

# Advanced Caching Internally and Externally

Anthony Burchell

@thepressguy

Slides Available at: [antpb.com/caching](http://antpb.com/caching)



# Why should I Cache?

Page load speed will keep readers on your site



# Anyone watch Halt and Catch Fire?

Cardiff's goal of building a computer with a system response time of 396 milliseconds can be easily translated to your webpage.

**A quick response will make users addicted to your product (your site).**



# Why is it slow??

WordPress is making sometimes thousands of calls to files. These files are then referring to the database to gather content and render the page.



# How do I speed it up?

There are many many ways to speed things up.  
The most turn key solution would be caching.



# Caching Plugins

WP Supercache  
WP Redis Cache  
W3 Total Cache  
WP Fast Cache



# THEY'RE ALL GREAT

But let's just focus on two...

WP Supercache

WP RedisCache



# WP Supercache

Generates static html files from  
your dynamic WordPress blog.



# Remove Pointless Processing

All those php files add up to a simple html page...why not just save that data and display it quicker.



# WP Supercache - Caching On

WP Super Cache • Trip The Light x

www.trip-the-light.com/wp-admin/options-general.php?page=wpsupercache

Trip The Light 4 2 + New Howdy, tripth

## WP Super Cache Settings

Easy Advanced CDN Contents Preload Plugins Debug

### Caching

☒ Caching On (Recommended)  
☐ Caching Off  
*Note: enables PHP caching, cache rebuild, and mobile support*

Update Status

### Cache Tester

Test your cached website by clicking the test button below.

Test Cache

### Delete Cached Pages

Cached pages are stored on your server as html and PHP files. If you need to delete them use the button below.

Delete Cache

### Recommended Links and Plugins

Caching is only one part of making a website faster. Here are some other plugins that will help:

- [Yahoo! Yslow](#) is an extension for the Firefox add-on Firebug. It analyzes web pages and suggests ways to improve their performance based on a set of rules for high performance web pages. Also try the performance tools online at [GTMetrix](#).

### Need Help?

1. Use the debug system in the Debug tab above. It will tell you what the plugin is doing.
2. [Installation Help](#)
3. [Frequently Asked Questions](#)
4. [Support Forum](#)
5. [Development Version](#)

### Rate This Plugin!

Please [rate](#) this plugin and tell me if it works for you or not. It really helps development.

Dashboard  
Posts  
Media  
Pages  
Comments 2  
AdRotate  
Projects  
Appearance  
Plugins  
Users  
Tools  
Settings  
General  
Writing  
Reading  
Discussion  
Media  
Permalinks  
WP Super Cache  
Redis Cache  
Collapse menu



# WP Supercache - Advanced Settings

## Caching

☒ Cache hits to this website for quick access. *(Recommended)*

☐ Use mod\_rewrite to serve cache files. *(Recommended)*

☒ Use PHP to serve cache files.

☐ Legacy page caching.

*Mod\_rewrite is fastest, PHP is almost as fast and easier to get working, while legacy caching is slower again, but more flexible and also easy to get working. New users should use PHP caching.*



# WP Supercache - Advanced Settings

- ☐ Remove UTF8/blog charset support from .htaccess file. Only necessary if you see odd characters or punctuation looks incorrect. Requires rewrite rules update.
- ☒ Clear all cache files when a post or page is published or updated.
- ☐ Extra homepage checks. (Very occasionally stops homepage caching) *(Recommended)*
- ☐ Only refresh current page when comments made.
- ☐ List the newest cached pages on this page.
- ☐ Coarse file locking. You probably don't need this but it may help if your server is underpowered.  
Warning! *May cause your server to lock up in very rare cases!*
- ☒ Late init. Display cached files after WordPress has loaded. Most useful in legacy mode.

DO NOT CACHE PAGE secret key: [52940127c409d0f8cf2173d624728548](#)



**Cool.**

**But what's the catch?**

If your content is constantly changing  
static files will be displaying old content.



# Two Ways to Avoid This

- 1.) Preload Pages
- 2.) Dynamic Caching



# Option 1 - Preload

## WP Super Cache Settings

[Easy](#)[Advanced](#)[CDN](#)[Contents](#)[Preload](#)[Plugins](#)[Debug](#)

This will cache every published post and page on your site. It will create supercache static files so unknown visitors (including bots) will hit a cached page. This will probably help your Google ranking as they are using speed as a metric when judging websites now.

Preloading creates lots of files however. Caching is done from the newest post to the oldest so please consider only caching the newest if you have lots (10,000+) of posts. This is especially important on shared hosting.

In 'Preload Mode' regular garbage collection will only clean out old legacy files for known users, not the preloaded supercache files. This is a recommended setting when the cache is preloaded.

Refresh preloaded cache files every  minutes. (0 to disable, minimum 30 minutes.)

☐ Preload mode (garbage collection only on legacy cache files. Recommended.)

☐ Preload tags, categories and other taxonomies.

☐ Send me status emails when files are refreshed.

☒ Many emails, 2 emails per 100 posts.

☐ Medium, 1 email per 100 posts.

☐ Less emails, 1 at the start and 1 at the end of preloading all posts.

[Update Settings](#)[Preload Cache Now](#)



# Preload Problems

Content will not update with every refresh  
but rather on a time interval.



# Option 2

# Dynamic Content

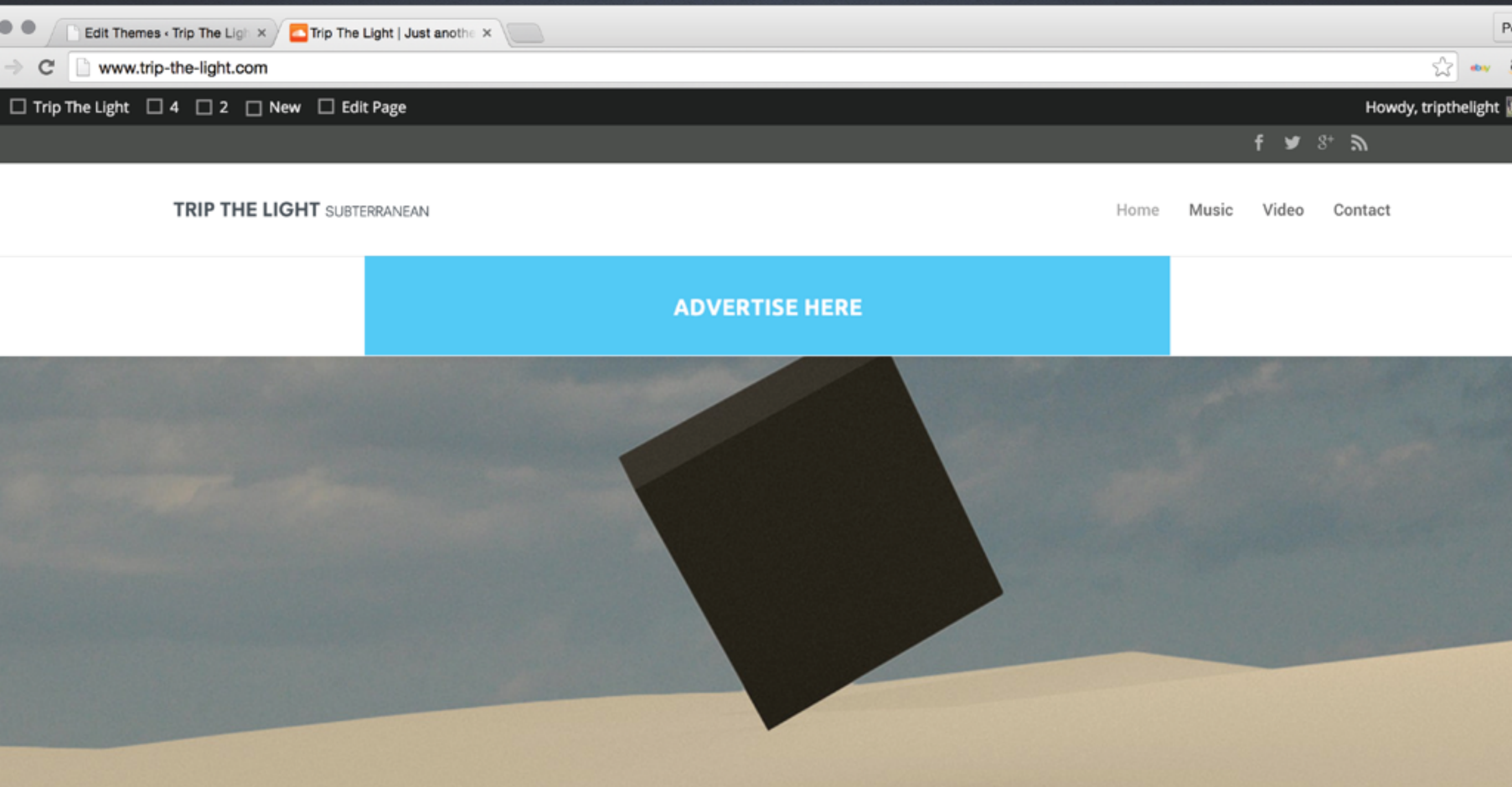
Store content to the output buffer of  
and statically cache only  
parts of the page.



**It's a bit confusing  
but hang in there...**

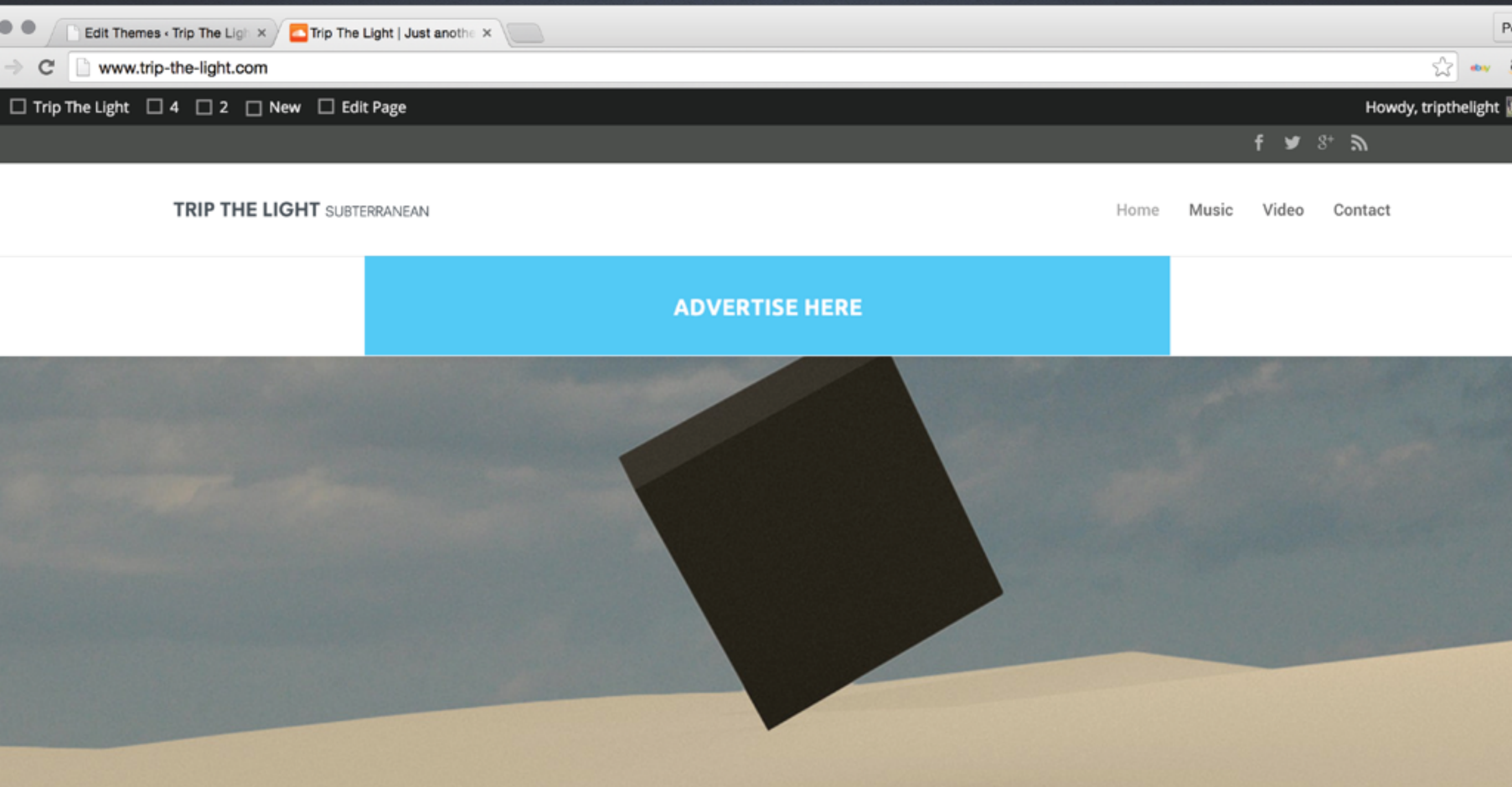


# Let's use the example of AdRotate Pro



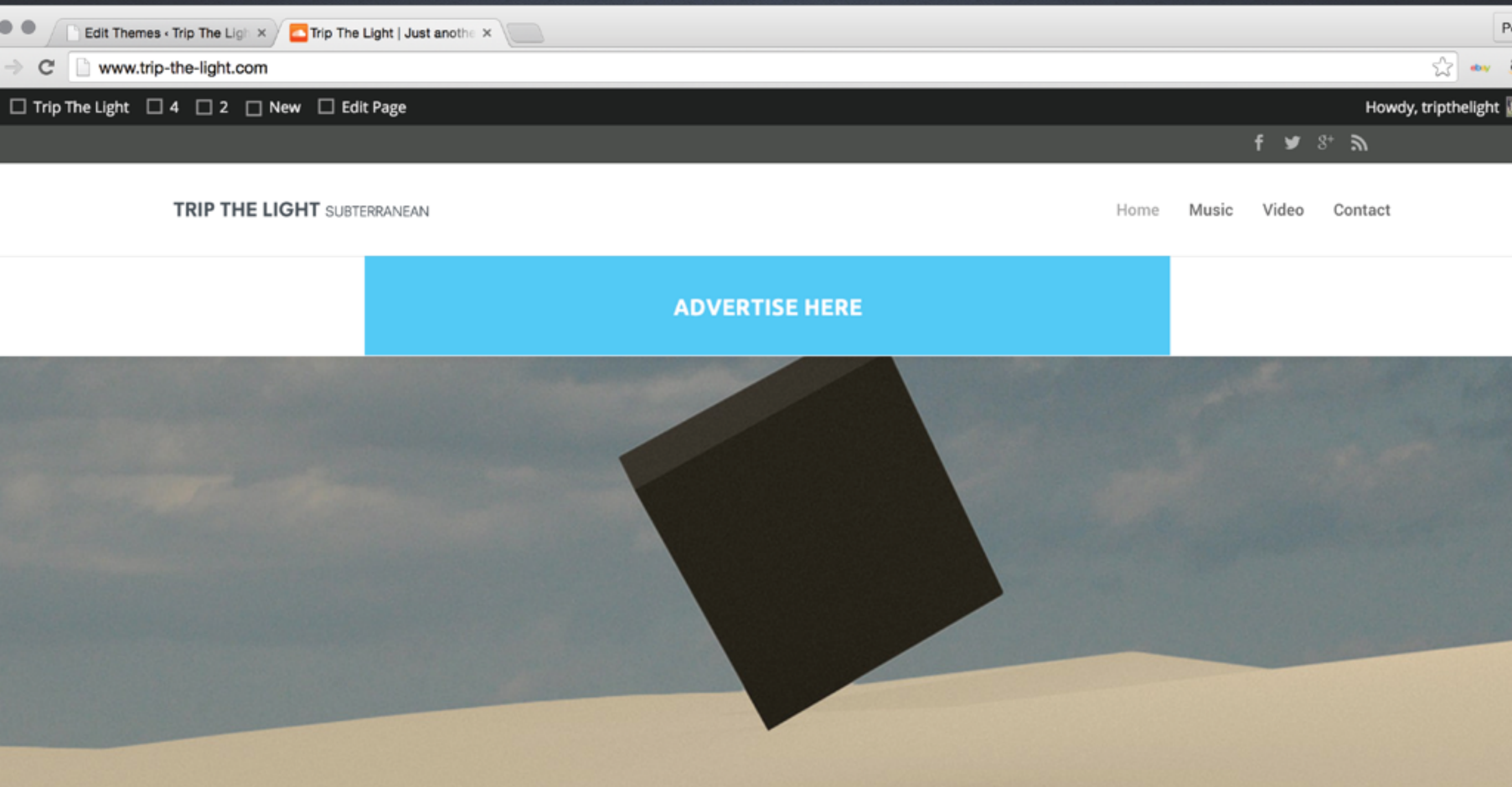


# Refresh the page...





# Refresh again...





# AdRotate Pro

## Required Code

```
<? php adrotate_group(1); ?>
```



# **WP-Supercache has a dynamic caching plugin**

</plugins/wp-super-cache/plugins/dynamic-cache-test.php>



# Let's define the string of text WP-Supercache should look for

```
define( 'DYNAMIC_OUTPUT_BUFFER_TAG', 'dynamicContent' ); // Change this to a secret placeholder tag

if ( DYNAMIC_OUTPUT_BUFFER_TAG != '' ) {
    function dynamic_output_buffer_test( &$cachedata = 0 ) {
        if ( defined( 'DYNAMIC_OB_TEXT' ) )
            return str_replace( DYNAMIC_OUTPUT_BUFFER_TAG, DYNAMIC_OB_TEXT, $cachedata );

        ob_start();
        // call the sidebar function, do something dynamic
        echo "<div style='margin-left:25%; margin-right:25%;>" . adrotate_group(1) . "</div>";
        $text = ob_get_contents();
        ob_end_clean();

        if ( $cachedata === 0 ) { // called directly from the theme so store the output
            define( 'DYNAMIC_OB_TEXT', $text );
        } else // called via the wpsc_cachedata filter. We only get here in cached pages
            return str_replace( DYNAMIC_OUTPUT_BUFFER_TAG, $text, $cachedata );
    }

    add_cacheaction( 'wpsc_cachedata', 'dynamic_output_buffer_test' );

    function dynamic_output_buffer_init() {
        add_action( 'wp_footer', 'dynamic_output_buffer_test' );
    }

    add_cacheaction( 'add_cacheaction', 'dynamic_output_buffer_init' );

    function dynamic_output_buffer_test_safety( $safety ) {
        if ( defined( 'DYNAMIC_OB_TEXT' ) ) // this is set when you call dynamic_output_buffer_test
            return 1; // ready to replace tag with dynamic content.
        else
            return 0;
    }
}
```



# Let's store the adrotate function in the output buffer of the plugin

```
define( 'DYNAMIC_OUTPUT_BUFFER_TAG', 'dynamicContent' ); // Change this to a secret placeholder tag

if ( DYNAMIC_OUTPUT_BUFFER_TAG != '' ) {
    function dynamic_output_buffer_test( &$cachedata = 0 ) {
        if ( defined( 'DYNAMIC_OB_TEXT' ) )
            return str_replace( DYNAMIC_OUTPUT_BUFFER_TAG, DYNAMIC_OB_TEXT, $cachedata );

        ob_start();
        // call the sidebar function, do something dynamic
        echo "<div style='margin-left:25%; margin-right:25%;'>" . adrotate_group(1) . "</div>";
        $text = ob_get_contents();
        ob_end_clean();

        if ( $cachedata === 0 ) { // called directly from the theme so store the output
            define( 'DYNAMIC_OB_TEXT', $text );
        } else // called via the wpsc_cachedata filter. We only get here in cached pages in wp-cache-phase1.php
            return str_replace( DYNAMIC_OUTPUT_BUFFER_TAG, $text, $cachedata );
    }
    add_cacheaction( 'wpsc_cachedata', 'dynamic_output_buffer_test' );

    function dynamic_output_buffer_init() {
        add_action( 'wp_footer', 'dynamic_output_buffer_test' );
    }
    add_cacheaction( 'add_cacheaction', 'dynamic_output_buffer_init' );

    function dynamic_output_buffer_test_safety( $safety ) {
        if ( defined( 'DYNAMIC_OB_TEXT' ) ) // this is set when you call dynamic_output_buffer_test() from the theme
            return $safety;
    }
}
```



**Awesome.**

**Now let's put the string  
into the theme.**



```

<html <?php language_attributes(); ?>
<!--<![endif]-->
<head>
    <meta charset="<?php bloginfo( 'charset' ); ?>" />
    <title><?php elegant_titles(); ?></title>
    <?php elegant_description(); ?>
    <?php elegant_keywords(); ?>
    <?php elegant_canonical(); ?>

    <?php do_action( 'et_head_meta' ); ?>

    <link rel="pingback" href="<?php bloginfo('pingback_url'); ?>" />

    <?php $template_directory_uri = get_template_directory_uri(); ?>
    <!--[if lt IE 9]>
    <script src="<?php echo esc_url( $template_directory_uri . '/js/html5.js' ); ?>" type="text/javascript"></script>
    <![endif]-->

    <script type="text/javascript">
        document.documentElement.className = 'js';
    </script>

    <?php wp_head(); ?>
</head>
<body <?php body_class(); ?>>
    <div id="page-container">

```

```

        <?php
        if ( function_exists( 'dynamic_output_buffer_test' ) )
            dynamic_output_buffer_test();
        ?>
        dynamicContent

```

```

<?php
    if ( is_page_template( 'page-template-blank.php' ) ) {
        return;
    }

    $set_secondary_nav_items = et_divi_get_top_nav_items();

    $set_phone_number = $set_secondary_nav_items->phone_number;

```



```
</script>
        document.documentElement.className = 'js';
</script>
```

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

```
</head>
```

```
<body <?php body_class(); ?>>
    <div id="page-container">
```

```
    <?php
    if ( function_exists( 'dynamic_output_buffer_test' ) )
    dynamic_output_buffer_test();
    ?>
    dynamicContent
```

```
<?php
    if ( is_page_template( 'page-template-blank.php' ) ) {
        return;
    }

    $et_secondary_nav_items = et_divi_get_top_nav_items();
```



**What if I don't need  
dynamic caching?  
MAKE IT FASTER!**





# WPREDIS

WORDPRESS CACHING MADE EASY



# What is Redis?

Redis is an advanced key-value  
cache and store

i.e. It's freakin quick.



**I won't get into the  
technical details today.  
But here is a step by step  
video tutorial I did  
configuring.**

[REDIS TUTORIAL](#)



# Benefits

Can serve pages from external  
or internal server  
as fast as .5 second

**FULL RENDER**



**Saves a hash key value  
and ties it to the html data  
from the page**



**Also minifies css to further  
capitalize on the speed  
Redis handles static content**



# Redis Cache

## Details

Enable

## Settings

Redis Server

localhost

Redis Instance

0

Redis Port

6379

Redis Password

password

Cache Expiration (in seconds)

3600

Save Settings

## Minify CSS

Disable

## Flush Cache

Clear Cache



**Q&A**



# Advanced Caching Internally and Externally

Anthony Burchell

@thepressguy

Slides Available at: [antpb.com/caching](http://antpb.com/caching)