



Aspect Ratios in CSS

One thing that the latest web vitals report worries about is cumulative layout shift or CLS (See the web.dev entry for [CLS](#)).

Quoting the web.dev article:

Unexpected movement of page content usually happens because resources are loaded asynchronously or DOM elements get dynamically added to the page above existing content. **The culprit might be an image or video with unknown dimensions**, a font that renders larger or smaller than its fallback, or a third-party ad or widget that dynamically resizes itself.

In this article we'll only worry about adding dimensions to images and other replaced content such as video and iframes that embed content on the page.

The way we used to do it

In the beginning, before CSS we would specify image dimensions directly in the `img` tag along with additional information for the browser to format the image. Remember that this is before CSS was introduced.

```

```

For a while this was the only way to add dimensions to the image but since we didn't worry about performance or content shift, it was all good.

Responsive Web Design and Fluid Images

When we started looking at Responsive Web Design and fluid images we changed.

Because the dimensions of the image would change based on the device the user access the content through, using fixed dimensions was not good anymore.

To make sure that fluid images worked in as many devices as possible we relied on CSS and relative measurements.

```
.my-image {  
  width: 100%;  
  height: auto;  
}
```

This CSS would ensure that the image took 100% of its parent container and the height is automatically calculated to keep the aspect ratio of the original image.

Intrinsic dimensions

Even if the image has no dimensions stated, there are still *intrinsic dimensions*, height, width and aspect ratio.

As defined in the [CSS Images Module Level 3](#) overview of sizing terms:

intrinsic dimensions

The term *intrinsic dimensions* refers to the set of the *intrinsic height*, *intrinsic width*, and *intrinsic aspect ratio* (the ratio between the width and height), each of which may or may not exist for a given object. These *intrinsic dimensions* represent a preferred or natural size of the object itself; that is, they are not a function of the context in which the object is used. CSS does not define how the *intrinsic dimensions* are found in general.

Raster images are an example of an object with all three *intrinsic dimensions* (height, width and aspect ratio). SVG images designed to scale might have only an *intrinsic aspect ratio*; SVG images can also be created with only an *intrinsic width* or height. CSS gradients, defined in this specification, are an example of an object with no *intrinsic dimensions* at all. Another example of this is embedded documents, such as the `iframe` element in HTML.

...

If an object (such as an icon) has multiple sizes, then the largest size (by area) is taken as its *intrinsic size*. If it has multiple aspect ratios at that size, or has

multiple aspect ratios and no size, then the aspect ratio closest to the aspect ratio of the default object size is used. Determine this by seeing which aspect ratio produces the largest area when fitting it within the default object size using a [contain constraint](#) fit; if multiple sizes tie for the largest area, the widest size is chosen as its intrinsic size.

So browsers will use intrinsic width and height to create the box for the the image with the right dimensions and aspect ratio.

Aspect-Ratio Rule