



CSS clipping and masking

Both Clipping and masking hide some parts of elements and show other parts. But there are, of course, differences between the two. Differences in what they can do, differences in syntaxes, different technologies involved, the new and the deprecated, and browser support differences.

clip-path and CSS

The clip-path CSS property allows you to specify a specific region of an element to display, instead of the complete area. There used to be a clip property, but it has been deprecated.

The common HTML for clipping with CSS

This is the markup we'll use for the examples below.

```
<div class="box">
  <h1>Box 1</h1>
</div>
```

Clip Path

The first exercise will create a purple circle with the string "box1" centered vertically and horizontally.

```
.box {
  display: flex;
  flex direction: column;
  align-items: center;
  justify-content: center;

  color: #fff;
  height: 300px;
  width: 300px;
  background-color: rebeccapurple;
```

```
clip-path: circle(50% at center center);
}
```

Animating clip-path

Once we have the shape defined, we can animate it using CSS transitions and animations.

To accomplish this we need to do two things:

1. add a transition attribute to the .box element
2. add a :hover pseudo class for the .box element where we change the clip path to whatever we want to use, in this case a smaller circle.

```
.box {
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;

  color: #fff;
  height: 300px;
  width: 300px;
  background-color: rebeccapurple;
  clip-path: circle(75%);
  transition: clip-path 1s; /* 1 */
}

.box:hover { /* 2 */
  clip-path: circle(20%);
}
```

Clipping with SVG

Developers have been able to do clipping and masking with SVG. We'll SVG to clip parts of an element from CSS.

The common HTML for clipping with SVG

This is the markup we'll use for the examples clipping using an SVG element. The HTML may look a little weird but it's OK to put the SVG inside the HTML document as this particular SVG fragment will not render anything, it just defines the shape we will use in CSS.

```
<svg class="svg-defs">
  <defs>
    <clipPath id="clipping">
      <polygon id="Star-1"
        points="98.4999978 153.75 38.2520165
          185.424245 49.7583542 118.337123
          1.01670635 70.8257603 68.3760155
          61.037872 98.5000012 1.1379786e-14
          128.624005 61.0378871 195.98331
          70.8258091 147.241642 118.337136
          158.747982 185.424247"/>
    </clipPath>
  </defs>
</svg>

<div class="box">
  <h1>Box 1</h1>
</div>
```

The CSS changes only slightly, instead of specifying a shape and a position, we use a relative URL to the SVG definition (`clip-path: url('#clipping')`)

```
.box {
  height: 300px;
  width: 300px;
  background-color: rebeccapurple;
  clip-path: url(#clipping);
}
```

Links

- [clip-path](#)
- [Animating with Clip-Path](#)
- [Clipping and Masking in CSS](#)
- [Masking vs. Clipping: When to Use Each](#)