

Adaptive, responsive, pure CSS3, bandwith-saving image gallery for WordPress

Image gallery for any device, for modern browsers, with nearly pure CSS3.

For a long while I've been using Galleriffic from Trent Foley^[^1] as my image gallery on my site; I've even created a plugin for WordPress to be used with^[^2].

As time passed by I realized that even I love this gallery it's not fulfilling my requirements of adaptive and responsive layout anymore. The past 3 years has resulted resolutions I've never seen before (800x480 ?!), revised some that has fallen into oblivion (800x600, 1024x768) and brought ppi-s never seem before (fullHD on less, than 5").

It's clear that lot's of thing needs to be changed. First of all: forget Photoshop. You just simply cannot create static layouts anymore. There's a lot of argument against responsive layouts, mostly because they are pretty hard to implement. As if CSS hack a few years ago were not... It's clearly the only way to go with at this point.

The main issue: the size of an image

The issue with image galleries is that *image sizes* due to their *limited* data are usually fixed. But as always, there's a good way to come around this, even if it has its price: **you need to use the image as background.**

Don't freak out: I know using background in CSS results the pre-download of all images - therefore preloading ALL sizes of ALL images -, but this will be avoided: the browsers skip the media queries that could not be applied for the current layout! Meaning only the default and the ones depending on the met conditions will need to load; all the others will be skipped.

The other issue is, that there's a limit you can stretch and image, otherwise it's going to look ugly. To avoid this limit, you need to have more sizes of the same image and replace the original when certain conditions are matched.

So with the help of CSS media queries^[^3] we'll define reasonable intervals for a specific size of an image and put the conditions into media queries.

Tricks to be used

- `:target` - I'll utilize the `:target` trick^[4] and CSS3 transitions to switch between the images. I'll gain two things: I'll not have images in the links (therefore the gallery is compatible with auto-lightbox plugins) and it'll be a lot faster thanks for the native mode.
- `padding-bottom: 50%; width: 50%;` - I'll also use a padding hack for responsive square sizes^[5] for the thumbnails.

The code

PHP

adaptgal.php

```
<?php

global adaptgal_image_sizes;
define ( 'THUMB_PREFIX', 'thumb' );
define ( 'STD_PREFIX', 'src' );
$adaptgal_image_sizes = array (
    460 => array (
        THUMB_PREFIX => 60,
        STD_PREFIX => 640,
    ),
    720 => array (
        THUMB_PREFIX =>120,
        STD_PREFIX => 1024,
    ),
    1600 => array (
        THUMB_PREFIX => 180,
        STD_PREFIX => 1200,
    )
);

add_action( 'init', 'adaptgal_init');

function adaptgal_init() {
    foreach ( $adaptgal_image_sizes as $resolution =>
    $sizes ) {
        add_image_size(
            THUMB_PREFIX . $resolution,
    $sizes[ THUMB_PREFIX ],
            $sizes[ THUMB_PREFIX ],
            true
        );
        add_image_size(
```

php

```

        STD_PREFIX . $resolution, $sizes[ STD_PREFIX ],
        $sizes[ STD_PREFIX ],
        false
    );
}
add_shortcode('adaptgal', 'adaptgal' );
}

function adaptgal( $atts , $content = null ) {
    global $post;
    global $adaptgal_image_sizes;
    $galtype = 'adaptgal';

    /* get image type attachments for the post by ID */
    $attachments = get_children( array (
        'post_parent'=>$post->ID,
        'post_type'=>'attachment',
        'post_mime_type'=>'image',
        'orderby'=>'menu_order',
        'order'=>'asc'
    ));

    if ( empty($attachments) ) {
        return false;
    }

    foreach ( $attachments as $aid => $attachment ) {
        $img = array();
        $_post = get_post($aid);

        /* set the titles and alternate texts */
        $img['title'] = strip_tags ( attribute_escape($_post->post_title) );
        $img['alttext'] = strip_tags ( get_post_meta($_post->id, '_wp_attachment_image_alt', true) );
        $img['caption'] = strip_tags
( attribute_escape($_post->post_excerpt) );
        $img['description'] = strip_tags
( attribute_escape($_post->post_content) );

        $std = wp_get_attachment_image_src( $aid, 'medium' );

```

```

        $thumbid = $galtype . '-' . THUMB_PREFIX . $aid;
        $previewid = $galtype . '-' . STD_PREFIX . $aid;
        if (!empty($img['description'])) $description =
'<span class="thumb-description">'. $img['description'] .'</
span>';

        $elements[ THUMB_PREFIX ][ $aid ] = '
<li>
        '. $img['title'] .'#'. $previewid .'
</li>';

        $elements[ STD_PREFIX ][ $aid ] = '
<figure id="'. $previewid .' ">
        
        <figcaption>'. $img['caption'] . $description .'<
/figcaption>
</figure>';

        foreach ( $adaptgal_image_sizes as $resolution =>
$size ) {
            $thumbnail = wp_get_attachment_image_src( $aid,
THUMB_PREFIX . $resolution );
            /* if there's no thumbnail in this size,
            * the fallback is the full-size image;
            * so we need to fall back to the default
thumbnail instead
            */
            if ( $thumbnail[3] != true ) {
                $thumbnail =
wp_get_attachment_image_src( $aid, 'thumbnail' );
            }
            $preview = wp_get_attachment_image_src( $aid,
STD_PREFIX . $resolution );
            $bgimages[ THUMB_PREFIX ][ $resolution ][ $aid ]
= '#'. $galtype . '-' . THUMB_PREFIX . $aid .' { background-
image: url('. $thumbnail[0] .'); }';
            $bgimages[ STD_PREFIX ][ $resolution ][ $aid ] =
'#'. $galtype . '-' . STD_PREFIX . $aid .' { background-
image: url('. $preview[0] .'); }';
        }

```

```

    }
}

$cntr = 0;
$resolutions = array_keys( $adaptgal_image_sizes );
foreach ( $bgimages[ THUMB_PREFIX ] as $resolution =>
$backgrounds ) {
    $seq = "\n" . join( "\n", $bgimages[ THUMB_PREFIX ]
[ $resolution ] ) . "\n" . join( "\n", $bgimages[ STD_PREFIX ]
[ $resolution ] );

    if ( $cntr == 0 ) {
        $mediaqueries .= $seq;
    }
    elseif ( $cntr != ( sizeof
( $bgimages[ THUMB_PREFIX ] ) -1 ) ) {
        $mediaqueries .= '
        @media ( min-width : '. $resolution .'px ) and
( max-width : '. ( $resolutions[ $cntr + 1 ] - 1 ) .'px ) {
            '. $seq .'
        }';
    }
    else {
        $mediaqueries .= '
        @media ( min-width : '. $resolution .'px ) {
            '. $seq .'
        }';
    }
    $cntr++;
}

$output = '
<style>'. $mediaqueries .'

```

```
        </nav>
    </section>
';

    return $output;
}
```

CSS

adaptgal.css

CSS

```
.adaptgal-images {
    position: relative;
}

.adaptgal-thumbs {
    display: block;
    position: relative;
    vertical-align: top;
    overflow: hidden;
    width: 16%;
    margin-left: 84%;
}

.adaptgal-previews {
    position: absolute;
    display: block;
    width: 82%;
    height: auto;
    bottom: 0;
    top: 0;
    left: 0;
    right: 0;
    z-index: 2;
    vertical-align: top;
    overflow: hidden;
    border-radius: 0.5em;
}

@media ( orientation: landscape ) and ( min-width :
```



```

600px ) {
    .adaptgal-thumbs {
        width: 15%;
        margin-left: 85%;
    }

    .adaptgal-previews {
        width:66%;
    }
}

@media ( orientation:landscape ) and ( min-width : 960px )
{
    .adaptgal-thumbs {
        width: 10%;
        margin-left: 90%;
    }
    .adaptgal-previews {
        width:49%;
    }
}

.adaptgal-previews figure {
    position:absolute;
    display: block;
    width:auto;
    height:auto;
    top:0.2em;
    left: 0.2em;
    right: 0.2em;
    bottom: 0.2em;
    z-index:2;
    vertical-align: top;
    overflow:hidden;
    text-indent: -9999px;
    opacity: 0;
    -webkit-transition: opacity 0.3s ease-in-out;
    transition: opacity 0.3s ease-in-out;
    background-size: contain;
    background-position: center center;
}

```

```
background-repeat: no-repeat;
z-index: 1;
}

.adaptgal-previews figure:target {
  opacity: 0.99;
  z-index: 10;
}

.adaptgal-thumbs ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
}

.adaptgal-thumbs ul li {
  float: left;
  position: relative;
  margin: 0.2%;
  width: 49%;
  padding-bottom: 49%;
}

.adaptgal-thumbs li a:link,
.adaptgal-thumbs li a:visited {
  position: absolute;
  left: 0.2em;
  right: 0.2em;
  top: 0.2em;
  bottom: 0.2em;
  overflow: hidden;
  border: 1px solid #666;
  box-shadow: 0 0 4px #111;
  overflow: hidden;
  opacity: 0.5;
  transition: opacity 0.2s ease-in-out;
  -webkit-transition: opacity 0.2s ease-in-out;
  background-size: contain;
  background-position: center center;
  background-repeat: no-repeat;
  text-indent: -999px;
```

```
}  
  
.adaptgal-thumbs li a:hover,  
.adaptgal-thumbs li a:active,  
.adaptgal-thumbs li a:focus,  
.adaptgal-thumbs li a.adaptgal-active {  
    opacity: 0.99;  
}  
}
```

Usage

Insert the `[adaptgal]` shortcode (without the withespaces) to display the attached images for the post.

Links

1. <http://www.twospy.com/galleriffic/>
2. <http://wordpress.org/plugins/wp-galleriffic/>
3. <http://mediaqueri.es/>
4. http://www.w3schools.com/cssref/tryit.asp?filename=trycss3_target
5. <http://tympanus.net/Blueprints/ResponsiveFullWidthGrid/>

Created by Peter Molnar <mail@petermolnar.net>, published at 2013-05-22 16:19 UTC, last modified at 2021-10-31 15:57 UTC , to canonical URL <https://petermolnar.net/article/adaptive-responsive-css3-gallery/> , licensed under CC-BY-4.0 .