BORDER IMAGES

For use with:
Learning Web Design, 5e

by Jennifer Robbins

In this article, I'll give you a quick overview of the **`border-image`** properties for filling the sides and corners of a border box with an image of your choice. Because of the complexity of border images, there's a lot more to say than I have room for here, so if you are excited about getting the most out of border images, I recommend you read more about them here:

- The CSS Background and Borders Module Level 3 (*www.w3.org/TR/css3-background/#the-border-image-source*)
- For a less dense explanation, see the **`border-image`** listing on CSS-Tricks (*css-tricks.com/almanac/properties/b/border-image/*)

Let's kick off this discussion with a visual to give you an idea of what I'm talking about. FIGURE A shows two elements and the respective images used to fill their borders. Notice that the corners of the image get placed in the corners of the elements. The sides of the image either stretch or repeat to fill the four sides. The images on the right show the same elements with their fallback border and background images. This is what users will see on non-supporting browsers or if the image fails to load.

IN THIS ARTICLE

THE BACKGROUND IMAGE PROPERTIES

Border images are created with a collection of five **`border-image-*`** properties, or the shorthand **`border-image`** (see **Browser Support Note**):

```
border-image-source
border-image-slice
border-image-width
border-image-outset
border-image-repeat
border-image
```

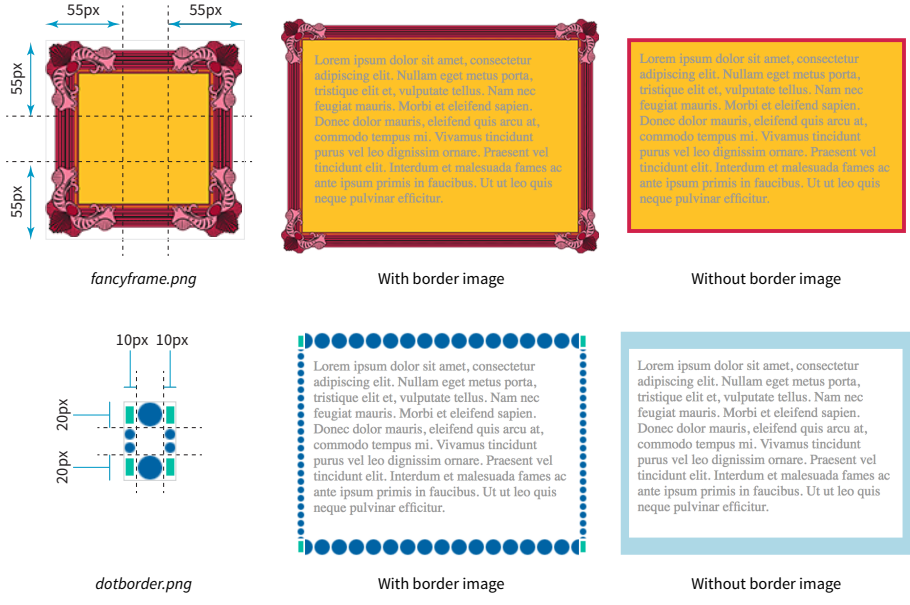


FIGURE A. Examples of border images with stretched sides and repeated sides.

BROWSER SUPPORT NOTE

All modern browsers now support the border image properties without vendor prefixes. Internet Explorer versions 10 and earlier do not support border images at all, so be sure to provide reasonable border and background fallbacks.

Browsers released prior to 2012 (Safari <6, Opera <12, Chrome <15, Firefox <15, iOS <6.1, and Android <4.4) support only the shorthand **`browser-image`** *property with* **`-webkit-`** *and* **`-moz-`** *prefixes and do not support the five* **`border-image-`** *properties individually.*

Let's start by taking a look at the individual properties and how they work.

`border-image-source`

Values: *URL for image* | none

Default: none

Applies to: all elements except internal table elements where `border-collapse` is set to `collapse`

Inherits: no

This one is self-explanatory. It specifies the location of the image to be used for the border. As you can see from FIGURE A, although not a requirement, it makes sense for the border image to be specially designed for use as a border. You can use any bitmapped image, SVGs, and even gradients as border images. Here are the declarations for the top and bottom examples in the figure, respectively:

```
border-image-source: url(fancyframe.png);
border-image-source: url(dotborder.png);
```

`border-image-slice`

Values: *percentage* | *number (of pixels)* | none | `fill`

Default: 100% (causes the whole image to be used in the corners, with nothing on the sides)

Applies to: all elements except internal table elements where `border-collapse` is set to `collapse`

Inherits: no

The **border-image-slice** property divides the image into nine sections (four corners, four sides, and a center) by using offset measurements from the top, right, bottom, and left sides.

This is another property that applies values in top, right, bottom, left (TRBL) order. You can specify four values (one for each side) or fewer, and they will repeat, as you've seen for padding and border values. The value is a number (indicating pixels) or a percentage of the width of the border box.

By default, the center section of the image is not displayed, but you can override that with the optional **fill** keyword after the slice value to make it display (see **Browser Support Note**).

The *fancyframe.png* image is sliced 55 pixels from each side, so I can provide one measurement for use on all four sides. Note that the value specifies a number only, indicating a number of pixels for the offset. You *must* omit the "px" after the measurement, or it will not work. I want the center of the frame image to fill the center of the element, so I've added the **fill** keyword:

```
border-image-slice: 55 fill;
```

The *dotborder.png* image is sliced 20 pixels from the top and bottom and 10 pixels from the left and right:

```
border-image-slice: 20 10;
```

border-image-width

Values: *length* | *percentage* | *number* | auto

Default: 1 (causes the border image to be the same as the border-width for the element)

Applies to: all elements except internal table elements where border-collapse is set to collapse

Inherits: no

By default, the width of the border image is the same as the **border-width** for the element. In the frame example in FIGURE A, I could set **border-width: 55px** to make the 55-pixel-wide slices fit exactly. But should the image not load, I'd be stuck with a weirdly large border around the content. To size the width of the border image independently, use the **border-image-width** property. Again, use the TRBL system for specifying lengths (such as pixels or ems) or percentages:

```
border: 5px solid #d1214a; /* red */
border-image-width: 55px;
```

You can also specify a number that gets used as a multiplier of the **border-width**. In the previous example, the **border-width** is 5, and I want the **border-image-width** to be 55, so I could have specified **border-image-width: 11** for the same result.

BROWSER SUPPORT NOTE

According to the CSS3 spec, browsers should not render the center of the image by default, but older browsers always display it. If you want to provide a different background color or image in the background of the content box and be sure it works in all browsers, create the border image with a transparent center so the background shows through.

TIP

If you use **border-image-width**, you may need to increase the padding and possibly the margin to accommodate the image. It won't create space for itself.

border-image-repeat

Values: `stretch` | `repeat` | `round` | `space`

Default: `1` (causes the border image to be the same as the `border-width` for the element)

Applies to: all elements except internal table elements where `border-collapse` is set to `collapse`

Inherits: no

Take another look at the examples in FIGURE A and you will see that they differ in the way the sides are filled in. In the frame image, the sides of the image stretch to fill the sides of the border. That is the default behavior, although it can be specified explicitly with the **`stretch`** keyword. By contrast, the dot in the bottom example repeats to fill the available space on the sides, as specified with the **`repeat`** keyword:

```
border-image-repeat: repeat;
```

FIGURE B shows the results of applying the keywords **`stretch`**, **`repeat`**, and **`round`**. The **`round`** and **`space`** keywords work the same as they do for background images; however, **`space`** has very poor browser support as of this writing. The **`round`** keyword may display as **`repeat`** in non-supporting browsers. See the **Browser Support Note** for details.

border-image-outset

Values: *length* | *number* | `inherit`

Default: `0` (causes the border image to be placed from the outside border edge into the padding or content area)

Applies to: all elements except internal table elements where `border-collapse` is set to `collapse`

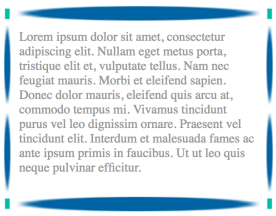
Inherits: no

BROWSER SUPPORT NOTE

The **`space`** *keyword is supported only on Internet Explorer 11 and Microsoft Edge as of this writing. No mobile browsers or other desktop browsers currently support it. The* **`round`** *keyword gained support in Chrome, Firefox, Opera, and Android around 2013, but Safari/iOS did not start supporting it until 2016 with versions 9.1 and 9.3, respectively.*

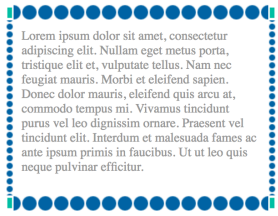
dotborder.png

The source image.



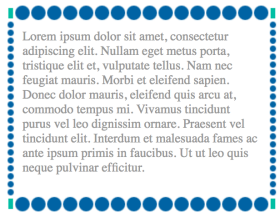
`stretch`

By default, the side segment stretches to fill the side.



`repeat`

The side segment of the image is repeated, starting at the center. It may get clipped at the ends, as shown.



`round`

The side segment of the image is repeated an even number of times by stretching or smooshing the image slightly to fit.

FIGURE B. Comparison of **`stretch`**, **`repeat`**, and **`round`** values for **`border-image-repeat`**.

One of the problems you may run into with border images is that they can overlap or bump into the content box. You can add a large amount of padding to keep the area clear, but then if the border image doesn't load, you're left with a lot of wasted space. The solution is to use the **`border-image-outset`** property to push the border image away from the content and into the margin.

FIGURE C shows the framed border as it appears with the 25-pixel outset and how it would appear by default with the image positioned inside the element box:

```
border-image-outset: 25px;
```

`border-image-outset: 25px;`

`border-image-outset: 0;` (default)

FIGURE C. The **`border-image-outset`** property pushes the image away from the edges of the border box.

Now that we are familiar with all the individual settings, let's see how they work in the **`border-image`** shorthand.

THE BORDER-IMAGE SHORTHAND

`border-image`

Values: *border-image-source* [*border-image-slice* / *border-image-width* / *border-image-outset*] *border-image-repeat*

Default: defaults for each property

Applies to: all elements except table elements where `border-collapse` is `collapse`

Inherits: no

The **`border-image`** shorthand combines values for the five **`border-image-*`** properties we've examined so far. However, because the values of those properties look very similar, there are some rules for keeping the values sorted. First, the values for slice, width, and outset must always appear in that order (S/W/O) and they must be separated by the forward slash (**/**) symbol.

(Technically, it's a *solidus* symbol, but the slash key on your keyboard does the trick.)

The **source** and **repeat** values may appear in any order around that set. Remember that if you omit values, they reset to the default for that property. Here is how the syntax looks:

```
border-image: source slice / width / outset repeat;
```

As long as slice-width-outset are kept in the proper order and separated by slashes, you can move the other values around:

```
border-image: source repeat slice / width / outset;
```

PUTTING IT ALL TOGETHER

Now let's see what that looks like in a real example, getting back to the fancy frame example in FIGURE A. Here is the complete style rule:

```
.framed {
  color: #999;
  width: 300px;
  height: 200px;
  padding: 10px;
  background-color: #fec227; /* same as image center */
  border: 5px solid #d1214a; /* red */
  border-image-source: url(fancyframe.png);
  border-image-slice: 55 fill;
  border-image-width: 55px;
  border-image-repeat: stretch;
  border-image-outset: 25px;
}
```

Notice that I've applied a background color and border to the element to be used as a fallback should the image not load and for non-supporting browsers. Also, according to the specification, there must be a **border-style** specified (other than **none**) in order for the border image to display; however, WebKit browsers may render the image anyway.

Here is the shorthand version, omitting the default **stretch** value:

```
.framed {
  …
  border-image: url(fancyframe.png) 55 fill / 55px / 25px;
}
```

The shorthand version sure does save a lot of typing, not to mention slightly reducing the size of the CSS file.

Here is the shorthand rule for the dotted border example:

```
.dotted {
  color: #999;
  width: 300px;
  height: 200px;
  padding: 10px;
  background-color: white;
  border: solid lightblue;
  border-width: 20px 10px;
  border-image: url(dotborder.png) 20 10 repeat;
}
```

Note that in this example, I have omitted the **`border-image-width`** value because it is fine for it to use the values set for **`border-width`**. I have also omitted the **`-outset`** value because I did not use it.

You might not need to use border images every day, but they may create a nice effect if used cleverly.

CSS REVIEW: BORDER IMAGE PROPERTIES

Property	Description
`border-image`	Shorthand property for specifying a border image
`border-image-source`	Provides the location of the image file to be used in the border
`border-image-slice`	Specifies the offset points at which the image is sliced to be placed in the nine areas of the border
`border-image-width`	Specifies the width of the space allocated to the border image
`border-image-outset`	Specifies a length by which the border image should be pushed out toward the element's margin
`border-image-repeat`	Specifies the manner in which the image is applied to the sides of the border