How to use the CSS clip and mask properties

In this overview of the CSS clip and mask properties, Ryan Boudreaux shows you how to use them with examples of code and basic syntax.

By Ryan Boudreaux | in Web Designer, April 16, 2013, 2:49 AM PST

(https://tr1.cbsistatic.com/hub/i/2015/05/07/5df67b2d-f4ae-11e4-940f-14feb5cc3d2a/04162013figure-a.gif)The commonly used clipping and masking actions that graphic designers have used for years in photo and imaging editing software applications, which were applied to PNG, PSD, or SVG files, can also be applied using pure CSS, and this post will examine the history of the CSS clip property along with a short demonstration of the property in action. I will also review the CSS *clip-path* and CSS *clip-mask* properties. Each description will include the syntax for use along with the current browser support to give you a better idea of what can be implemented within your web development environments and customer requirements. Then I will wrap up with several additional resources for reading and demonstrations that highlight the clip, clip-path, and clip-masking properties.

Original specification for CSS Clip

The CSS2.1 specification (http://www.w3.org/TR/CSS21/visufx.html#clipping) for visual effects includes the clip property, which came out some time ago, and was limited to a rectangular shape defined by offsets from the border edge of the box with distances for top, right, bottom, and left. In the example code snippet below, the only valid value works for an absolute positioned element with the following syntax:

```
rect (top, right, bottom, left).
```

And the example CSS code snippet below:

```
img {

   position: absolute;

   clip: rect(40px,520px,400px,80px);

}
```

Notice that the image position is set to absolute; also of note is that both the top and bottom values are offsets from the top border and not respectively from top and bottom borders. The Grand Canyon example below shows the "before" image in **Figure B**, and then after the clip property is applied in **Figure C**:

Figure B

(https://tr1.cbsistatic.com/hub/i/2015/05/07/5e8de710-f4ae-11e4-940f-14feb5cc3d2a/04162013figure-b.gif)
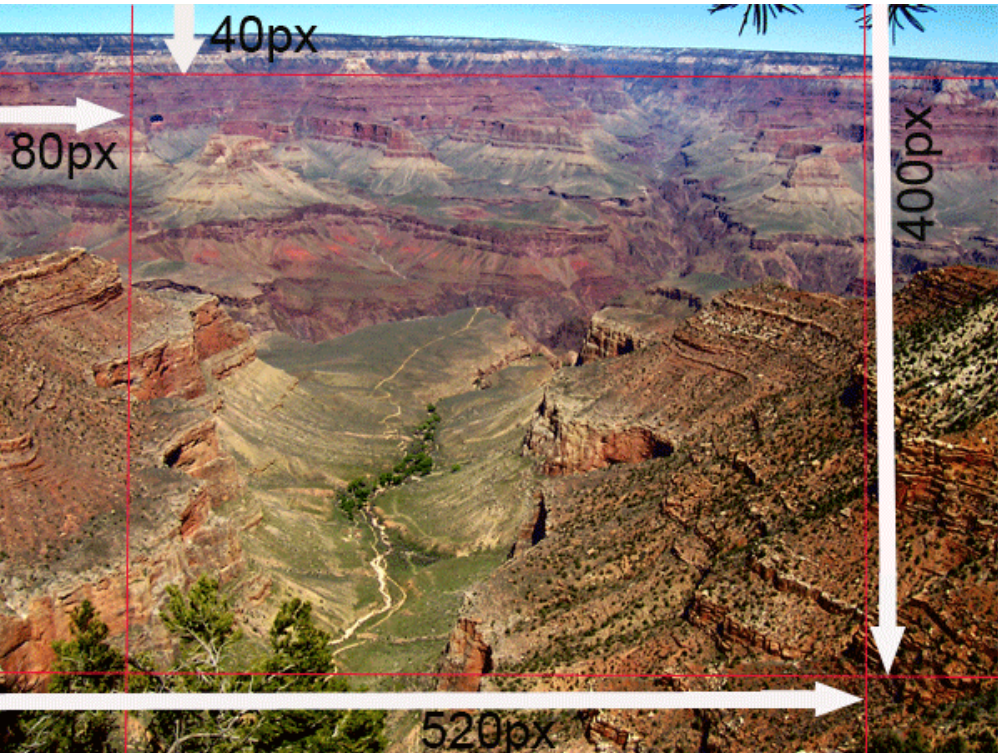
Before image with no clipping applied

Figure C

After image with clipping applied

The grid lines below demonstrate how the clipping effect is applied in the original image as seen in **Figure D** below:

Figure D

Once you see the clipping in action, it is easier to understand the logic behind how the element gets clipped. Probably the hardest concept to wrap your head around is getting the bottom and right clipping set to where you want the element to be clipped. Naturally you would think of clipping the right from the right edge and the bottom from the bottom edge, but it is the opposite, you clip the bottom as it relates to the top edge and the right as it relates to the left edge as illustrated above. If you can think in terms of a cascading waterfall, then you might grab the concept much easier, with a top down and then left to right model.

Basic browser support (http://www.impressivewebs.com/css-clip-property/) for the CSS clip property is limited in IE6 and IE7; however, a workaround for IE requires you to separate the values with white space in place of the comma separated values.

Clip-path property

The W3C Scalable Vector Graphics (SVG) 1.1 (Second Edition) (http://www.w3.org/TR/SVG11/masking.html#ClipPathProperty) specification states that it can contain any HTML element including path elements, text elements, basic shapes or use elements, and SVG container elements as well. If a use element is a child of a <clipPath> element, it must directly reference path, text, or basic shape elements. The <clipPath> element can also be combined with multiple graphical elements to create a merged single-clipping region.

The syntax for clip-path is represented below for shapes, URLs or none:

```
clip-path: <shape> | <url> | none
```

The basic shapes and graphical elements in SVG are listed below and can be utilized to define a particular clipping region, for all syntax examples shown below; the *x* refers to top setting and *y* refers to the left setting.

Rectangle

Defines a rectangle with an origin expressed by x and y coordinates, and a size expressed by width and height in pixels. The optional arguments *rx* and *ry* define the "rounded corners" within the horizontal and vertical direction, the syntax is displayed below.

```
rectangle(<x>, <y>, <width>, <height> [, <rx>, <ry>])
```

Circle

Defines a circle was a center point expressed by *cx* and *xy* coordinates, and a radius *r*, the syntax is displayed below.

```
circle(<cx, <cy>, <r>)
```

Ellipse

Defines a circle with a center point expressed by *cx* and *cy*, and two radii *rx* and *ry* for the horizontal and vertical radius paths, the syntax is displayed below.

```
ellipse(<cx, <cy>, <rx>, <ry>)
```

Polygon

Defines a polygon based on the point list expressed in pixels for *x1, y1, x2, y2* and on *xn, yn*.

```
polygon(<x1> <y1>, <x2> <y2>, ..., <xn> <yn>)
```

Basic browser support (https://developer.mozilla.org/en-US/docs/SVG/Element/clipPath#Browser_compatibility) for the clip-path property includes Chrome 1, Firefox 1.5, IE 9, Opera 9, and Safari 3.0.4.

Clip mask property

The W3C Masking specification (https://dvcs.w3.org/hg/FXTF/raw-file/default/masking/index.html) states that it provides a second method for partially or fully hiding portions of visual elements by masking. The specification also details the types of masks which include sixteen variations on the property for layered masks and box-image masks, and they include: mask, mask-image, mask-repeat, mask-position, mask-clip, mask-origin, and mask-type. The mask-box-image property allows an image to be divided into nine tiles with four corners, four edges, and a single middle section. Each tile can be sliced, scaled, and/or stretched into any combination of shape or size that will fit the masked region. The most useful aspect of this property allows you to modify the borders and edges of elements. Several of the box-image mask properties include the mask-box-image, mask-box-image-source, and mask-box-image-slice.

Basic browser support (http://www.browsersupport.net/CSS/-webkit-mask-clip) for the clip mask property is currently limited in Chrome 17+, and Safari 5.1+.

Further reading and examples

For more examples and information on implementing the SVG clipping and masking properties, you can check out the article, "Clipping and Masking (http://commons.oreilly.com/wiki/index.php/SVG_Essentials/Clipping_and_Masking)," from the SVG Essentials wiki by OReilly. For another great example of the clip-path property in action, you can view the SVG clipPath demo (http://labs.sawyerhollenshead.com/lab/svg-clippath/) by Sawyer Hollenshead. More examples that embrace several principles of the clipping, clip-path, and masking properties are brought to

light by Dirk Schulze's article on CSS Masking (http://www.html5rocks.com/en/tutorials/masking/adobe/) from HTML5 Rocks Tutorials (http://www.html5rocks.com/en/tutorials/).

RELATED TOPICS: | MOBILITY | | TECH & WORK | | CXO | | SOFTWARE | | HARDWARE | | SMART PERSONS GUIDES |

About Ryan Boudreaux

Ryan has performed in a broad range of technology support roles for electric-generation utilities, including nuclear power plants, and for the telecommunications industry. He has worked in web development for the restaurant industry and the Federal g...

Recommended

Create the virtual you and join the fun - Play Second life!
Second Life

Cheap Hotels - At Last!
tripsinsider.com

6 month Digital Marketing Course from XLRI - Apply Now
XLRI - Talentedge

5 fundamental differences between Windows 10 and Linux

What is Windows 10 Fall Creators Update? Everything you need to know about Microsoft's big upgrade

10 things you should never do in Excel

ADD YOUR COMMENT

WHITE PAPERS, WEBCASTS, AND DOWNLOADS

White Papers // From IBM

Cognitive systems will change the workplace in ways we haven't yet imagined

There is a tectonic shift in the way organizations work. We expect the same kind of intuitive, tactile experience with our workplace technology that we now take for granted with our smartphones, tablets and gaming systems.

Planning for the workplace of the future means striking the right balance between

finding the right tools for each user today and accessing an infrastructure that can expand with the intelligence and the power of the technology of the future.

DOWNLOAD NOW

White Papers // From IBM

Business Resiliency: The Need for 24/7/365 Operations (Japanese)

DOWNLOAD NOW

White Papers // From IBM

How people can collaborate more efficiently and access the content that matters most to them

LEARN MORE

White Papers // From IBM

Cloud Computing Improves System and Data Backup, Archive and Recovery Operations (Japanese)

DOWNLOAD NOW

White Papers // From IBM

Taming IT Complexity with Managed Services (Japanese)

DOWNLOAD NOW

RECOMMENDED FOR YOU

Maximising Security and Minimising Overheads with Kaspersky Endpoint Security for Cloud

Downloads provided by Kaspersky Lab

GIVE IT A TRY!

EDITOR'S PICKS

NASA's unsung heroes: The Apollo coders who put men on the moon

The state of women in computer science: An investigative report

Elon Musk and the cult of Tesla: How a tech startup rattled the auto industry to its core

Cyberweapons are now in play: From US sabotage of a North Korean missile test to hacked emergency sirens in Dallas

Recommended Content:

White Papers: The Evolving perspective of Business Intelligence Tools - IBM Repo

This Blue Hill Research paper examines the evolving perspective of BI tools and outlines a framework that decision m
follow to invest in an enterprise-wide self-service BI solution that is right for their organization. ...