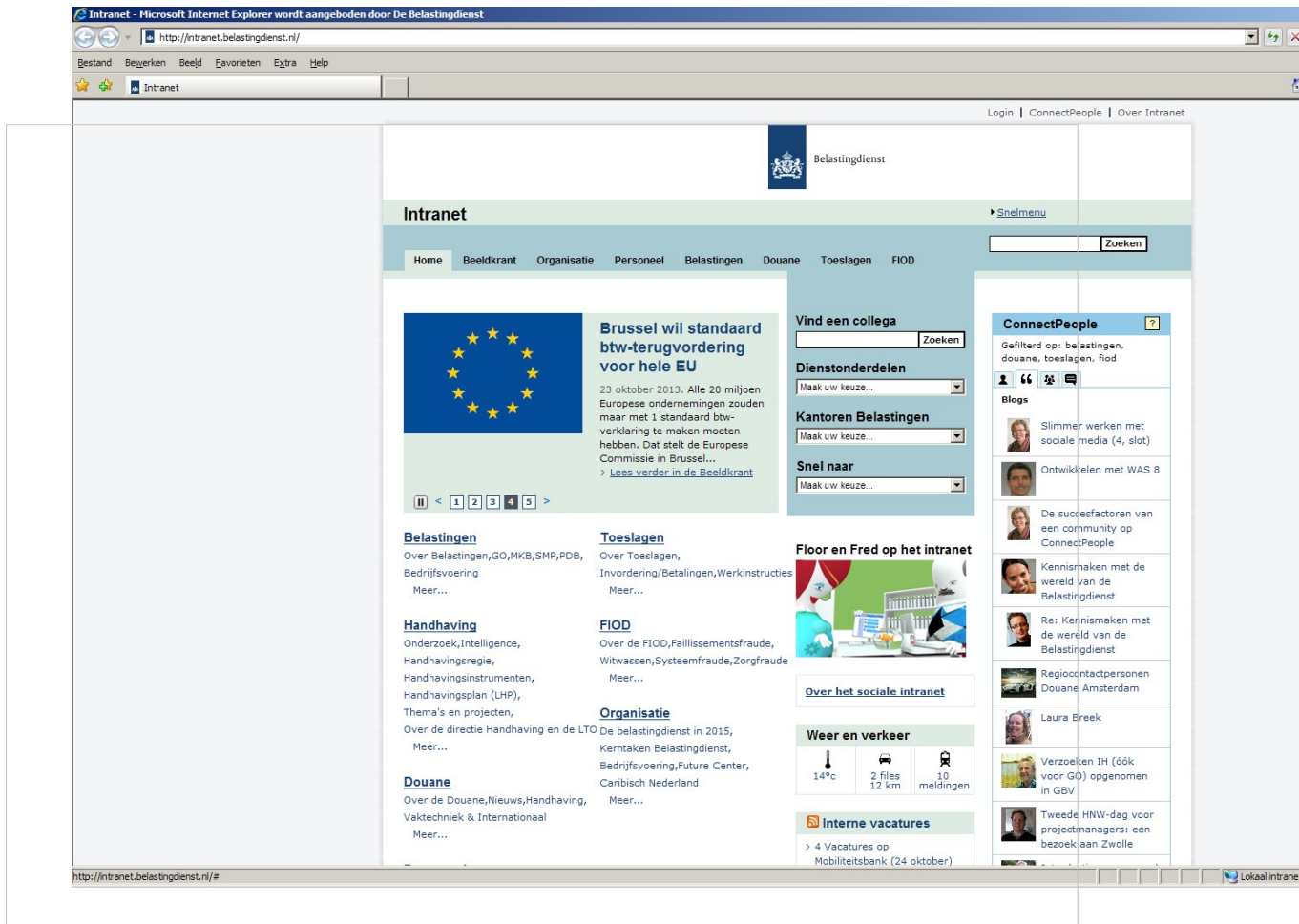
Now to give you an idea of the types of things WordPress is used for. Here are some examples:



burobjom.nl/presentations/nluug-2016

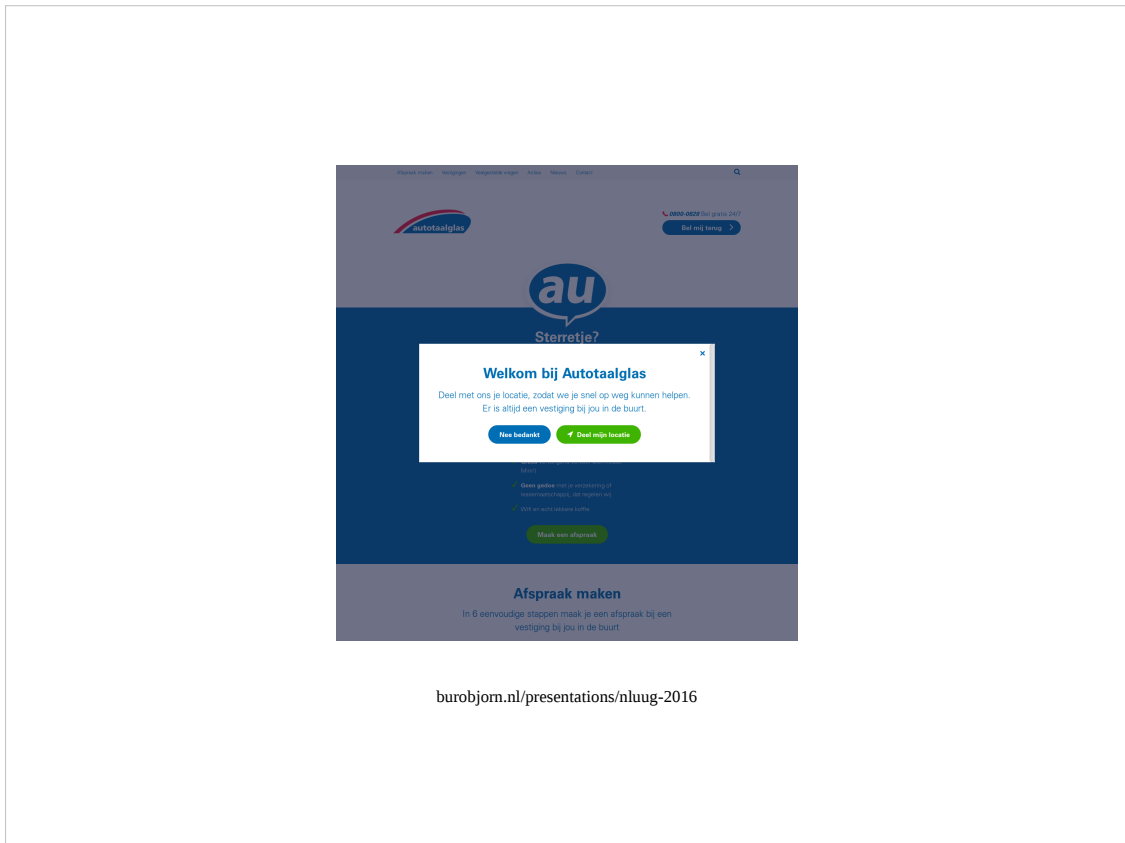
The Rolling Stones website

Image source (screenshot taken on 2016-11-16):
<http://www.rollingstones.com/>.



The Dutch Tax Services intranet

Image Source:
Private URL, screenshot date unknown. Probably somewhere in 2014.



Autotaalglas, one the main companies in the Netherlands to get a broken windshield of your car fixed. This is using WordPress not only for the website, but also for providing the mobile app with data and connects to old AS-400 legacy systems.

Have a look at the interesting usecase presentation by the Dutch company LevelLevel at WordPress.tv (<http://wordpress.tv/2016/10/28/taeke-reijenga-autotaalglas-a-case-study/>)

Image Source (screenshot 2016-11-16): <https://www.autotaalglas.nl>

Blog → CMS → Framework → REST provider

burobjorn.nl/presentations/nluug-2016

So WordPress evolved into something more than just a blogging application. It became a CMS, framework and nowadays since the integration of a REST API in version 4.4: a REST provider for other applications such as mobile apps.

***Basic ingredients of WordPress
are still the same...***

burobjorn.nl/presentations/nluug-2016

However the basics are still the same..

PHP
CSS
HTML
JAVASCRIPT

burobjorn.nl/presentations/nluug-2016

I've underlined Javascript because the role of Javascript in WordPress has increased (and probably will increase even more) in the last few years.

***Basic requirements of WordPress
are also still the same...***

burobjorn.nl/presentations/nluug-2016

A webserver (Apache/nginx)
A database server (MySQL/mariaDB)

burobjorn.nl/presentations/nluug-2016

Notice that the recommended servers now include nginx and mariaDB.

The latter is probably due to the influence of Oracle on MySQL and their history of killing any opensource projects they touch.

The Good

WordPress is a software application
In production for almost thirteen years.
Battle-hardened & user-refined
by numerous people & projects.

burobjorn.nl/presentations/nluug-2016

The Bad

WordPress' code is thirteen years old legacy code
written by lots of different people
with different skill sets...and it shows.

burobjorn.nl/presentations/nluug-2016

Although there is an ongoing process of refactoring and fixing the legacy code with better code it's far from done...

The Ugly

Improving & maintaining thirteen year old legacy code
requires stamina & perseverance...

burobjorn.nl/presentations/nluug-2016

It attracts & NEEDS a different type of person than those interested in the latest shiny technology.

Customize WordPress?

stay within /wp-content

burobjorn.nl/presentations/nluug-2016

Okay, now that you've gotten an idea of where WordPress is coming from how can you change it to fit your needs? First rule: you may only change things within the wp-content directory!

Plugins

burobjorn.nl/presentations/nluug-2016

You may add, remove plugins which add or remove WordPress functionality

Myplugin.php

```
<?php
/*
  Plugin Name: My Plugin
  Plugin URI: http://plugin_URI.ext
  Description: My plugin is a mockup plugin
  Version: 1.0
  Author: BjornW
  Author URI: http://Auteur_URI.ext
  License: GPL2
*/
?>
```

burobjorn.nl/presentations/nluug-2016

This is the most basic plugin. It does nothing at all, but it will be parsed by WordPress as a plugin due to the use of these 'magic comments'. Using these lines WordPress knows that the file is a WordPress plugin.

MustUse-Plugins

(no directories..)

burobjorn.nl/presentations/nluug-2016

There are also mu-plugins. These were part of WordPress Multisite (hence mu-plugins) but have been integrated into WordPress since version 3.0 in which WordPress Mu became part of the WordPress codebase (one codebase → easier to maintain).

The 2 most important things to know about these are:

- 1) They will run automatically and you cannot de-activate them from the wp-admin interface.
- 2) You cannot place these in subdirectories in wp-content/mu-plugins unless you write some code to load the files in the subdirectories.

WordPress MU?

burobjorn.nl/presentations/nluug-2016

So there was a time when there were 2 WordPress versions: WordPress and WordPress MU.

WordPress MU allowed you to use one codebase for multiple sites. So you could have multiple websites being run from the same codebase. It would use the same database and just add tables per site to it

WordPress has been merged into WordPress since version 3. and you can use WordPress to power multiple sites in a network. Have a look at the WordPress documentation for the details on how to set this up. It's a bit more advanced than a basic WordPress installation.

Themes

burobjorn.nl/presentations/nluug-2016

Themes also reside in the wp-content directory and are used to change the look and feel of the WordPress frontend. So it allows you to change the look and feel of your website for your visitors.

Style.css & empty index.html

```
/*  
Theme Name: Minimaal  
Theme URI: https://burobjorn.nl  
Author: Björn Wijers <burobjorn@burobjorn.nl>  
Author URI: https://burobjorn.nl  
Description: Most minimal theme  
Version: 1.0  
License: GNU General Public License v2 or later  
License URI: http://www.gnu.org/licenses/gpl-2.0.html  
Tags: example  
Text Domain: twentyfifteen  
*/
```

burobjorn.nl/presentations/nluug-2016

The most basic theme consists of a style.css file with the 'magic comments' as seen on the slide and an empty index.php file. It does nothing and will display a white screen when visiting your website, but WordPress executes it as a valid theme without a problem.

Plugins === Themes

burobjorn.nl/presentations/nluug-2016

Under the hood, plugins and themes are very much the same. And in general you can achieve most of what you want in both a plugin or a theme.

The Good

Freely available on WordPress.org:

47734 plugins

4304 themes

burobjorn.nl/presentations/nluug-2016

The Bad
47734 plugins
& 4304 themes
Without any quality assurance...

burobjorn.nl/presentations/nluug-2016

The ugly

Each plugin or theme has all rights
within your installation and
may be auto-updated* or 'phone home*' ...

burobjorn.nl/presentations/nluug-2016

Basically you hand over the keys of the Kingdom when you are installing plugins or themes. So make sure you can trust a plugin or theme before installing it!

Actions & Filters

'event listeners'

burobjorn.nl/presentations/nluug-2016

Without actions and filters you would not be able to easily change the default behavior of the WordPress core. WordPress' actions and filter are similar to even listeners. They allow you to hook into a certain event and add & remove code or in the case of filters replace certain values to fit your needs.

Nowadays WordPress has so many actions and filters that you should never have to patch WordPress' code in order to change it.

Action (add/remove) example

```
/** Assume a well-written class **/  
  
function __construct() {  
    add_action('before_delete_post', array( $this, 'my_function'), 10, 1);  
}  
  
function my_function ($post_id) {  
    // use the post_id to perform actions before the delete has been run  
}
```

burobjorn.nl/presentations/nluug-2016

The code on the slide is a simplistic example of how to implement a basic action from within a class.

In this case we'll hook into the 'before_delete_post' hook. This will execute just before a post is removed from WordPress and allows you to for instance make a copy of the the post before the original is removed.

In the code you can see that this action will call the 'my_function' method in our assumed class.

The number 10 indicates the priority of the action. As an unwritten rule anything below 10 is WordPress Core territory and anything above 10 is plugin/themes territory. By hooking into an action with a lower or higher priority you are able to execute your code just before someone else's code (or just after it).

The number 1 indicates the number of arguments your custom function needs to receive.

Filter (replace) example

```
/** Assume a well-written class */  
  
function __construct() {  
    add_filter( 'wp_mail_from_name', array( $this, 'set_from_name' ) );  
}  
  
function set_from_name ( $current_name ) {  
    return 'Bjorn';  
}
```

burobjorn.nl/presentations/nluug-2016

On this slide I've written a small example of how to use a filter. The filter is called 'wp_mail_from_name' and allows you to set a name in the From header of any mail sent from WordPress.

As you might have noticed, the numbers 10 and 1 are absent. By default WordPress will use priority 10 and the amount of arguments the filter has. So in this case I know that there will be only 1 argument and I have omitted this from the code.

The 'set_from_name' method is called each time the 'wp_mail_from_name' filter is executed and it will receive the current from name. In this case it will be ignored and the code will return my (hardcoded) name.

Keep in mind filters need to return values! Without it you may break things as filters are often chained together. So don't break the chain :)

The Good

*Almost anything can be changed in WordPress
without touching the default (Core) code*

burobjorn.nl/presentations/nluug-2016

The Bad

Performance issues and side-effects will occur if you're doing this the wrong way.

burobjorn.nl/presentations/nluug-2016

Sadly it's rather easy to do this the wrong way. I recommend to always look at the WordPress code as a reference since most documentation cannot be trusted due to the high speed of development. In other words: the code is the documentation.

The Ugly

It is hard to determine when and which action or filter to use.

burobjorn.nl/presentations/nluug-2016

You may use the WordPress codex

(https://codex.wordpress.org/Plugin_API/Action_Reference), but I also recommend to look at the code itself to get a better idea of when and which action to use.

Customizer jargon:

Custom Post Type

Custom Taxonomy

Attachments (media files)

Roles / Capabilities (Users)

Short Codes

Templates (Page)

Metadata (Post 'customfields' / User / Taxonomy / Site 'options')

burobjorn.nl/presentations/nluug-2016

There's more than just actions and filters off course. To give you an idea I've put up some jargon used in the WordPress world.

Custom Post Types can be considered as data types molded to your liking. The basis of these types are the general Post/Page type. Have a look at make.wordpress.org to learn more about it.

Custom Taxonomy allows you to combine and group Custom Post Types into taxonomies. Out of the box WordPress has the category and tags taxonomy, but you can add your own. For instance a custom taxonomy 'genre' which can group your custom post type 'films' into the appropriate genres.

Attachments are media types such as images, videos, pdf files etc. Have a look at the WordPress codex for a bit more background information on these. Oh and in 4.7 PDF files will get previews which makes it easier to work with them.

Shortcodes are basic tokens you may insert in your content which will be replaced with other content using code. Try to avoid them if you can, since using shortcodes will litter your database with tokens that may not be replaced if a plugin or theme has been disabled.

A theme consists of Templates. A lot of them are predefined, such as `single.php` for a single Post or Page, but you can also add your own for instance as Page Templates. This allows you to switch the layout or usage of a WordPress page. I'm quite fond of these and prefer to use them instead of shortcodes.

There's a lot of places where you can add extra data (metadata) to a data entity. In general metadata is defined by a key-value pair and the id of the corresponding entity. For instance a site id, a user id, taxonomy term id or a post id. Have a look at it on the WordPress Codex. It enables a lot of interesting use cases!

REST API

burobjorn.nl/presentations/nluug-2016

Since version 4.4 WordPress has a REST api built in, which allows you to connect WordPress to other applications via REST. For example, you may use WordPress to add data to your mobile application which retrieves this data via REST.

The Good

Use WordPress as a REST provider for other applications such as mobile apps.

burobjorn.nl/presentations/nluug-2016

The Bad

It too soon to have bad things yet ;)

burobjorn.nl/presentations/nluug-2016

The Ugly
It's not yet feature complete and quite new.

burobjorn.nl/presentations/nluug-2016

Not really ugly...it's a work in progress..

WordPress auto-updater

burobjorn.nl/presentations/nluug-2016

WordPress has an auto-updater since version 3.7 (?).

It will keep your WordPress installation up to date. It works similar to Chrome & Firefox's auto-update mechanism. Off course it relies on having access to the updates server (single-point-of-failure) and having write permission.

You are also able to disable it in WordPress (although I'd disable it via permissions in your filesystem and firewall just to be sure). By default it will not update plugins or themes automagically although you can configure WordPress in such a way that it will update those as well. Translations are updated automatically.

For simple WordPress sites without a lot of plugins or themes this works 99% of the time without any issues, for more complex websites I'd prefer other update methods (for instance via Git or wp-cli).

The Good
Automatically get the latest version

burobjorn.nl/presentations/nluug-2016

And make sure your simple website has the latest security updates...

The Bad

Third-party plugins or themes
may break your site...

burobjorn.nl/presentations/nluug-2016

Although personally I've had never any problems with the auto-updater. This is probably because I only use it for simple websites and not for more complex setups.

The Ugly

The auto-update mechanism needs write permission to replace the old version...

burobjorn.nl/presentations/nluug-2016

Yeah...there's always a trade-off between security and ease of use.

WP-CLI

burobjorn.nl/presentations/nluug-2016

WP-CLI (wp-cli.org) is a commandline application which allows you to use all of WordPress' admin interface functionality and more from the commandline. And it's scriptable!

It makes it really easy to setup and configure a WordPress installation from for instance a configuration tool like Ansible.

The Good

A command-line (scriptable)
client for WordPress' admin interface.

burobjorn.nl/presentations/nluug-2016

The Bad
It's not included by default

burobjorn.nl/presentations/nluug-2016

It should be part of core WordPress.

The Ugly

Support for third-party plugins or theme functionality may not be supported or break wp-cli

burobjorn.nl/presentations/nluug-2016

As always, bad plugins or themes may rain on your parade...

Database

burobjorn.nl/presentations/nluug-2016

A few words about WordPress' database use.

It is not using a database abstraction layer which makes it hard to transition from for instance MySQL to Postgress. Although to be honest, I've never had to move from one database to another during a project's lifecycle.

It is using one object to deal with the database, which enables you to replace it with another one. For instance HyperDB which is designed with master-slaves and all kinds of more 'enterprise' use of database. You can download it from WordPress.org

In a multisite installation you'll still use one database with 12 tables per sites besides some overhead tables. Yes, it looks ugly, but it has proven itself over the years as the more scalable solution.

WordPress places serialized data in the database which sucks for database migrations from test to production for instance.

The Good

Fairly simple database model with 12 tables
accessible via one global (ouch) object (\$wpdb)
acting as a 'abstraction layer'

The Bad

Using serialized arrays and URLs all over the place makes deployment & database harder* than it IMHO should be.

burobjorn.nl/presentations/nluug-2016

WP-DB-Migrate is a great Python script to help you with migrating databases.

The Ugly

WordPress MU installations use 12 tables per site + mu overhead tables. Lots of sites = lots of tables.

Security

burobjorn.nl/presentations/nluug-2016

WordPress security or to be honest web application security in general is a topic in itself. However I'd say that WordPress has seen some good changes in this area and might be considered pretty ok nowadays. Unless you start installing plugins and themes..these may or may not be secure. And more often those claiming to help security are the ones causing the most issues..

The Good

Default WordPress is pretty ok*, nowadays

burobjorn.nl/presentations/nluug-2016

The Bad

It's still a web application, so secure it by all means necessary on different levels. Be careful with plugins & themes!

burobjorn.nl/presentations/nluug-2016

Security of webapplications is a multi-layered beast which depends on so many variables that it is hard to give any advice in a talk like this. So I'd recommend to have a good look at OWASP and general web application security best practices!

The Ugly

WordPress leaks lots of information by default
and has no brute-force protection by default

burobjorn.nl/presentations/nluug-2016

At least, install some kind of brute-force protection against brute-force login attacks. A possible solution is wpfail2ban plugin in conjunction with fail2ban. See <https://bjornjohansen.no/using-fail2ban-with-wordpress>

wp-config.php

burobjorn.nl/presentations/nluug-2016

The file `wp-config.php` is arguable the most important file in a WordPress installation. It holds all configuration for your WordPress installation and you can use it to modify certain things, such as setting the language of the installation or to enable debugging. In the next slide I will discuss some useful constants which allow you to tune your WordPress installation. For more information have a look at https://codex.wordpress.org/Editing_wp-config.php

```
// disable external requests
define( 'WP_HTTP_BLOCK_EXTERNAL', true );
define( 'WP_ACCESSIBLE_HOSTS', 'api.wordpress.org,*.github.com' );

// disable auto-updater
define( 'AUTOMATIC_UPDATER_DISABLED', true );

// disable file editor & plugin theme updates
define( 'DISALLOW_FILE_MODS', true );

// disable file editor only
define( 'DISALLOW_FILE_EDIT', true );

// set trash time
define( 'EMPTY_TRASH_DAYS', 3 ); // 3 days

// set debugging true also for js scripts
define( 'WP_DEBUG', true );
define( 'SCRIPT_DEBUG', true );
```

burobjorn.nl/presentations/nluug-2016

The first 2 constants defined in this slide allow you to limit connections to the outside world. Although I personally would not only use this, but also limit connections via a good-old firewall.

The third constant allows you to disable the auto-updater and the fourth even removes any traces of being able to update from within the WordPress interface.

The fifth 'DISALLOW_FILE_EDIT' should imho by default part of your wp-config.php file. This will prevent anyone with access to wp-admin to use an editor to change files. Personally I'd like the editor to be removed and made into a plugin.

The 'EMPTY_TRASH_DAYS' constant allows you to set the time after which the WordPress trash bin will be emptied. By default this is 30 days.

The last 2 constants help you with debugging. On the https://codex.wordpress.org/Editing_wp-config.php you will see more ways of making debugging even easier.

The Good

There are lots of extra options to configure
WordPress to your liking.

burobjorn.nl/presentations/nluug-2016

The Bad

The documentation is not always clear and you may cause unwanted side-effects.

burobjorn.nl/presentations/nluug-2016

The Ugly

It's a PHP file...

burobjorn.nl/presentations/nluug-2016

So if you're not fluent with PHP you may make a syntax error and your site will fail to load...

**Questions? Remarks?
Let me know!**

...You may also ask me later..

*@Bjorn_W
buobjorn@buobjorn.nl
<https://buobjorn.nl>*

buobjorn.nl/presentations/nluug-2016

Feel free to contact me with any questions or remarks related to this presentation.

Thank you!

burobjorn.nl/presentations/nluug-2016

Thanks for attending my presentation. Have fun with the other talks today!

Feel free to share this presentation!

Licensed under: CC-BY-SA-NC

Attribution / Credits:

Björn Wijers, <https://burobjorn.nl>

Commercial re-use? Please contact me.

burobjorn.nl/presentations/nluug-2016

The images & code examples used in this presentation are licensed under their respective licenses and not necessarily covered by the license on this side



BURO**BJORN**

burobjorn.nl/presentations/nluug-2016