Responsive Images: srcset

Overview

We have worked with basic image optimization using image types, sizing, and quality level adjustments. We also want to be aware of the image's appearance in different screen sizes. The work of responsive image design is to support smaller page weights and to deliver content based upon screen size. The objective is to reduce page weight while being conscious of image clarity and functionality in each screen size and subsequent viewports (the visible area of a web page).

> "Serving desktop-sized images to mobile devices can use 2-4x more data than needed. Instead of a "one-size-fits-all" approach to images, serve different image sizes to different devices. How many image versions should you create? There is no single 'correct' answer to this question. However, it's common to serve 3-5 different sizes of an image. Serving more image sizes is better for performance, but will take up more space on your servers and require writing a tiny bit more HTML." - Web.dev

Prepare

The user experience (**UX**) concept is that users expect to scroll web pages vertically with no need to scroll horizontally to view content. Nor should the user should be required to zoom in or out to see the page content. We cannot expect nor rely upon on a particular viewport width in order for the page to render properly.

This activity focuses on applying the srcset attribute to images using the picture and source elements. The browser parses the responsive image syntax and decides which image is the best match.

The activity content **assumes** the following:

✓ The **viewport meta element** has been properly applied to the document.

✓ The images to be used have been **optimized** for the web by

- Limiting the images' **width and height** (dimensions) to no more than is necessary for the page design in any view size.
- The images are of a valid web **type/format**.
- The images are **compressed** to reduce their overall file size. In many image editing software packages this is referred to as reducing the 'quality'.

✓ The developer has designed and saved images for use on different screen sizes.

> **Art Direction** is the cropping and dimensional sizing of images to focus the user's attention on a specific part of the image to meet the design objectives of the page in different view sizes.
>
> Read: Art Direction - MDN

Responsive images are images that scale nicely to fit any browser size. The srcset attribute allows the browser to choose the best image to display based on the device's screen size and resolution. The srcset attribute is used with the picture and source elements.

Srcset supports the need to target different browser ranges. A different image can be rendered depending up on the width of the screen. Here is an example:

```
<picture>
  <source srcset="images/hero-large.webp" media="min-width: 1000px">
  <source srcset="images/hero-medium.webp" media="min-width: 550px">
  <img src="images/hero-small.webp" alt="Orem Utah Temple" width="400" height="
</picture>
```

> The **picture** tag has a reference to three different images, two sources and a default img tag. Which image is rendered by the browser depends upon the media queries. The following is a description of the rendering process:
>
> Is the width of the browser at least a minimum width of 1000px? If so, the srcset path replaces the src from the img tag and the **`hero-large.webp`** image is displayed. If the browser view width is less than 1000px then the browser tests the next source given.
>
> Is the width of the browser at least a minimum width of 550px? If so, the **`hero-medium.webp`** image is used.

> The default is the img tag src reference to `hero-small.webp`.

Activity Instructions

Step 1: File and Folder Setup

1. If it does not already exist, create a folder named "**week03**" in the wdd131 directory.
2. Add a new file named "**responsive-images.html**" to that week03 folder.
3. Provide a basic HTML structure to the responsive-images.html file.
4. If it does not already exist, add a "**styles**" folder to the week03 folder and add a new CSS file named "**responsive-images.css**" to that styles folder.
5. If it does not already exist, create a folder named "**images**" in your week03 directory.

Step 2: Image Editing

1. Navigate to Temple List and click on a temple link of your choice to gain access to that temple's **Media Gallery**.
2. In the Media Gallery of the temple, click on one of the images and download a **Large** version of the image by using **Save Image As...** or equivalent method.
3. Using your image editing, like PixIr E, load the downloaded image file and create three (3) different images:
   - ✓ Each image should have a different crop (Art Direction)
   - ✓ Save all three images in the WebP file format with a quality around 50%.
   - ✓ Create a small version named "**hero-small.webp**" that is 500px in width by 250px in height.
   - ✓ Create a medium version named "**hero-medium.webp**" that is 1000px in width by 500px in height.
   - ✓ Create a large version named "**hero-large.webp**" that is 1500px in width by 750px in height.

   > Note that the **ratios** of all these images are 2:1 for consistency.

4. Put all of these images in your images folder for week03.

Step 3: HTML: Picture Element

1. In your html file, add a `main` element with a generic division element with the class of "**hero**
2. Within the `div` element, add a `picture` element.
3. Within the `picture` element, add a `source` element with attributes for `media` and `srcset`

> Emmet Shorthand (enter syntax + tab):
>
> ```
> main>div.hero>picture>source:media*2+img
> ```

4. Set the path for the srcset to the largest image.
5. Set the media attribute to `min-width: 1000px` which is the width of the large hero image created earlier.
6. If needed, copy and paste the source line of markup and then change the image to the medium image and the min-width to 500px.

> Neither of these images will render on a page. An `img` element with a `src` is needed.

7. Add the `img` element with a src referencing the smallest image.
8. Set the `alt` attribute value to a description of the image.
9. Set the `width` and `height` attributes to the width and height of the largest image. What should the values for width and height be since we have a 500px by 250px photo and a 1000px by 500px photo and a 1500px by 750px photo? The important point is to maintain the aspect ratio of the image as the ratio is used by the browser to render the space required by the image once it loads. We can use the large image of 1500 and 750 or any equivalent ratio.

▶ Example

Why are img width and height necessary?

Some of the most common causes of a **layout shift** or reflow during the rendering of a page are images, ads, embeds, and iframes without dimensions. "Layout shifts can be distracting to users. Imagine you've started reading an article when all of a sudden elements shift around the page, throwing you off and requiring you to find your place again. This is very common on the web, including when reading the news, or trying to click those 'Search' or 'Add to Cart' buttons. Such

experiences are visually jarring and frustrating. They are often caused when visible elements are forced to move because another element was suddenly added to the page or resized."

Optimize Cumulative Layout Shift - Osmani and Pollard, web.dev

Step 4: CSS:

- In the style sheet, enable responsive behavior on the image by setting the **`width`** of the img element to 100% and the **`height`** to **`auto`** (or vice-versa). This is the easiest way to maintain the image aspect ratio.

```
.hero img {
  width: 100%;
  height: auto;
}
```

- Or to use object-fit with set image containers. Just be mindful of the art direction with the responsive results.

```
.hero img {
  width: 100%;
  height: 100%;
  object-fit: cover;
}
```

Here is an example of this error along with a solution using CSS object-fit on CodePen: ⚙ object-fit

- Another solution is to use the aspect-ratio property.

We want responsive images to maintain their intrinsic **aspect ratio**. Alternating the aspect ratio by setting the width AND height to values (including setting the height to relative measure 100%) in order to fill your design space may distort and pixelate the image. Lighthouse will report on major aspect ratio violations.

More information about aspect ratios.

Optional Resources

- Working with Responsive Images | *(6:49 mins, Responsive Images Transcript)*

- Serve Responsive Images - Katie Hempenius - dev.to
- Aspect-Ratio CSS Property - Una Kravets - web.dev