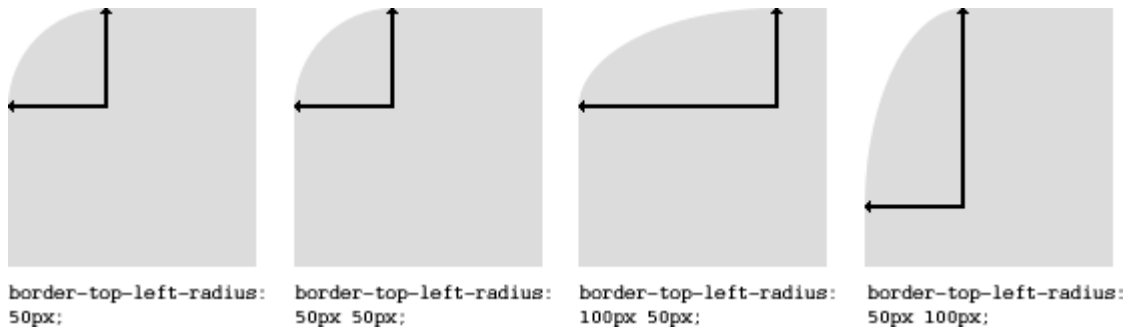


# Chapter 10: Border

## Section 10.1: border-radius

The border-radius property allows you to change the shape of the basic box model.

Every corner of an element can have up to two values, for the vertical and horizontal radius of that corner (for a maximum of 8 values).



The first set of values defines the horizontal radius. The optional second set of values, preceded by a `'/'`, defines the vertical radius. If only one set of values is supplied, it is used for both the vertical and horizontal radius.

```
border-radius: 10px 5% / 20px 25em 30px 35em;
```

The `10px` is the horizontal radius of the top-left-and-bottom-right. And the `5%` is the horizontal radius of the top-right-and-bottom-left. The other four values after `'/'` are the vertical radii for top-left, top-right, bottom-right and bottom-left.

As with many CSS properties, shorthands can be used for any or all possible values. You can therefore specify anything from one to eight values. The following shorthand allows you to set the horizontal and vertical radius of every corner to the same value:

HTML:

```
<div class='box'></div>
```

CSS:

```
.box {  
  width: 250px;  
  height: 250px;  
  background-color: black;  
  border-radius: 10px;  
}
```

Border-radius is most commonly used to convert box elements into circles. By setting the border-radius to half of the length of a square element, a circular element is created:

```
.circle {  
  width: 200px;  
  height: 200px;  
  border-radius: 100px;  
}
```

Because border-radius accepts percentages, it is common to use 50% to avoid manually calculating the border-radius value:

```
.circle {  
  width: 150px;  
  height: 150px;  
  border-radius: 50%;  
}
```

If the width and height properties are not equal, the resulting shape will be an oval rather than a circle.

Browser specific border-radius example:

```
-webkit-border-top-right-radius: 4px;  
-webkit-border-bottom-right-radius: 4px;  
-webkit-border-bottom-left-radius: 0;  
-webkit-border-top-left-radius: 0;  
-moz-border-radius-topright: 4px;  
-moz-border-radius-bottomright: 4px;  
-moz-border-radius-bottomleft: 0;  
-moz-border-radius-topleft: 0;  
border-top-right-radius: 4px;  
border-bottom-right-radius: 4px;  
border-bottom-left-radius: 0;  
border-top-left-radius: 0;
```

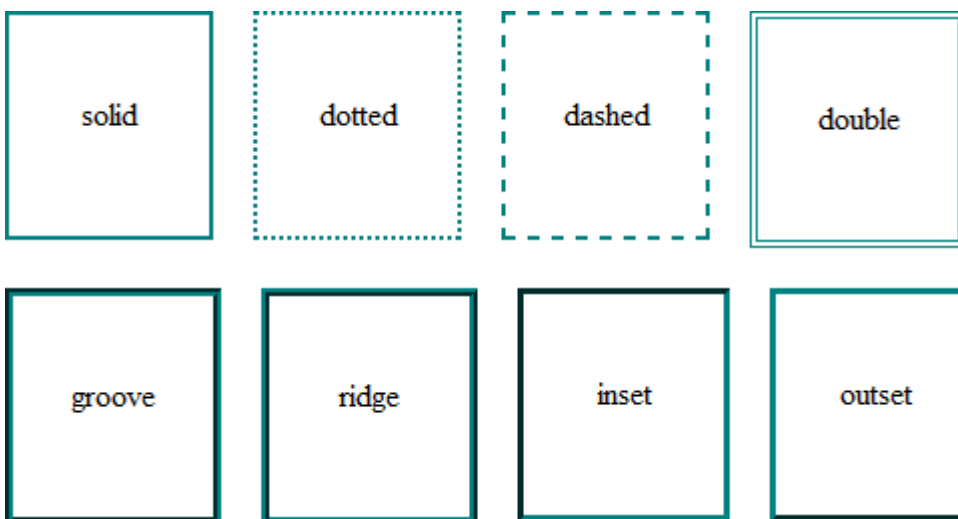
## Section 10.2: border-style

The border-style property sets the style of an element's border. This property can have from one to four values (for every side of the element one value.)

Examples:

**border-style:** dotted;

**border-style:** dotted solid double dashed;



border-style can also have the values none and hidden. They have the same effect, except hidden works for border conflict resolution for <table> elements. In a <table> with multiple borders, none has the lowest priority (meaning in a conflict, the border would show), and hidden has the highest priority (meaning in a conflict, the border would not show).

## Section 10.3: Multiple Borders

Using outline:

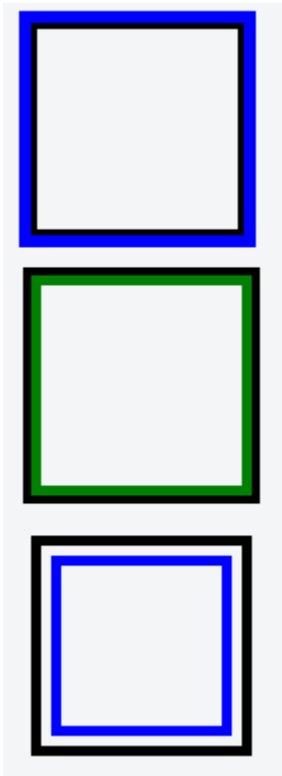
```
.div1 {  
  border: 3px solid black;  
  outline: 6px solid blue;  
  width: 100px;  
  height: 100px;  
  margin: 20px;  
}
```

Using box-shadow:

```
.div2 {  
  border: 5px solid green;  
  box-shadow: 0px 0px 0px 4px #000;  
  width: 100px;  
  height: 100px;  
  margin: 20px;  
}
```

Using a pseudo element:

```
.div3 {  
  position: relative;  
  border: 5px solid #000;  
  width: 100px;  
  height: 100px;  
  margin: 20px;  
}  
.div3:before {  
  content: " ";  
  position: absolute;  
  border: 5px solid blue;  
  z-index: -1;  
  top: 5px;  
  left: 5px;  
  right: 5px;  
  bottom: 5px;  
}
```



<http://jsfiddle.net/MadalinaTn/bvqpcohm/2/>

## Section 10.4: border (shorthands)

In most cases you want to define several border properties (`border-width`, `border-style` and `border-color`) for all sides of an element.

Instead of writing:

```
border-width: 1px;  
border-style: solid;  
border-color: #000;
```

You can simply write:

```
border: 1px solid #000;
```

These shorthands are also available for every side of an element: `border-top`, `border-left`, `border-right` and `border-bottom`. So you can do:

```
border-top: 2px double #aaaaaa;
```

## Section 10.5: border-collapse

The `border-collapse` property applies only to `tables` (and elements displayed as `display: table` or `inline-table`) and sets whether the table borders are collapsed into a single border or detached as in standard HTML.

```
table {  
  border-collapse: separate; /* default */  
  border-spacing: 2px; /* Only works if border-collapse is separate */  
}
```

Also see Tables - border-collapse documentation entry

## Section 10.6: border-image

With the `border-image` property you have the possibility to set an image to be used instead of normal border styles.

A `border-image` essentially consist of a

- `border-image-source`: The path to the image to be used
- `border-image-slice`: Specifies the offset that is used to divide the image into **nine regions** (four **corners**, four **edges** and a **middle**)
- `border-image-repeat`: Specifies how the images for the sides and the middle of the border image are scaled

Consider the following example whereas `border.png` is a image of 90x90 pixels:

```
border-image: url("border.png") 30 stretch;
```

The image will be split into nine regions with 30x30 pixels. The edges will be used as the corners of the border while the side will be used in between. If the element is higher / wider than 30px this part of the image will be **stretched**. The middle part of the image defaults to be transparent.

## Section 10.7: Creating a multi-colored border using border-image

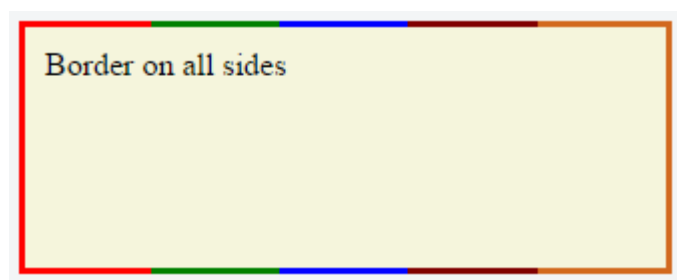
### CSS

```
.bordered {  
  border-image: linear-gradient(to right, red 20%, green 20%, green 40%, blue 40%, blue 60%, maroon 60%,  
  maroon 80%, chocolate 80%); /* gradient with required colors */  
  border-image-slice: 1;  
}
```

### HTML

```
<div class="bordered">Border on all sides</div>
```

The above example would produce a border that comprises of 5 different colors. The colors are defined through a linear-gradient (you can find more information about gradients in the docs). You can find more information about `border-image-slice` property in the `border-image` example in same page.

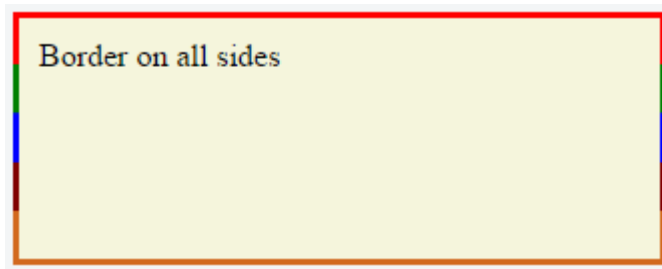


(Note: Additional properties were added to the element for presentational purpose.)

You'd have noticed that the left border has only a single color (the start color of the gradient) while the right border also has only a single color (the gradient's end color). This is because of the way that border image property works. It is as though the gradient is applied to the entire box and then the colors are masked from the padding and content areas, thus making it look as though only the border has the gradient.

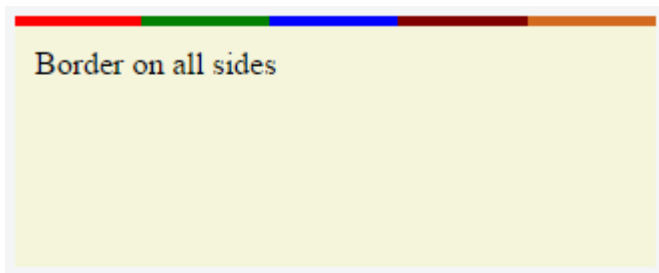
Which border(s) have a single color is dependant on the gradient definition. If the gradient is a `to right` gradient, the left border would be the start color of the gradient and right border would be the end color. If it was a `to bottom` gradient the top border would be the gradient's start color and bottom border would be end color. Below is

the output of a to `bottom` 5 colored gradient.



If the border is required only on specific sides of the element then the `border-width` property can be used just like with any other normal border. For example, adding the below code would produce a border only on the top of the element.

```
border-width: 5px 0px 0px 0px;
```



Note that, any element that has `border-image` property **won't respect the** `border-radius` (that is the border won't curve). This is based on the below statement in the spec:

A box's backgrounds, but not its border-image, are clipped to the appropriate curve (as determined by 'background-clip').

## Section 10.8: `border-[left | right | top | bottom]`

The `border-[left|right|top|bottom]` property is used to add a border to a specific side of an element.

For example if you wanted to add a border to the left side of an element, you could do:

```
#element {  
  border-left: 1px solid black;  
}
```