

# Auto setting post format in my WordPress setup

---

WordPress with it's custom taxonomy engine has the Post Format option, but it would be much better to do this automatically, wouldn't it? If me as a human can tell the which post is what I just need to translate the logic to PHP.

---

WordPress with it's custom taxonomy engine has the Post Format<sup>[1]</sup> option, but it would be much better to do this automatically, wouldn't it?

If me as a human can tell the which post is what I just need to translate it to PHP.

Also: post-type-discovery<sup>[2]</sup> is an interesting idea, and though I like Amy's approach<sup>[3]</sup>, it's not really what I need. Especially that unlike most of the sites, I have two, different types of picture-based posts: Photo ( made by me ) and Image ( from somewhere else ).

I have a constant here, to make things easier: ARTICLE\_MIN\_LENGTH is 1100 right now. All my posts are in Markdown, to the regexes are against Markdown formatting.

- if
  - there is a meta webmention url
  - and a meta webmention type
  - and the webmention type is 'u-in-reply-to'
  - and there is a meta webmention rsvp entry the post is `rsvp`

- elseif
  - either
    - there is a meta webmention url
    - and a meta webmention type
    - and the webmention type is 'u-in-reply-to'
    - and there is no meta webmention rsvp entry
  - or
    - the post is imported from Twitter
    - and is a reply to a tweet the post is a `reply`
- elseif
  - `preg_match "/^`` `(?:[a-z]+)?/m"` is true for the post content, so it has some fenced Markdown code blocks<sup>[4]</sup> the type is `article`
- elseif
  - there is a featured image attached
  - and the featured image has EXIF
    - matching any of my cameras
    - OR my name in the copyright field
  - and the post content is at least 90% similar to the image IPTC description ( which I copy for the post content when I upload a photo ) the type is `photo`
- elseif
  - the content length is below `ARTICLE_MIN_LENGTH`
  - and the post has an attached image it an `image`
- elseif
  - content is a single url the type is `bookmark`
- elseif
  - the content length is below `ARTICLE_MIN_LENGTH`
  - and the post content `preg_match "/(?:www\.)?youtube\.com/watch?v=[a-zA-Z0-9_-]+/"` the type is `video`
- elseif
  - the content length is below `ARTICLE_MIN_LENGTH`
  - and the post content `preg_match "/\[audio.*\]/"` the type is `audio`
- elseif
  - the content length is below `ARTICLE_MIN_LENGTH`
  - and the post content `preg_match "/^> /m"` the type is `quote`
- elseif
  - the post title is empty
  - or
    - content length is below `ARTICLE_MIN_LENGTH`
    - and the post has the category 'blips' the format is `note`
- else it's an `article`

It's not perfect yet, but I'm getting there.

In PHP, including all helper functions ( **this is not a copy-paste code; it will probably break if you do that** ):

```
<?php

define('ARTICLE_MIN_LENGTH', 1100);
define('CACHE_EXPIRE', 300);

function post_format ( &$post ) {
    if ( empty($post) || !is_object($post) || !isset($post->ID) )
        return false;

    if ( $cached = wp_cache_get ( $post->ID, __FUNCTION__ ) )
        return $cached;

    $slug = 'article';
    $name = __('Article', 'petermolnareu');

    $post_length = strlen( $post->post_content );

    $webmention_url = get_post_meta( $post->ID,
'webmention_url', true );
    $webmention_type = get_post_meta( $post->ID,
'webmention_type', true );
    $webmention_rsvp = get_post_meta( $post->ID,
'webmention_rsvp', true );

    $links = extract_urls($post->post_content);
    $content = $post->post_content;

    if (!empty($links) && count($links) == 1) {
        // one single link in the post, so it's most probably
a bookmark
        $webmention_url = $links[0];
        $content = str_replace($webmention_url, '', $content);
        $content = trim($content);
    }

    $is_twitter_reply = is_twitter_reply($post);
```

```

// /m for multiline, so ^ means beginning of line
$has_quote = preg_match("/^> /m", $post->post_content);

// /m for multiline, so ^ means beginning of line
$has_code = preg_match("/^`` `(?:[a-z]+)?/m", $post->post_content);

$difff = 0;
$has_thumbnail = get_post_thumbnail_id( $post->ID );
if ( $has_thumbnail ) {
    $thumbnail_meta = get_extended_meta( $has_thumbnail );
    if (isset($thumbnail_meta['image_meta']['caption'])
    && !empty($thumbnail_meta['image_meta']['caption'])) {
        similar_text( $post->post_content,
        $thumbnail_meta['image_meta']['caption'], $difff);
    }
}

$has_youtube = preg_match("/(?:www\.)?youtube\.com\/watch?v=[a-zA-Z0-9_-]+/", $post->post_content);

$has_audio = preg_match("/\[audio.*\]/", $post->post_content);

$has_blips = has_category( 'blips', $post );

/**
 * Actual discovery
 */
if ( !empty($webmention_url) && !empty($webmention_type)
&& $webmention_type == 'u-in-reply-to' && !
empty($webmention_rsvp) ) {
    $slug = 'rsvp';
    $name = __('Response to event','petermolnareu');
}
elseif ( (!empty($webmention_url) && !
empty($webmention_type) && $webmention_type == 'u-in-reply-
to') || $is_twitter_reply ) {
    $slug = 'reply';

```

```

        $name = __('Reply','petermolnareu');
    }
    elseif ( $has_code ) {
        $slug = 'article';
        $name = __('Article','petermolnareu');
    }
    elseif ( $has_thumbnail &&
static::is_photo($has_thumbnail) && $diff > 90 ) {
        $slug = 'photo';
        $name = __('Photo','petermolnareu');
    }
    elseif ( $post_length < ARTICLE_MIN_LENGTH &&
$has_thumbnail ) {
        $slug = 'image';
        $name = __('Image','petermolnareu');
    }
    elseif ( !empty($webmention_url) && empty($content) ) {
        $slug = 'bookmark';
        $name = __('Bookmark','petermolnareu');
    }
    elseif ( $post_length < ARTICLE_MIN_LENGTH &&
$has_youtube ) {
        $slug = 'video';
        $name = __('Video','petermolnareu');
    }
    elseif ( $post_length < ARTICLE_MIN_LENGTH &&
$has_audio ) {
        $slug = 'audio';
        $name = __('Audio','petermolnareu');
    }
    elseif ( $post_length < ARTICLE_MIN_LENGTH &&
$has_quote ) {
        $slug = 'quote';
        $name = __('Quote','petermolnareu');
    }
    elseif ( strlen($post->post_title) == 0 || ($post_length
< ARTICLE_MIN_LENGTH && $has_blips) ) {
        $slug = 'note';
        $name = __('Note','petermolnareu');
    }
}

```

```

    wp_cache_set ( $post->ID, $slug, __FUNCTION__,
CACHE_EXPIRE );
    return $slug;
}

function is_photo (&$thid) {
    if ( empty($thid) )
        return false;

    if ( $cached = wp_cache_get ( $thid, __FUNCTION__ ) )
        return $cached;

    $return = false;

    $rawmeta = wp_get_attachment_metadata( $thid );

    if ( isset( $rawmeta['image_meta'] ) && !
empty($rawmeta['image_meta'])) {

        if ( isset($rawmeta['image_meta']['copyright']) && !
empty($rawmeta['image_meta']['copyright']) &&
( strstr($rawmeta['image_meta']['copyright'], 'Peter
Molnar') || strstr($rawmeta['image_meta']['copyright'],
'petermolnar.eu')) ) {
            $return = true;
        }

        $my_devs = array ( 'PENTAX K-5 II s', 'NIKON D80' );
        if ( isset($rawmeta['image_meta']['camera']) && !
empty($rawmeta['image_meta']['camera']) &&
in_array(trim($rawmeta['image_meta']['camera']), $my_devs)) {
            $return = true;
        }
    }
}

    wp_cache_set ( $thid, $return, __FUNCTION__,
CACHE_EXPIRE );

    return $return;
}

```

```

function get_extended_meta ( &$thid ) {
    if ( empty ( $thid ) )
        return false;

    if ( $cached = wp_cache_get ( $thid, __FUNCTION__ ) )
        return $cached;

    $attachment = get_post( $thid );

    $meta = array();
    if ( if ( empty($attachment) || !is_object($attachment)
|| !isset($attachment->ID)) {
        $meta = wp_get_attachment_metadata($thid);

        if ( !empty ( $attachment->post_parent ) ) {
            $parent = get_post( $attachment->post_parent );
            $meta['parent'] = $parent->ID;
        }

        $src = wp_get_attachment_image_src ( $thid, 'full' );
        $meta['src'] = $src[0];

        if ( isset($meta['sizes']) && !empty($meta['sizes'])) {
            foreach ( $meta['sizes'] as $size => $data ) {
                $src = wp_get_attachment_image_src ( $thid,
$size);
                $meta['sizes'][$size]['src'] = $src;
            }
        }

        if ( empty($meta['image_meta']['title'])
            $meta['image_meta']['title'] =
esc_attr($attachment->post_title);

        $slug = sanitize_title ( $meta['image_meta']
['title'] , $thid );
        if ( is_numeric( substr( $slug, 0, 1 ) ) )
            $slug = 'img-' . $slug;
        $meta['image_meta']['slug'] = $slug;
    }
}

```

```

        $meta['image_meta']['alt'] = '';
        $salt = get_post_meta($thid,
'_wp_attachment_image_alt', true);
        if ( !empty($salt))
            $meta['image_meta']['alt'] = strip_tags($salt);
    }

    wp_cache_set ( $thid, $meta, __FUNCTION__, CACHE_EXPIRE );

    return $meta;
}

function is_twitter_reply( &$post ) {
    if ( $cached = wp_cache_get ( $post->ID, __FUNCTION__ ) )
        return $cached;

    $r = false;

    $twitter_in_reply_to_screen_name = get_post_meta ( $post->ID, 'twitter_in_reply_to_screen_name', true);
    if (!empty($twitter_in_reply_to_screen_name)) {
        $r = true;
    }

    $twitter_reply_id = get_post_meta ( $post->ID, 'twitter_reply_id', true);
    if (!empty($twitter_reply_id)) {
        $r = true;
    }

    wp_cache_set ( $post->ID, $r, __FUNCTION__,
CACHE_EXPIRE );
    return $r;
}

// this is mainly for Markdown, HTML may fail
function extract_urls( &$text ) {
    $matches = array();
    preg_match_all("/\b(?:http|https)\:\/\/\/?[a-zA-Z0-9\.\\/\?
\:\@\-\_=#]+\.[a-zA-Z0-9\.\\/\?:\@\-\_=#]*/i", $text, $matches);

```



```
$matches = $matches[0];  
return $matches;  
}
```

## Links

1. [https://codex.wordpress.org/Post\\_Formats](https://codex.wordpress.org/Post_Formats)
2. <http://indiewebcamp.com/post-type-discovery>
3. <http://rhiaro.co.uk/2015/09/post-type>
4. <https://help.github.com/articles/github-flavored-markdown/#fenced-code-blocks>

Created by Peter Molnar <[mail@petermolnar.net](mailto:mail@petermolnar.net)>, published at 2015-10-30 09:28 UTC, last modified at 2021-10-31 15:57 UTC , to canonical URL <https://petermolnar.net/article/wordpress-post-format-discovery/> , licensed under CC-BY-4.0 .