

WordPress for developers

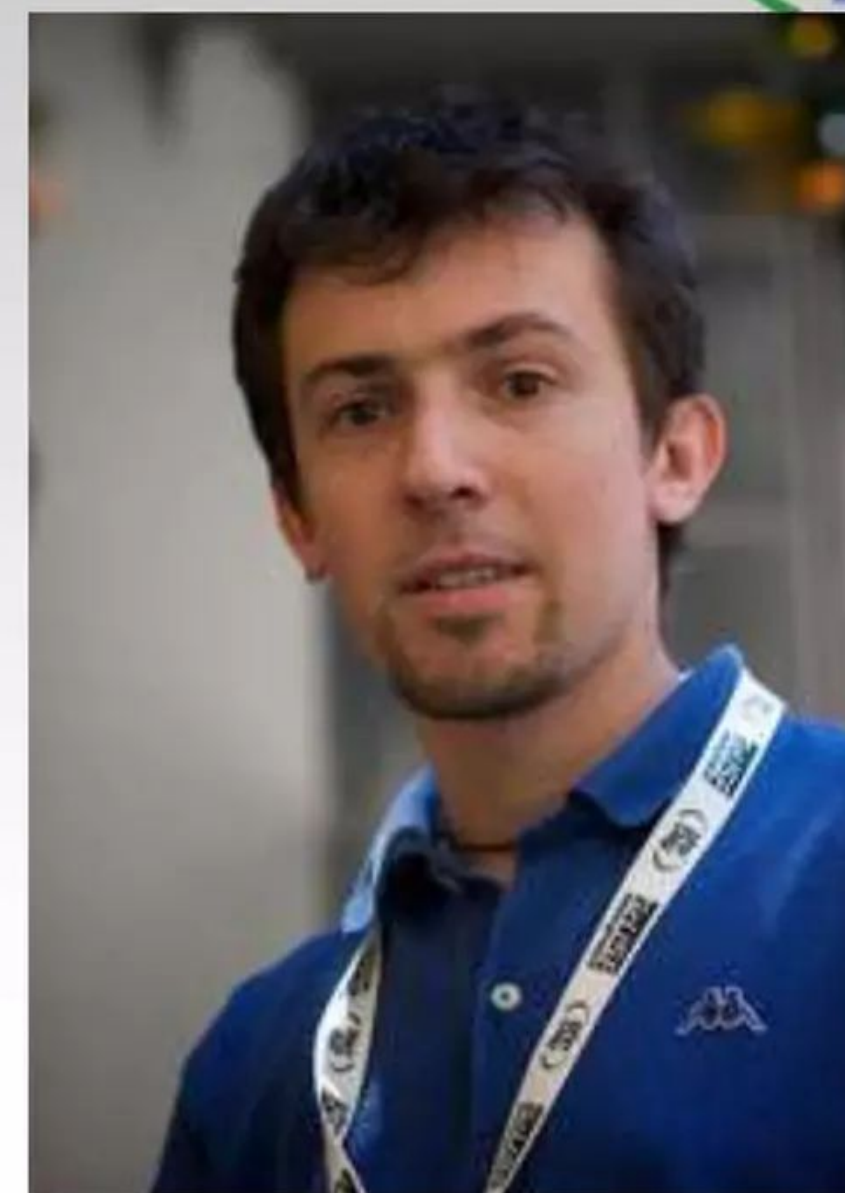
Maurizio Pelizzone



May 14, Verona 2011

About me

- 35 years old
- Birth in Turin (Italy)
- Co-Founder mavida.com
- ***WordPress added***



- <http://maurizio.mavida.com>
- <http://www.linkedin.com/in/mauriziopelizzone>



About this speech

- About WordPress
- WordPress features overview
- Custom type and Taxonomy
- Routing and rewrite rules
- Custom query and manipulation
- Cache tips
- Debugging tools
- Links and reference



About WordPress

“WordPress is a web software you can use to create a beautiful website or blog” *

“WordPress is a state-of-the-art publishing platform with a focus on aesthetics, web standards, and usability. ”

- 25 million people have chosen WordPress
- 54 % content management system market share
- More then 14,000 plugins (May 2011)
- More then 1.300 themes (May 2011)
- 32 million WordPress 3 download (February 2011)
- More then 9 million WordPress 3.1 download (May 2011)



Releases timeline history

1.2	Mingus	22 May 2004
1.5	Strayhorn	17 February 2005
2.0	Duke	31 December 2005
2.1	Ella	22 January 2007
2.2	Getz	16 May 2007
2.3	Dexter	24 September 2007
2.5	Brecker	29 March 2008
2.6	Tyner	15 July 2008
2.7	Coltrane	11 December 2008
2.8	Baker	10 June 2009
2.9	Carmen	19 December 2009
3.0	Thelonious	17 July 2010
3.1	Django	23 February 2011
3.2		30 June 2011



Is WordPress better than Drupal or Joomla ?



No, it's different!!!



now some boring slides.
sorry...



WordPress users (my experience)

Publisher: > 50%

Designers: < 30%

Developers: ~ 15%



sample of 143 blogs
statistics based on my personal contacts

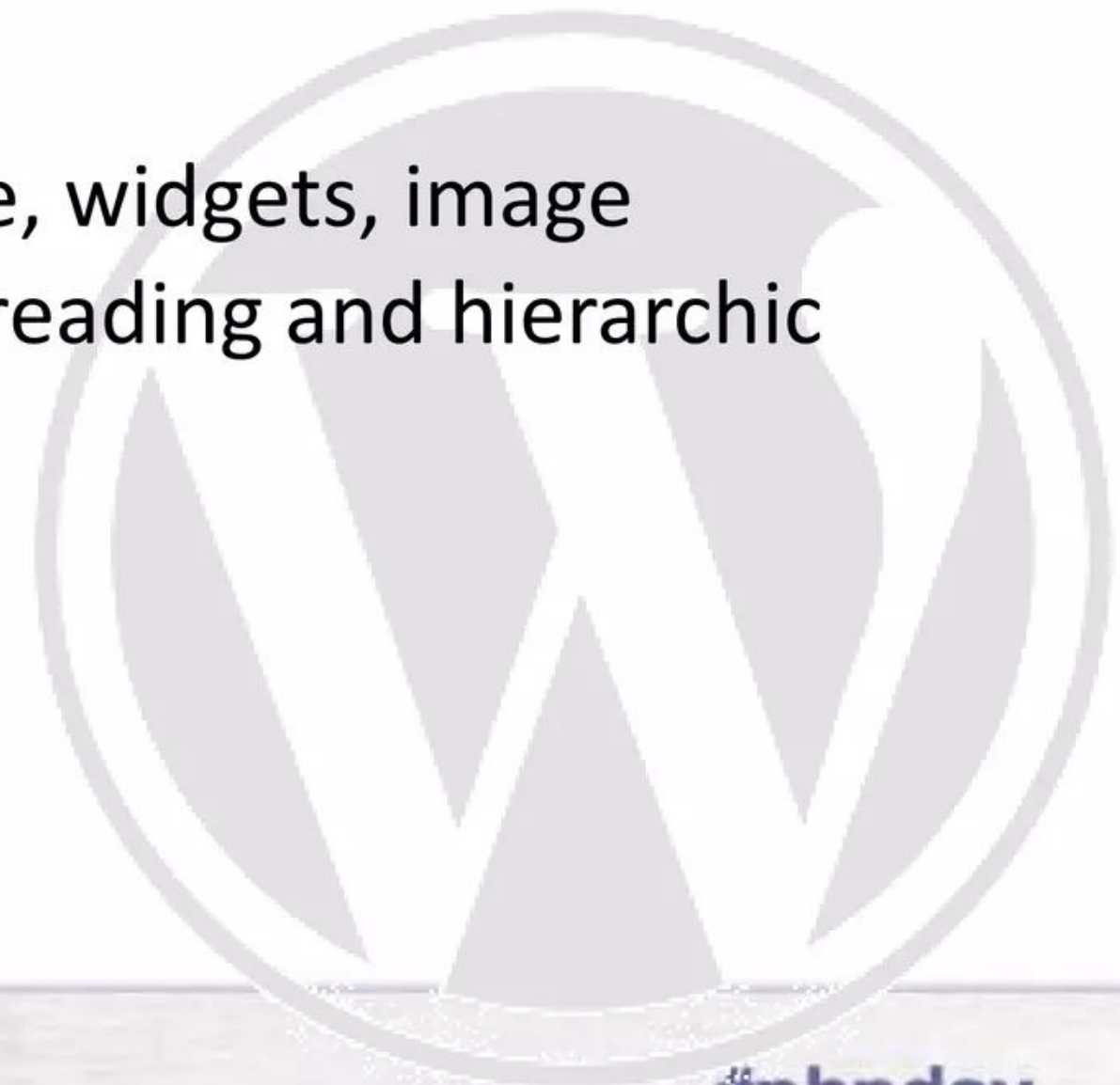
What else?

- Strong backward compatibility
- Documentation (<http://codex.wordpress.org/>)
- Free themes and plugins direcorey
- Super fast template manipulation
- Automattic



Main Features

- One click automatic update (core, plugins, themes)
 - Core multisite implementation
 - Custom post type
 - Custom taxonomies
 - XML-RPC interface
 - Child themes
-
- Some other staff like: oembed, shortcode, widgets, image editing, automatic thumbnails, comments threading and hierarchic menu generator



WordPress Weaknesses

1. Globals variable
2. Not fully OO
3. EZSQL DB Class
4. Uneasy Unit Test
5. Need tuning and maintenance (very often)



About Custom Post Type

“**Post type** refers to the various structured data that is maintained in the WordPress posts table and can represent any type of content you want”

Default native post type

1. post
2. page
3. attachment
4. revision
5. nav-menu-item (> wp 3.0)



[//http://codex.wordpress.org/Function_Reference/register_post_type](http://codex.wordpress.org/Function_Reference/register_post_type)

```
add_action( 'init', 'add_post_type' );
```

```
function add_post_type( ) {
```

```
    $args = array(
        'label' => __('films'),
        'public' => true,
        'hierarchical' => false,
    )
```

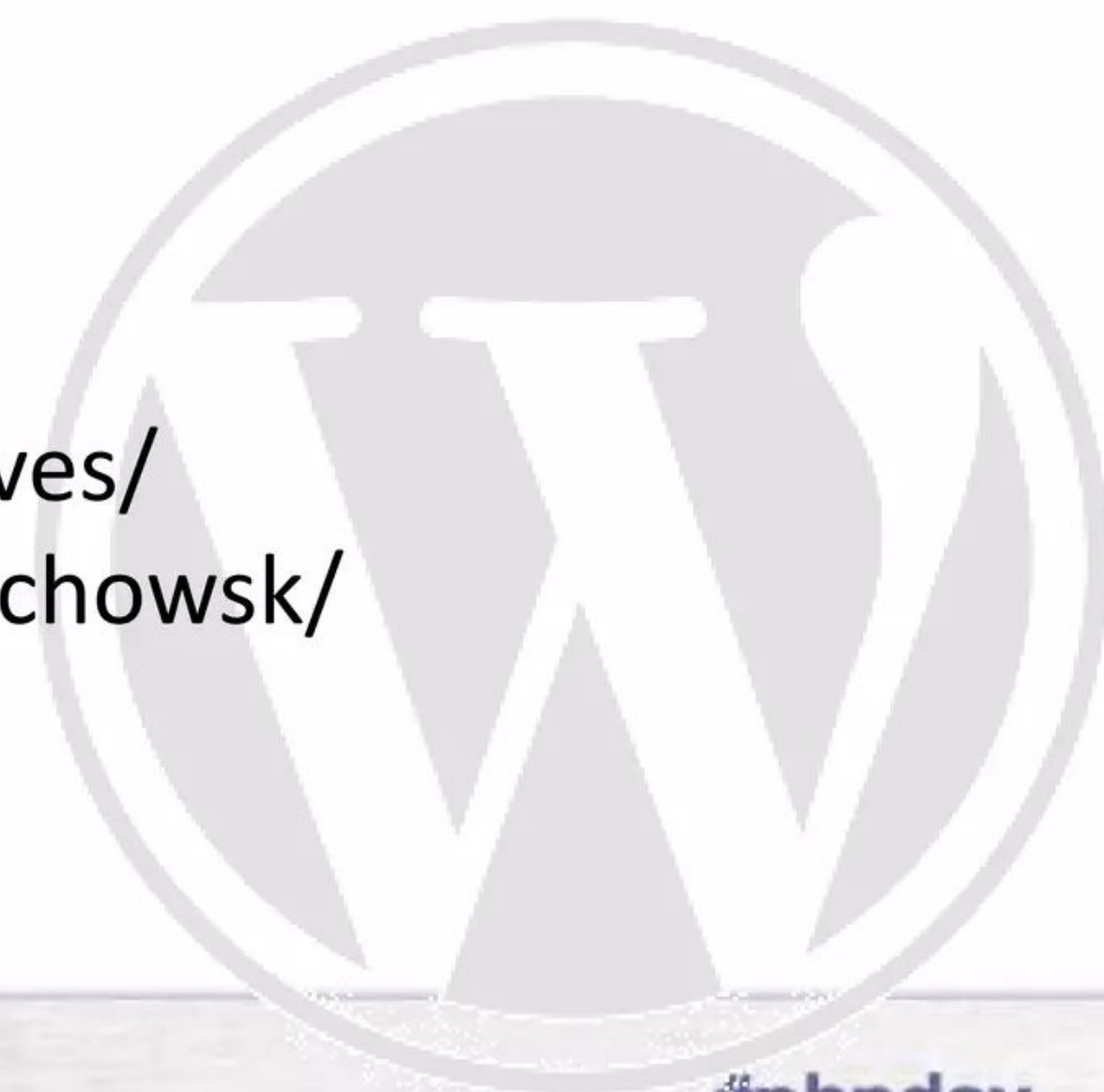
```
    register_post_type( 'film', $args );
```

```
}
```

About Taxonomies

“**Taxonomy** is the practice and science of classification.” (Wikipedia)

- More simple content organization
 - Books (Genre, Authors, Publisher, Edition)
 - Films (Genre, Actors, Director, Year)
- Greater semantics
 - <http://blogname.com/film/matrix/>
 - <http://blogname.com/genre/action/>
 - <http://blogname.com/actors/keanu-reeves/>
 - <http://blogname.com/director/andy-wachowsk/>



[//http://codex.wordpress.org/Function_Reference/register_taxonomy](http://codex.wordpress.org/Function_Reference/register_taxonomy)

```
add_action( 'init', 'add_taxonomies' );
```

```
function add_taxonomies( ) {
```

```
    $args = array(
        'hierarchical' => true,
        'public' => true,
        'label' => 'Genre',
    )
```

```
    register_taxonomy( 'genre', array('post', 'book', $args );
```

```
}
```

About Routing and Rewrite

“WP_Rewrite is WordPress' class for managing the rewrite rules that allow you to use [Pretty Permalinks](#) feature. It has several methods that generate the rewrite rules from values in the database.”



[//http://codex.wordpress.org/Function_Reference/WP_Rewrite](http://codex.wordpress.org/Function_Reference/WP_Rewrite)

```
add_action('generate_rewrite_rules', 'my_rewrite_rules');
```

```
function my_rewrite_rules( $wp_rewrite ) {
```

```
    $new_rules = array(
```

```
        'favorite-films' => 'index.php?post_type=films&tag=favorite',
```

```
    );
```

```
    $wp_rewrite->rules = $new_rules + $wp_rewrite->rules;
```

```
}
```

[//http://codex.wordpress.org/Function_Reference/WP_Rewrite](http://codex.wordpress.org/Function_Reference/WP_Rewrite)

```
add_action('generate_rewrite_rules', 'my_rewrite_rules');
```

```
function my_rewrite_rules( $wp_rewrite ) {

    $new_rules = array(

        'favorite-films/([^/]*)' => 'index.php?post_type=films&tag=favorite&genre='
                                . $wp_rewrite->preg_index(1),

        'favorite-films' => 'index.php?post_type=films&tag=favorite',

    );

    $wp_rewrite->rules = $new_rules + $wp_rewrite->rules;

}
```

[//http://codex.wordpress.org/Function_Reference/WP_Rewrite](http://codex.wordpress.org/Function_Reference/WP_Rewrite)

```
add_action('generate_rewrite_rules', 'my_rewrite_rules ');
add_filter('query_vars', 'my_query_vars');
add_action('template_redirect', 'my_redirect' );

function my_rewrite_rules ( $wp_rewrite ) {
    $new_rules = array(
        'advanced-search' => 'index.php?custom_route=advanced-search',
    );

    $wp_rewrite->rules = $new_rules + $wp_rewrite->rules;
}

function my_query_vars($vars) {
    $vars[] = 'custom_route';
    return $vars;
}

function my_redirect() {
    if ( get_query_var('custom_route') != "" ) :
        $template = array( get_query_var('custom_route') . ".php" );
        array_push( $template, 'index.php' );
        locate_template( $template, true );
        die();
    endif;
}
```

[//http://codex.wordpress.org/Function_Reference/WP_Rewrite](http://codex.wordpress.org/Function_Reference/WP_Rewrite)

```
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteRule ^index\.php$ - [L]
RewriteRule ^login /wp-login.php [QSA,L]
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [L]
</IfModule>
```

[//http://codex.wordpress.org/Function_Reference/WP_Rewrite](http://codex.wordpress.org/Function_Reference/WP_Rewrite)

```
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteRule ^index\.php$ - [L]
RewriteRule ^login /wp-login.php [QSA,L]
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [L]
</IfModule>
```

About Custom Query and wpdb

“WordPress provides a class of functions for all database manipulations. The class is called wpdb and is loosely based on the ezSQL class written and maintained by Justin Vincent.”



[//http://codex.wordpress.org/Function_Reference/wpdb_Class](http://codex.wordpress.org/Function_Reference/wpdb_Class)

```
function list_last_film( $limit ) {

    global $wpdb;

    $sql_source = "SELECT ID, post_title
                  FROM $wpdb->posts
                  WHERE post_type = 'films' and post_status = 'published'
                  ORDER BY post_date
                  LIMIT %d";

    $sql = $wpdb->prepare($sql_source , $limit );

    $last_films = $wpdb->get_results( $sql );

    foreach ( $last_films as $film_item) {
        echo $film_item->post_title;
    }

}
```

[//http://codex.wordpress.org/Function_Reference/Custom_Queries](http://codex.wordpress.org/Function_Reference/Custom_Queries)

```
add_filter( 'pre_get_posts', 'add_films_in_home' );
```

```
function add_films_in_home( $query ) {
```

```
    if ( is_home() && !$query->query_vars['suppress_filters'] ) {  
        $query->set( 'post_type', array( 'post', 'films' ) );  
    }
```

```
    return $query;
```

```
}
```

[//http://codex.wordpress.org/Function_Reference/Custom_Queries](http://codex.wordpress.org/Function_Reference/Custom_Queries)

```
add_filter( 'posts_join', 'my_join' );
add_filter( 'posts_fields', 'my_fields' );

function my_join ( $join ) {
    global $wpdb;

    if( is_single() && get_query_var('post_type') == 'films' ) {
        $join .= " LEFT JOIN my_table ON " .
            $wpdb->posts . ".ID = my_table.ID ";
    }

    return $join;
}

function my_fields( $fields ) {
    global $wpdb, $post;

    if( is_single() && get_query_var('post_type') == 'films' ) {
        $fields .= ", my_table.* ";
    }

    return $fields;
}
```

Transient API

“**WordPress Transients API** offers a simple and standardized way of storing cached data in the database temporarily by giving it a custom name and a timeframe after which it will expire and be deleted.”



[//http://codex.wordpress.org/Function_Reference/Transients_API](http://codex.wordpress.org/Function_Reference/Transients_API)

```
function list_last_film( $limit ) {
    global $wpdb;

    $last_films = get_transient( 'last_films_' . $limit );

    if ( $last_films === false ) {
        $sql_source = "SELECT ID, post_title
                        FROM $wpdb->posts
                        WHERE post_type = 'films' and post_status = 'published'
                        ORDER BY post_date
                        LIMIT %d";

        $sql = $wpdb->prepare($sql_source , $limit );
        $last_films = $wpdb->get_results( $sql );

        set_transient( 'last_films_' . $limit , $last_films , 60*60*2 );
    }

    foreach ( $last_films as $film_item) {
        echo $film_item->post_title;
    }
}
```

APC Object cache

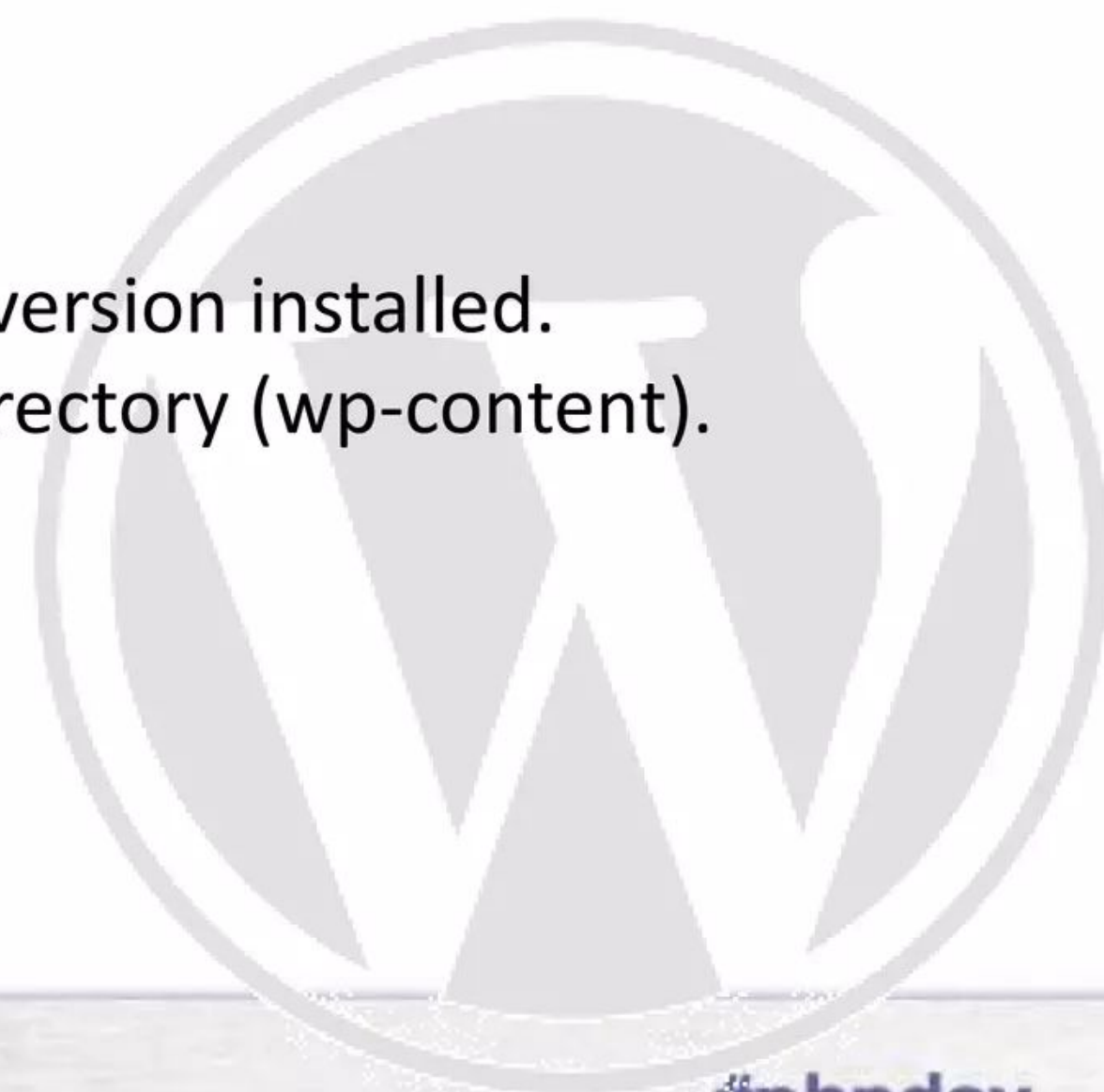
“APC Object Cache provides a persistent memory-based backend for the WordPress object cache.

An object cache is a place for WordPress and WordPress extensions to store the results of complex operations.”

<http://wordpress.org/extend/plugins/apc/>

Installation:

1. Verify that you have PHP 5.2+ and a compatible APC version installed.
2. Copy object-cache.php to your WordPress content directory (wp-content).
3. Done!



Cache template part

“Simple custom function to store manually a template part on filesystem”



```
function cache_template_part( $file , $args = null ) {

    $defaults = array(
        'always' => true,
        'rewrite' => false,
        'cachetime' => 60*60*2,
        'cachepath' => 'wp-content/cache/',
    );

    $args = wp_parse_args( $args, $defaults );
    extract( $args, EXTR_SKIP );

    $cachefile = $cachepath . str_replace( "/", "-", $file);
    $cachefile_created = (@file_exists($cachefile)) ? @filemtime($cachefile) : 0;

    if ( !$rewrite
        && ( time() - $cachetime < $cachefile_created )
        && ( $always || !is_user_logged_in() ) ) {

        echo file_get_contents( $cachefile );

    } else {

        ....

    }

}
```

```
ob_start();
include($file);

if ( $always || !is_user_logged_in() ) {
    $b = ob_get_clean();
    $b = preg_replace('/[ ]+/', ' ', $b);
    $b = str_replace( array("\r\n", "\r", "\n", "\t" ), ' ', $b);

    if ( strlen($b) > 1 ) {
        $signature = "<!-- Cached " . $cachefile
            . " - " . date('jS F Y H:i') . " -->";

        file_put_contents( $cachefile , $signature . $b);
    }

    echo $b;

}

ob_end_flush();
```

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

```
    <div id="container">
```

```
        <div id="content" role="main">
```

```
            <?php
```

```
            /* Run the loop to output the post.
```

```
            * If you want to overload this in a child theme then include a  
file
```

```
            * called loop-single.php and that will be used instead.
```

```
            */
```

```
            get_template_part( 'loop', 'single' );
```

```
            ?>
```

```
        </div><!-- #content -->
```

```
    </div><!-- #container -->
```

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

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

```
<?php cache_template_part('header.php'); ?>
```

```
    <div id="container">
```

```
        <div id="content" role="main">
```

```
            <?php
```

```
            /* Run the loop to output the post.
```

```
            * If you want to overload this in a child theme then include a  
file
```

```
            * called loop-single.php and that will be used instead.
```

```
            */
```

```
            get_template_part( 'loop', 'single' );
```

```
            ?>
```

```
        </div><!-- #content -->
```

```
    </div><!-- #container -->
```

```
<?php cache_template_part('sidebar.php'); ?>
```

```
<?php cache_template_part('footer.php'); ?>
```

```
add_action( 'save_post', 'clear_cache_on_save' );

function clear_cache_on_save( $post_id ) {
    global $post;

    if ( $post->post_status == "publish" ) {

        $cachepath = 'wp-content/cache/';

        $cachefile = array();
        $cachefile[] = $cachepath . "header.php";
        $cachefile[] = $cachepath . "sidebar.php";
        $cachefile[] = $cachepath . "footer.php";

        foreach($cachefile as $file){
            if( file_exists( $file ) ) {
                unlink( $file );
            }
        }

    }
}
```

Debugging tools

- Debug query
<http://wordpress.org/extend/plugins/debug-queries/>
- Debug bar
<http://wordpress.org/extend/plugins/debug-bar/>

Add this line to your wp-config.php

- `define( 'WP_DEBUG', true );`
- `define( 'SAVEQUERIES', true );`



What's coming in WordPress 3.2

1. Requirements Changes:

- PHP version 5.2 or greater (old requirement 4.3 or greater)
- MySQL version 5.0.15 or greater (old requirement 4.1.2 or greater)

2. Twenty Eleven Theme

3. Speed Improvements

For more info:

http://codex.wordpress.org/Version_3.2



Plugins Toolbox

- Custom post type UI

<http://wordpress.org/extend/plugins/custom-post-type-ui/>

- Simple Custom Post Type Archives

<http://wordpress.org/extend/plugins/simple-custom-post-type-archives/>

- Query Multiple Taxonomies

<http://wordpress.org/extend/plugins/query-multiple-taxonomies/>

- Super widgets

<http://wordpress.org/extend/plugins/super-widgets/>

- Widget Logic

<http://wordpress.org/extend/plugins/widget-logic/>



Links Reference

Stats

- http://w3techs.com/technologies/overview/content_management/all
- <http://trends.builtwith.com/blog/WordPress>
- <http://en.wordpress.com/stats/>
- <http://wordpress.org/download/counter/>



Links to follow

- <http://codex.wordpress.org/>
- <http://wpengineer.com/>
- <http://www.wprecipes.com/>
- <http://www.wpbeginner.com/>
- <http://wpshout.com/>



Questions?



Thanks



Pelizzone Maurizio

maurizio@mavida.com

<http://www.mavida.com>

<http://maurizio.mavida.com>

