Responsive Images

- Some History
- How it works
  - Browser side
- New functions and hooks
- A new default image size
- Customizing responsive image markup
- Sources

Since WordPress 4.4, native responsive images is supported by including `srcset` and `sizes` attributes to the image markup it generates. For background on this feature, read the merge proposal.

Some History

When users upload images in WordPress, it automatically crops new images to smaller sizes. For example, if you upload an image that's 1500 x 706, the image sizes might look like this:

- Full Size – 1500 x 706
- Large – 500 x 235
- Medium – 300 x 141
- Thumbnail – 150 x 150

So WordPress automatically creates several sizes of each image uploaded to the media library. Additional sizes are created depending on the theme. If the full size image is attached to a post, users on desktop and mobile devices will see the full size image. However, it doesn't make sense to use the full size image on mobile devices because of its display and file size.

Before responsive design was popular, many sites attempted to dynamically serve different layouts (including images) to browsers based on the device type (e.g. phone, tablet, etc.). In these cases, all of the dynamic stuff happened at the server, before the page was rendered. This strategy is usually associated with the term **adaptive design**.

Responsive design, on the other hand, uses tools like media queries to allow a single page to be rendered that will *respond* in the browser based on things like viewport width and display density.

Responsive images follows the second strategy and sends all of the information to the browser up front and lets the browser take care of loading the appropriate image rather than making those decisions on the server before the page is loaded.

How it works

By including the available sizes of an image into a `srcset` attribute, it allows the software to automatically use and display the right image based on a device's screen size. If you attach a full size 1500 x 706 image to a post in WordPress, mobile devices will see the large or medium-sized image instead—potentially saving bandwidth and speeding up page load times in the process.

Note that for compatibility with existing markup, neither `srcset` nor `sizes` are added or modified if they already exist in content HTML. Responsive images don't have any settings to configure as the magic happens behind the scenes.

Browser side

To help browsers select the best image from the source set list, WordPress also include a default `sizes` attribute that is equivalent to `(max-width: {{image-width}}px) 100vw, {{image-width}}px` . While this default will work out of the box for a majority of sites, themes should customize the default `sizes` attribute as needed using the `wp_calculate_image_sizes` filter.

A normal browser request goes to server, server sends back response. This response includes links to other resources – fonts, css, JS, and images. The browser notices these resources, and sends additional requests to the server and fetches those resources.

What this responsive image approach does is provide additional attributes to the image tag that alerts the browser to the different image files available for that particular image tag so that the browser can then intelligently request the right image file (source) for whatever window/viewport size or even resolution support the browser has. This means the browser can request the right "sized" image file for an image instead of being served an overly large image and resizing down to fit the space after the fact.

For a full overview of how `srcset` and `sizes` works, read *Responsive Images in Practice*, by *Eric Portis* over at A List Apart.

New functions and hooks

To implement this feature, the following new functions were added to WordPress:

- `wp_get_attachment_image_srcset()` – Retrieves the value for an image attachment's `srcset` attribute.
- `wp_calculate_image_srcset()` – A helper function to calculate the image sources to include in a `srcset` attribute.
- `wp_get_attachment_image_sizes()` – Creates a `sizes` attribute value for an image.
- `wp_calculate_image_sizes()` – A helper function to create the `sizes` attribute for an image.
- `wp_image_add_srcset_and_sizes()` – Adds `srcset` and `sizes` attributes to an existing `img` element.

As a safeguard against adding very large images to `srcset` attributes, a `max_srcset_image_width` filter has been added, which allows themes to set a maximum image width for images include in source set lists. The default value is *2048px*.

A new default image size

A new default intermediate size, `medium_large` has been added to better take advantage of responsive image support. The new size is 768px wide by default, with no height limit, and can be used like any other size available in WordPress. As it is a standard size, it will only be generated when new images are uploaded or sizes are regenerated with third party plugins.

The `medium_large` size is not included in the UI when selecting an image to insert in posts, nor are we including UI to change the image size from the media settings page. However, developers can modify the width of this new size using the `update_option()` function, similar to any other default image size.

Customizing responsive image markup

To modify the default `srcset` and `sizes` attributes, you should use the `wp_calculate_image_srcset` and `wp_calculate_image_sizes` filters, respectively.

Overriding the `srcset` or `sizes` attributes for images not embedded in post content (e.g. post thumbnails, galleries, etc.), can be accomplished using the `wp_get_attachment_image_attributes` filter, similar to how other image attributes are modified.

Additionally, you can create your own custom markup patterns by using `wp_get_attachment_image_srcset()` directly in your templates. Here is an example of how you

could use this function to build an `<img>` element with a custom `sizes` attribute:

```
<?php
$img_src = wp_get_attachment_image_url( $attachment_id, 'medium' );
$img_srcset = wp_get_attachment_image_srcset( $attachment_id, 'medium' );
?>
<img src="<?php echo esc_url( $img_src ); ?>"
    srcset="<?php echo esc_attr( $img_srcset ); ?>"
    sizes="(max-width: 50em) 87vw, 680px" alt="Foo Bar">
```

Need more developer details?

Learn more about customizing responsive images markup on this GitHub repository.

Sources

- https://make.wordpress.org/core/2015/11/10/responsive-images-in-wordpress-4-4/
- https://wptavern.com/joe-mcgill-explains-how-responsive-images-work-in-wordpress-4-4

□□□ #2
□ □□□ □□□ 2026-10-31 18:07:15 UTC
□ □□□ □□□ 2026-10-31 18:07:17 UTC