



# Aspect Ratio in CSS

Aspect ratio is the ration between an item's width to height.

In HTML some elements have an aspect ratio as part of the Browser's built-in stylesheet so we're OK. That is why you can do this in CSS:

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

Or

```
img {  
  width: 50vw;  
}
```

## Uses

There are several cases where keeping the aspect ratio of an item is important. Some of these examples (taken from [New aspect-ratio CSS property supported in Chromium, Safari Technology Preview, and Firefox Nightly](#)):

- Creating responsive iframes, where they are 100% of a parent's width, and the height should remain a specific viewport ratio
- Creating intrinsic placeholder containers for images, videos, and embeds to prevent re-layout when the items load and take up space
- Creating uniform, responsive space for interactive data visualizations or SVG animations
- Creating uniform, responsive space for multi-element components such as cards or calendar dates
- Creating uniform, responsive space for multiple images of varying dimension (can be used alongside object-fit)

All these cases involve keeping the same aspect ratio for an item when inside a parent container.

# The aspect-ratio CSS property

The aspect-ratio CSS property sets a preferred aspect ratio for the box, which will be used in the calculation of auto sizes and some other layout functions.

The example below, taken From Una Kravetz [codepen](#) shows how this works with a div element.

We use this HTML code:

```
<div>I am always 16x9</div>
```

And this CSS:

```
div {  
  background: lightblue;  
  width: 50vw;  
  
  /*  
    New aspect-ratio property  
  */  
  aspect-ratio: 16 / 9;  
}
```

The browser will calculate the height of the element based on the explicit dimensions and the aspect ratio you indicate in your CSS regardless of the size of the window.

## Fallbacks

For browsers that don't support the property yet, you can use a fallback like the following code.

```
/* Fallback (current, using padding hack) */  
@supports not (aspect-ratio: 16 / 9) {  
  div::before {
```

```
float: left;
padding-top: 56.25%;
content: '';
}

'div::after {
display: block;
content: '';
clear: both;
}
}
```

This uses a vertical padding hack first described in [Creating Intrinsic Ratios for Video](#) and shown working with images in [Aspect Ratios in CSS are a Hack](#). It is not the most elegant solution but it works in older browsers.

Something else to be aware, even in the browsers that support aspect ratio in CSS: Safari (the current technical preview), Chrome Canary (version 89 as of this writing) and Firefox Nightly (version 85), work on the feature may still be ongoing. If you're interested check these issues for current status:

- **Chromium:** <https://bugs.chromium.org/p/chromium/issues/detail?id=1045668>
- **Firefox:** [https://bugzilla.mozilla.org/show\\_bug.cgi?id=1528375](https://bugzilla.mozilla.org/show_bug.cgi?id=1528375)
- **Webkit (Safari):** [https://bugs.webkit.org/show\\_bug.cgi?id=47738](https://bugs.webkit.org/show_bug.cgi?id=47738)