

Chapter 4 4: Clipping and Masking

| Parameter | Details |
|----------------------|--|
| clip-source | A URL which can point to an inline SVG element (or) an SVG element in an external file that contains the clip path's definition. |
| basic-shape | Refers to one among <code>inset()</code> , <code>circle()</code> , <code>ellipse()</code> or <code>polygon()</code> . Using one of these functions the clipping path is defined. These shape functions work exactly the same way as they do in Shapes for Floats |
| clip-geometry-box | This can have one among <code>content-box</code> , <code>padding-box</code> , <code>border-box</code> , <code>margin-box</code> , <code>fill-box</code> , <code>stroke-box</code> , <code>view-box</code> as values. When this is provided without any value for <code><basic-shape></code> , the edges of the corresponding box is used as the path for clipping. When used with a <code><basic-shape></code> , this acts as the reference box for the shape. |
| mask-reference | This can be <code>none</code> or an image or a reference URL to a mask image source. |
| repeat-style | This specifies how the mask should be repeated or tiled in the X and Y axes. The supported values are <code>repeat-x</code> , <code>repeat-y</code> , <code>repeat</code> , <code>space</code> , <code>round</code> , <code>no-repeat</code> . |
| mask-mode | Can be <code>alpha</code> or <code>luminance</code> or <code>auto</code> and indicates whether the mask should be treated as a alpha mask or a luminance mask. If no value is provided and the mask-reference is a direct image then it would be considered as alpha mask (or) if the mask-reference is a URL then it would be considered as luminance mask. |
| position | This specifies the position of each mask layer and is similar in behavior to the <code>background-position</code> property. The value can be provided in 1 value syntax (like <code>top</code> , <code>10%</code>) or in 2 value syntax (like <code>top right</code> , <code>50% 50%</code>). |
| geometry-box | This specifies the box to which the mask should be clipped (<i>mask painting area</i>) or the box which should be used as reference for the mask's origin (<i>mask positioning area</i>) depending on the property. The list of possible values are <code>content-box</code> , <code>padding-box</code> , <code>border-box</code> , <code>margin-box</code> , <code>fill-box</code> , <code>stroke-box</code> , <code>view-box</code> . Detailed explanation of how each of those values work is available in the W3C Spec . |
| bg-size | This represents the size of each mask-image layer and has the same syntax as <code>background-size</code> . The value can be length or percentage or auto or cover or contain. Length, percentage and auto can either be provided as a single value or as one for each axis. |
| compositing-operator | This can be any one among <code>add</code> , <code>subtract</code> , <code>exclude</code> , <code>multiply</code> per layer and defines the type of compositing operation that should be used for this layer with those below it. Detailed explanation about each value is available in the W3C Specs . |

Section 4 4.1: Clipping and Masking: Overview and Difference

With **Clipping** and **Masking** you can make some specified parts of elements transparent or opaque. Both can be applied to any HTML element.

Clipping

Clips are vector paths. Outside of this path the element will be transparent, inside it's opaque. Therefore you can define a `clip-path` property on elements. Every graphical element that also exists in SVG you can use here as a function to define the path. Examples are `circle()`, `polygon()` or `ellipse()`.



Example

```
clip-path: circle(100px at center);
```

The element will be only visible inside of this circle, which is positioned at the center of the element and has a radius of 100px.

Masking

Masks are similar to Clips, but instead of defining a path you define a mask what layers over the element. You can imagine this mask as an image what consist of mainly two colors: black and white.

Luminance Mask: Black means the region is opaque, and white that it's transparent, but there is also a grey area which is semi-transparent, so you are able to make smooth transitions.

Alpha Mask: Only on the transparent areas of the mask the element will be opaque.



This image for example can be used as a luminance mask to make for an element a very smooth transition from right to left and from opaque to transparent.

The mask property let you specify the the mask type and an image to be used as layer.

Example

```
mask: url(masks.svg#rectangle) luminance;
```

An element called `rectangle` defined in `masks.svg` will be used as an **luminance mask** on the element.

Section 4 4.2: Simple mask that fades an image from solid to transparent

CSS

```
div {  
  height: 200px;  
  width: 200px;  
  background: url(http://lorempixel.com/200/200/nature/1);  
  mask-image: linear-gradient(to right, white, transparent);  
}
```

HTML

```
<div></div>
```

In the above example there is an element with an image as its background. The mask that is applied on the image (using CSS) makes it look as though it is fading out from left to right.

The masking is achieved by using a linear-gradient that goes from white (on the left) to transparent (on the right) as the mask. As it is an alpha mask, image becomes transparent where the mask is transparent.

Output without the mask:



Output with the mask:



Note: As mentioned in remarks, the above example would work in Chrome, Safari and Opera only when used with the `-webkit` prefix. This example (with a linear-gradient as mask image) is not yet supported in Firefox.

Section 4 4.3: Clipping (Circle)

CSS:

```
div{
```

```
width: 200px;
height: 200px;
background: teal;
clip-path: circle(30% at 50% 50%); /* refer remarks before usage */
}
```

HTML

```
<div></div>
```

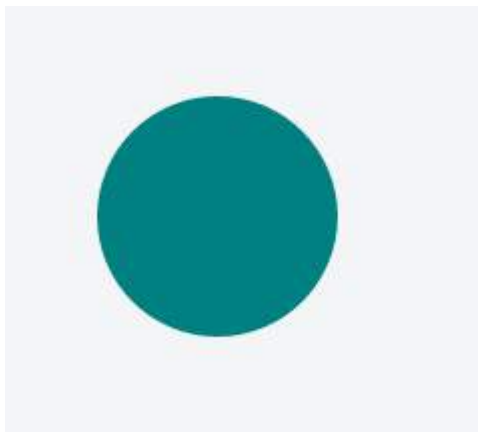
This example shows how to clip a div to a circle. The element is clipped into a circle whose radius is 30% based on the dimensions of the reference box with its center point at the center of the reference box. Here since no `<clip-geometry-box>` (in other words, reference box) is provided, the **border-box** of the element will be used as the reference box.

The circle shape needs to have a radius and a center with (x,y) coordinates:

```
circle(radius at x y)
```

[View Example](#)

Output:



Section 4 4.4: Clipping (Polygon)

CSS:

```
div{
  width:200px;
  height:200px;
  background:teal;
  clip-path: polygon(0 0, 0 100%, 100% 50%); /* refer remarks before usage */
}
```

HTML:

```
<div></div>
```

In the above example, a **polygonal** clipping path is used to clip the square (200 x 200) element into a triangle shape. The output shape is a triangle because the path starts at (that is, first coordinates are at) `0 0` - which is the top-left corner of the box, then goes to `0 100%` - which is bottom-left corner of the box and then finally to `100% 50%` which is nothing but the right-middle point of the box. These paths are self closing (that is, the starting point will be the ending point) and so the final shape is that of a triangle.

This can also be used on an element with an image or a gradient as background.

[View Example](#)

Output:



Section 4 4.5: Using masks to cut a hole in the middle of an image

CSS

```
div {  
  width: 200px;  
  height: 200px;  
  background: url(http://lorempixel.com/200/200/abstract/6);  
  mask-image: radial-gradient(circle farthest-side at center, transparent 49%, white 50%); /* check  
  remarks before using */  
}
```

HTML

In the above example, a transparent circle is created at the center using `radial-gradient` and this is then used as a mask to produce the effect of a circle being cut out from the center of an image.

Image without mask:



Image with mask:



Section 44.6: Using masks to create images with irregular shapes

CSS

```
div { /* check remarks before usage */  
  height: 200px;  
  width: 400px;  
  background-image: url(http://lorempixel.com/400/200/nature/4);  
  mask-image: linear-gradient(to top right, transparent 49.5%, white 50.5%), linear-gradient(to top  
left, transparent 49.5%, white 50.5%), linear-gradient(white, white);  
  mask-size: 75% 25%, 25% 25%, 100% 75%;  
  mask-position: bottom left, bottom right, top left;  
  mask-repeat: no-repeat;  
}
```

HTML

```
<div></div>
```

In the above example, three linear-gradient images (which when placed in their appropriate positions would cover 100% x 100% of the container's size) are used as masks to produce a transparent triangular shaped cut at the bottom of the image.

Image without the mask:



Image with the mask:

