



Cropping Images in CSS With object-fit

I've always struggled to get images to fit where I need them to without giving them explicit size in pixels that will only work on one screen size. Percentages are better but I've always struggled figure out what is the right percentage and whether it'll look ok in smaller form factors.

With the `object-fit` property (and its partner `object-position`) we get very granular control on how an element responds to the size of its content box. Used by itself, `object-fit` allows for image cropping and how it alters its size inside the containing box.

`object-fit` can take one of 5 values:

- **fill**: this is the default value which stretches the image to fit the content box, regardless of its aspect-ratio.
- **contain**: increases or decreases the size of the image to fill the box whilst preserving its aspect-ratio.
- **cover**: the image will fill the height and width of its box, once again maintaining its aspect ratio but often cropping the image in the process.
- **none**: image will ignore the height and width of the parent and retain its original size.
- **scale-down**: the image will compare the difference between none and contain in order to find the smallest concrete object size.

The value that I use most often is `contain` to make sure that the image will fit its container. This is most useful in figures and similar containers where I do something like this:

```
figure {  
  width: 100%;  
}  
  
figure img {  
  object-fit: contain;  
}
```

```
figure figcaption {  
  width:max-width;  
}
```

The idea here is that the figure element will take 100% of the width of its parent. You can also use `object-position` to move the image inside its container.

More information

- [CSS Tricks Almanac](#)
- [MDN](#)
- [Aligator.io](#)
- [Sitepoint](#)