

## レスポンス対応をしながら、Clippathでsvgを使ってクリッピング！

WEB DESIGN

2021-10-08 / Updated: 2022-01-06

# Clip-path



clippathとは、指定をした特定の形に切り抜くためのCSSプロパティです。

svgのパスデータなどを使うことができ、クリッピングをすると、切り抜いた外側は表示されません。

改めて理解を深めたかったので、レスポンス対応の仕方も含めて、clippathについての使い方をまとめてみました。

### CONTENTS

- 1 clippathの基本の使い方
  - ▶ 基本的な形の指定
- 2 clip-sourceの利用
- 3 レスポンス対応をする！
- 4 clippathのジェネレーター
- 5 ブラウザのサポート状況は？
- 6 まとめ

# clippathの基本の使い方

切り抜きたい画像にたいして、**clippath:OO;**でパスの形を指定します。例えば、下記のような書き方です。

HTML

```
<p class="sample"></p>
```

CSS

```
img{
width:100%;
height:auto;
object-fit:cover;
}

.sample{
width:150px;
height:150px;
}

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

## 基本的な形の指定

切り抜くための基本的な形は、以下のような指定ができます。

inset()	<b>inset( Top Right left Bottom (round OO );</b> 内側へオフセットします。roundを追加すると、角丸指定になります clip-path: inset(10px 20px 10px 20px round 5px);
circle()	<b>circle(円の半径 at 位置);</b> 円形の指定になります。 clip-path: circle(50px at center); clip-path: circle(50% at 50% 50%);
ellipse()	<b>ellipse(横方向 縦方向 at 位置);</b> 楕円形の指定になります。 clip-path: ellipse(50% 35% at 50% 50%); clip-path:ellipse(50% 25% at center);
polygon()	<b>polygon( x1 y1,x2 y2, ...);</b> 多角形の指定になります。それぞれ、角になる点の位置をx,yで示します。 clip-path: polygon(50% 0%, 100% 38%, 82% 100%, 18% 100%, 0% 38%);
path()	<b>url(#インラインclipPathの#id名);</b> 任意のパスの指定になります。 clip-path: url(#clipping-mask);



inset



circle



ellipse



polygon



path

## clip-sourceの利用

1はそのまま直接指定したクラスに切り抜きを反映させるやり方ですが、インラインのsvgを別の箇所で用意しておく(clip-source)、id名をつけておいて、**clip-path: url(#ID名);**で呼び出すことも可能です。

注意点として、別の場所を用意しておくsvgは、**width="0" height="0"**としておき、単体としての表示がされないようにしておきます。また**svgタグのviewbox属性を必ず記述**しておきましょう。レスポンス対応で必要になります。下記のような書き方になります。

### HTML

```

<svg width="0" height="0" viewBox="0 0 400 400">
  <defs>
    <clipPath id="clipshape">
      <path d='M0,100C0,44.8,44.8,0,100,0s100,44.8,100,100C200,44.8,244.8,0,300,0s100,44.8,100,100c0,100-66.7,200-200,300
        C66.7,300,0,200,0,100z'>
    </clipPath>
  </defs>
</svg>
```

### CSS

```
.clip-me{
  clip-path: url(#clipshape);
}
```

## レスポンス対応をする！

path()を使い、ハートマークのインラインsvgを用意して切り抜いてみました。

### HTML

```
<p class="sample"></p>
<svg width="0" height="0" viewBox="0 0 400 400">
  <clipPath id="clipshape">
    <path
d='M0,100C0,44.8,44.8,0,100,0s100,44.8,100,100C200,44.8,244.8,0,300,0s100,44.8,100,100c0,100-66.7,200-200,300
  C66.7,300,0,200,0,100z'>
    </clipPath>
</svg>
```

### CSS

```
.clip-me{
  clip-path: url(#clipshape);
}
```

%で切り抜きのサイズ指定している円形などとは異なり、path()で上記のように指定した画像の方は、**切り抜きのサイズが固定**されており、そのままだとレスポンス非対応で、幅が狭まった際に切れてしまうようです。

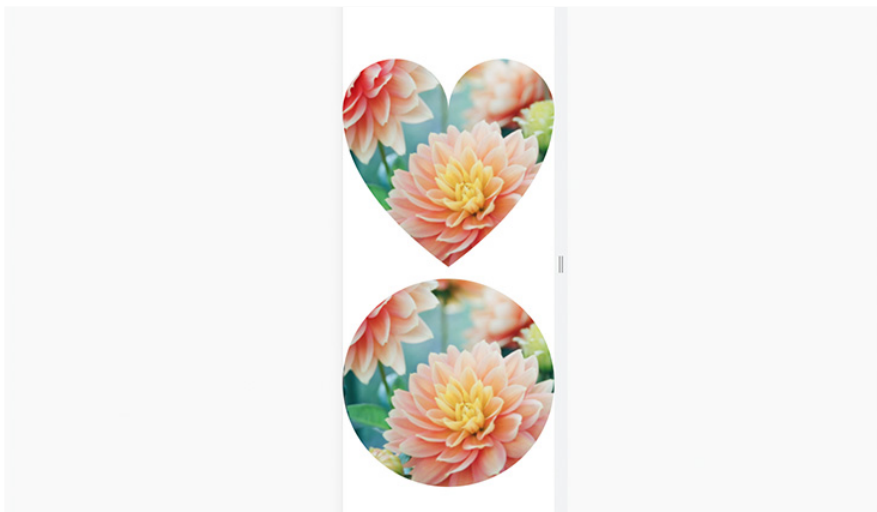


path()の時にもレスポンス対応をしたいので、下記の2点を追加します。

- ▶ **<clipPath>**にclipPathUnits="objectBoundingBox"
- ▶ **<path>**にtransform="scale( 1/ViewBoxのwidth幅 , 1/ViewBoxのheight幅)"

#### HTML

```
<p class="sample"></p>
<svg width="0" height="0" viewBox="0 0 400 400">
  <clipPath id="clipshape" clipPathUnits="objectBoundingBox">
    <path transform="scale(0.0025,0.0025)"
d='M0,100C0,44.8,44.8,0,100,0s100,44.8,100,100C200,44.8,244.8,0,300,0s100,44.8,100,100c0,100-66.7,200-200,300
C66.7,300,0,200,0,100z'>
  </clipPath>
</svg>
```



上記二点を追加したことで、ちゃんと幅に合わせた表示になりました！

それぞれの記述の意味合いとしては、以下のようになります。

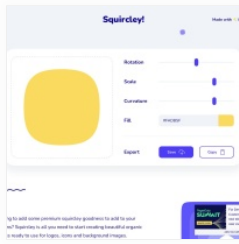
**clipPathUnits="objectBoundingBox"**を追加することで、切り抜くパスのサイズを、切り抜きたい要素のサイズに応じて相対的に変える設定が可能になります。

その時、切り抜きたい画像の左上を(0,0)、右下を(1,1)の位置とすると、切り抜くパスを0-1の範囲に収める必要があるため、**transform="scale()"**で縮小します。例えばパスの大きさ(viewbox)がX軸幅100、Y軸高さ150だったら、それぞれ1/100 1/150 でtransform="scale(1/100, 1/150)"を指定します。

# clippathのジェネレーター

svg形式の画像やclippathのデータを作るのに、ジェネレーターサイトを使って作るのも簡単おすすめです。

## 【 1 Squircle! 】

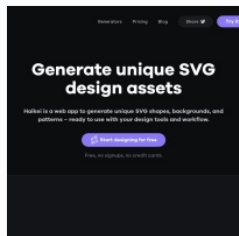


<https://squircle.app/>

シンプルな角丸の長方形をつくるサイトです。

機能としては少ないけれど、親しみを感じる形が可愛い印象です。

## 【 2 Haikai 】



### Generate unique SVG design assets | Haikai

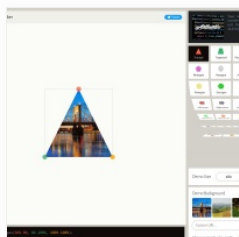
A web-based design tool to generate unique SVG design assets...

<https://haikai.app/>

名前の通り、背景画像に使われそうなパス画像の制作ができます。

うまく使えばとてもモダンなサイトになりそう。

## 【 3 clippy 】



### Clippy — CSS clip-path maker

<https://bennettfeely.com/clippy/>

名前の通り、クリッピングをしたい時にピッタリのサイトです。

基本的な形を作りたい時は特に活躍しそうです。

## ブラウザのサポート状況は？



IE以外は基本的に対応していますが、  
Chrome とsafari、Edgeは、`url(#foo)`で内部のSVGを呼び出すのは対応しているものの  
**外部のSVG呼び出しには対応していない**(2022年1月の時点)ため、その点は使用する際に注意が必要です。

## まとめ

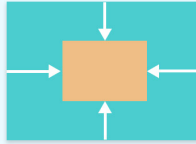
photoshopなどでもう切り抜きが済んでいる画像を使っている場合、  
その目的で使うことは少なそうなclippath ですが、  
手段のひとつとして知っておくのもアリかなと思いました。

さらに、animationと組み合わせて使うことも可能なため、  
改めて時間があるときにまとめてみたいと思います。

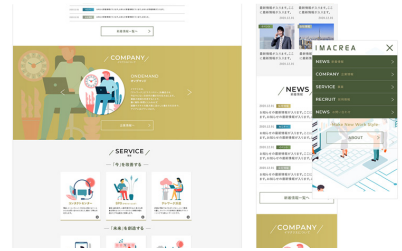


商用可！Unsplashをダミー画像でつかう使い方

## Center



上下や上下左右を中央揃えにするCSSの方法8つ



Web Design for Imacrea Company

## SNS Share



Word Press

プラグインなしでWordPressのSNSボタンを追加する方法

## Resize Images



Word Press

wordpressで、リサイズされる画像の自動生成を停止する方法

## Flex:1;

Flex:1; Flex:2;

まとめて指定！flexbox子要素を決めるflexプロパティについて



Mana

ウサギと美術館好きな、ロンドン在住のイラストレーター&デザイナーです。Webやデザインに関するアウトプットがメインです。

イギリス暮らしのあれこれを綴ったマンガ+UKはこちら↓

Manga+UK

## CATEGORY

App (5)  
Illustration (4)  
Others (10)  
UK (9)  
Web Design (25)



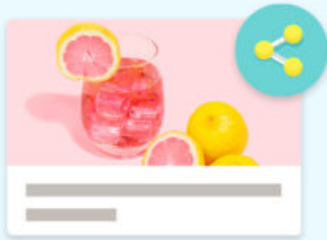
## Flex:1;

Flex:1;

Flex:2;

まとめて指定！flexbox子要素を決めるflexプロパティについて

## OGP



【内部SEO対策】OGPの<meta>タグとOGP画像の設定方法

## SSL



【常時SSL化】HTTPSと、リダイレクトなどの設定方法

## Google Material Icons



商用利用もOK！Google Material Iconsの使い方

## FREE ICONS

おすすめしたい、アイコンのフリー素材のwebサイト10選