



# CSS Shapes: Once more, with feeling

CSS Shapes are an awesome way to make our designs look less like boxes inside boxes inside boxes. It let us wrap text around and image regardless of its shape.

I've written before about shapes. The posts, in erverse chronological order:

- [Getting fancy: Shapes and masks](#)
- [SVG Clip Path and Shapes. An interesting alternative](#)
- [CSS shapes: an update and an expansion](#)
- [CSS regions, exclusions, shapes and new publishing paradigms](#)

There are news and new tools to use with shapes and that got me excited again so here we go again. :)

## So What Are CSS Shapes?

[CSS Shapes](#) describe geometric shapes for use in CSS. For Level 1, CSS Shapes can be applied only to floats. A circle shape on a float will cause inline content to wrap around the circle shape instead of the float's bounding box.

We can play with standard shapes like circles or ellipses or we can create, irregular polygons or other custom shapes. The text will automatically flow around them.

So you may be asking why is this important... who actually cares?

For a long time I've thought that it is possible to create web layouts that are as engaging as those in printed media. Shapes are a step in the right direction and it oppens possibilities for more creative views of the same content. For those browsers that don't support shapes the text will flow round the box like we are used to... not that big a loss.

# Shapes: from the simple to the complex

One of the first examples I saw was text shaped outside a circular image.

But there are much more intricate shapes you can draw and use with your content. We can start with triangles and other geometrical shapes.

When creating the additional shapes you can leverage polygons to make the content more interesting, even if the shape itself may not be.

You can also give the impression of the text being inside an image (what the level shape specification calls shape inside) by placing the text between two images and using shape-outside on each one of them.

The best example of this techniques is Adobe's [Demo for Alice in Wonderland by Lewis Carroll](#) and document in the [Adobe Web Platform blog](#) Retrieved from the Wayback machine.

# Why New Tools?

The biggest problems when creating polygons to use with CSS Shapes is that they are far from intuitive. To create a triangle you have to use the following CSS:

```
img#penetrator {  
  width: 50%;  
  margin-left: -12%;  
  float: left;  
  shape-outside: polygon(24% 0, 24% 100%, 100% 54%);  
}
```

The Polygon gets more complicated the larger and the more irregular shape you want to use. Coding the polygon shape by hand but it's complicated enough to make it very hard.

That's where Firefox's [CSS Shape Editor](#) comes in.

If you've already created the shape you can customize it, however it will not let you create a brand new shape visually or add anchor points to an existing shape.

It's not a complete shape editor but it's still the best way to play with shapes in a live browser environment.