

5.4 Special effects

Text shadows

Shadows are added to text using the CSS property **text-shadow**, which accepts four values:

- **offset-x** - Horizontal pixel offset of shadow
- **offset-y** - Vertical pixel offset of shadow
- **blur-radius** - Optional shadow blur (default is 0)
- **color** - Optional shadow color (default is usually the current CSS color)

The figure below illustrates how the four property values change the shadow.

Figure 5.4.1: Examples of different text-shadow values.

```
<p style="text-shadow: 5px 5px">Example 1</p>
<p style="text-shadow: 5px 5px 1px;">Example 2</p>
<p style="text-shadow: -5px -5px 1px green;">Example 3</p>
<p style="text-shadow: 0 0 3px red;">Example 4</p>
<p style="text-shadow: 0 0 3px red, 0 0 6px purple;">Example
5</p>
```

Example 1

Example 2

Example 3

Example 4

Example 5

[Feedback?](#)

PARTICIPATION
ACTIVITY

5.4.1: Text shadows.



1) Positive `offset-x` and `offset-y` make the shadow appear to the right and below the text, but negative values make the shadow appear to the left and above the text.

- True
 False

2) The `offset-x` and `offset-y` must be a non-zero value.

- True
 False

3) A shadow with `blur-radius: 4px` is less blurry than a shadow with `blur-radius: 2px`.

- True
 False

4) Multiple shadows can apply to the same text.

- True
 False

Correct

Positive offset place shadows to the right and down, and negative offsets place shadows to the left and up.



Correct

When the offsets are 0, the shadow appears to make the text glow.



Correct

The larger the `blur-radius`, the more blurry the shadow. Negative values are not allowed.



Correct

The CSS below shows two shadows applied to the same text.

```
<p style="text-shadow: 0 0 3px red, 0 0 6px purple;">Two shadows</p>
```



[Feedback?](#)

Box shadows

The CSS property **box-shadow** adds a shadow to the box around an element using the following properties:

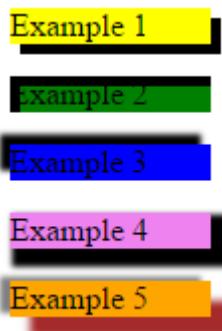
- `inset` - Optional value that draws the shadow inside the box (default is outside the box)
- `offset-x` - Horizontal pixel offset of shadow
- `offset-y` - Vertical pixel offset of shadow
- `blur-radius` - Optional shadow blur (default is 0)

- **spread-radius** - Positive value causes shadow to grow, negative values to shrink (default is 0)
- **color** - Optional shadow color (default is usually the current CSS color)

Figure 5.4.2: Examples of different box-shadow values.

```
p {  
    width: 100px;  
}  
#example1 {  
    background-color: yellow;  
    box-shadow: 5px 5px;  
}  
#example2 {  
    background-color: green;  
    box-shadow: inset 5px 5px;  
}  
#example3 {  
    background-color: blue;  
    box-shadow: -5px -5px 3px;  
}  
#example4 {  
    background-color: violet;  
    box-shadow: 5px 5px 3px 4px;  
}  
#example5 {  
    background-color: orange;  
    box-shadow: -5px -2px 3px gray, 10px 10px 5px brown;  
}
```

```
<p id="example1">Example 1</p>  
<p id="example2">Example 2</p>  
<p id="example3">Example 3</p>  
<p id="example4">Example 4</p>  
<p id="example5">Example 5</p>
```



[Feedback?](#)

PARTICIPATION
ACTIVITY

5.4.2: Box shadows.



1) The `box-shadow` property creates a shadow for text.

- True
- False

2) If the `box-shadow` uses the value `inset`, then the shadow appears inside the box.

- True
- False

3) A zero `spread-radius` makes the shadow the same size as the box.

- True
- False

Correct

`box-shadow` creates shadows for the box around an element. `text-shadow` creates a shadow for text.



Correct

An inset shadow makes the box appear to be pressed, but the default shadow makes the box appear raised.



Correct

Positive `spread-radius` values make the shadow grow bigger, and negative values make the shadow grow smaller.



[Feedback?](#)

PARTICIPATION ACTIVITY

5.4.3: Shadow practice.



The webpage below displays three flash cards with web history questions and answers. Modify the CSS to add shadows to the cards, question text, and answer text. Make the shadows use different colors, offsets, and blur radiiuses.

[HTML](#) [CSS](#)

```
1 .card {
2     width: 300px;
3     background-color: #eee;
4     border: solid 1px black;
5     padding: 10px;
6     margin-bottom: 10px;
7     box-shadow: 5px 3px 5px blue;
8 }
9
10 .question {
11     box-shadow: 2px 2px 7px green;
12 }
13
14 .answer {
15     box-shadow: 6px 6px 10px red;
16 }
```

Render webpageReset code

Your webpage

Q: Who invented the WWW?

A: Tim Berners-Lee

Q: When was the first website published?

A: 1991

Q: What web browser did most people use in the early 2000s?

▼ View solution

 Explain

--- START FILE: CSS ---

```
.card {
    width: 300px;
    background-color: #eee;
    border: solid 1px black;
    padding: 10px;
    margin-bottom: 10px;
    box-shadow: 5px 3px 5px blue;
```

}

```
.question {  
    box-shadow: 2px 2px 7px green;  
}  
  
.answer {  
    box-shadow: 6px 6px 10px red;  
}
```

--- END FILE: CSS ---

[Feedback?](#)

Rounded corners

An element border's corners can be rounded using the CSS property **border-radius**, which is assigned one to four radius values.

- Single value - All four corners are equally rounded
- Two values - First value is top-left and bottom-right corners, second value is top-right and bottom-left corners
- Three values - First value is top-left, second is top-right and bottom-left, third is bottom-right
- Four values - First value is top-left, second is top-right, third is bottom-right, fourth is bottom-left

Each corner may also be assigned a radius using four CSS properties:

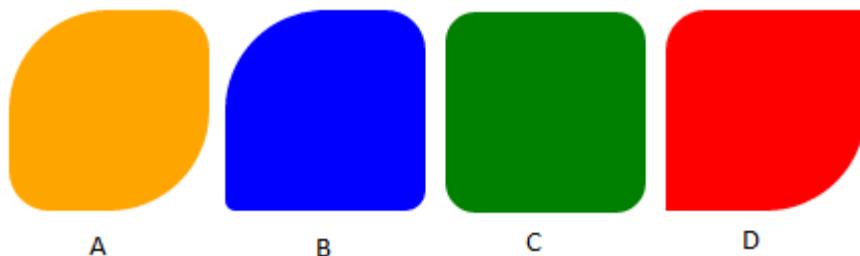
- `border-top-left-radius`
- `border-top-right-radius`
- `border-bottom-left-radius`
- `border-bottom-right-radius`

PARTICIPATION
ACTIVITY

5.4.4: Rounded corners.



Match the square with the CSS that produces the square's rounded corners.



If unable to drag and drop, refresh the page.

Bborder-radius: 40px 20px 10px 5px;

Top-left corner is 40px radius, top-right is 20px, bottom-right is 10px, and bottom-left is 5px.

Aborder-radius: 40px 20px;

Top-left and bottom-right corners are 40px radius and top-right and bottom-left corners are 20px radius.

Correct**D**border-top-left-radius: 20px;
border-bottom-right-radius: 50px;

Top-left corner is 20px radius and bottom-right is 50px radius.

Cborder-radius: 15px;

All corners are 15px radius.

Correct**Reset****Feedback?**

Border images

The CSS property **border-image** renders an element's border using sections of an image. The border image takes the place of any border properties specified by **border-style**. The following CSS properties are specified by **border-image** all at once:

- **border-image-source** - Image URL
- **border-image-slice** - Image section size
- **border-image-repeat** - "repeat" to repeat the image section, "round" to repeat the image section but resize the image if needed to fit, or "stretch" to stretch an image section

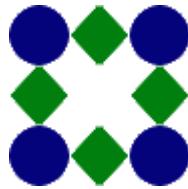
A border image will not appear unless the **border-width** property is non-zero and **border-style** is set to any legal value except **none** and **hidden**.

PARTICIPATION ACTIVITY

5.4.5: Try different border-image values.



The borderv2.png image is used to display a border image around the **<div>** in the webpage below. The blue circles and green diamonds in the image are each 30×30 pixels.



Change the following CSS property values to see the effect on the border image:

1. Change the image slice size from 30 to 15 in the CSS property **border-image**. Render the page and observe how half the circle and half the diamond is used to render the border.
2. Change the **border-image** slice size from 15 to 60. Render the page and observe how a 60×60 pixel section ($2/3$ of the image) is used to render the border corners. Since borderv2.png is only 90×90 pixels, an unused 60×60 pixel section does not exist, so the border sides are empty.
3. Change the **border** size from 15px to 30px. Render the webpage and observe how the border size increased.
4. Change the **border-image** slice size back to 30, and change "round" to "repeat". Render the page and observe how the green diamonds are repeated but do not fit perfectly on the left and right sides.
5. Finally, change the **border-image** "repeat" to "stretch". Render the page and observe how the green diamonds stretch to fill the border.

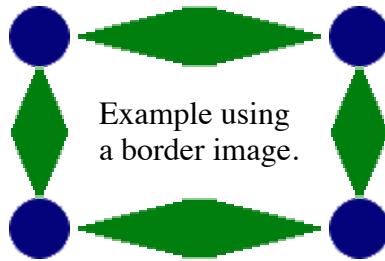
[HTML](#) [CSS](#)

```
1 #example {  
2     width: 100px;  
3     border: 30px solid;  
4     padding: 15px;  
5     border-image: url(https://static-resources.zybooks.com/static/)  
6 }  
7
```

[Render webpage](#)

[Reset code](#)

Your webpage



Example using
a border image.

▼ View solution

Explain

--- START FILE: CSS ---

```
#example {  
    width: 100px;  
    border: 30px solid;  
    padding: 15px;  
    border-image: url(https://static-resources.zybooks.com/static/borderv2.png) 30 stretch;  
}
```

--- END FILE: CSS ---

[Feedback?](#)

PARTICIPATION
ACTIVITY

5.4.6: Border images.



Refer to the CSS below.

```
border-image: url(some-border.png) 50 repeat;
```

1) 50×50 pixel sections of some-border.png are used to create the border image.

- True
- False

2) If some-border.png is 50×50 pixels, then the border will have empty sides.

- True
- False

3) If some-border.png is 150×150 pixels, the border image section is stretched on the sides.

- True
- False

4) To display the 50×50 pixel sections evenly around an element's border image, the element's border should be specified as:

```
border: 50  
solid;
```

- True
- False

Correct

A single number 50 is the size of the image section used to create a border image.



Correct

Using the image's width for **border-image-slice** shows the entire image in the border corners but does not leave any portion of the image to be used to display the border sides.



Correct

"repeat" repeats the image section. "stretch" stretches the image section.



Correct

The **border** size should be 50px, not 50. Although **border-image-slice** does not specify px, the border size must specify a measurement unit like px.



[Feedback?](#)

CSS3 browser support

Most modern browsers support CSS3, but some CSS3 properties require vendor prefixes to work on certain browsers. A **vendor prefix** is a prefix added to an experimental or nonstandard CSS property that only works on a specific browser type. Typical vendor prefixes are:

- **-webkit-** for Chrome, Safari, and newer versions of Opera
- **-moz-** for Firefox

- **-ms-** for Internet Explorer
- **-o-** for older versions of Opera

The following CSS specifies a ***border-image*** property for WebKit and Opera browsers:

```
#borderimg {
    -webkit-border-image: url(border.png) 30 round; /* Safari
3.1-5 */
    -o-border-image: url(border.png) 30 round; /* Opera 11-12.1
*/
    border-image: url(border.png) 30 round;
}
```

Linear gradients

A CSS background may use gradient colors that transition from one color to another. Two CSS gradients exist:

1. Linear gradient - A gradient that follows a straight line
2. Radial gradient - A gradient that radiates outward into an ellipse

The CSS function ***linear-gradient(color1, color2)*** creates a linear gradient that transitions from **color1** to **color2** when moving from the top edge to the bottom edge. Additional colors can be supplied to the function. Ex:

`linear-gradient(red, green, blue, yellow)` transitions from red to green to blue to yellow when moving from top to bottom.

To change the gradient's direction, the first argument to ***linear-gradient*** can be a direction or an angle:

- **Direction** - A direction of **left**, **right**, **top**, or **bottom** with the word **to** in front. Ex:
to left creates a linear gradient that moves from right to left, and
to bottom right goes from the top-left corner to the bottom-right corner.
- **Angle** - A CSS angle that points in the direction of the linear gradient. The angles **0deg**, **90deg**, **180deg**, and **270deg** correspond to **to top**, **to right**, **to bottom**, and **to left**, respectively.

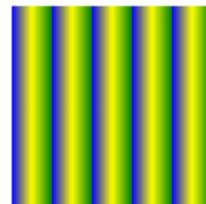
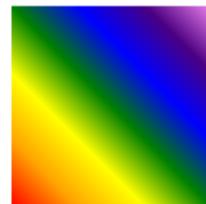
The ***repeating-linear-gradient()*** function repeats a linear gradient where the color values are supplied an optional percent. The percentage value after the last color is the percent of the gradient's total length the repeating gradient should occupy. Ex:

`repeating-linear-gradient(red, yellow 10%)` means the red to yellow gradient occupies 10% of the gradient's total length and is repeated to fill the entire background.

Figure 5.4.3: Examples of linear gradients.

```
#example1 {  
    background: linear-gradient(green, yellow);  
}  
#example2 {  
    background: linear-gradient(to bottom right, black, red);  
}  
#example3 {  
    background: linear-gradient(45deg, red, orange, yellow, green,  
        blue, indigo, violet);  
}  
#example4 {  
    background: repeating-linear-gradient(to right, blue, yellow,  
        green 20%);  
}
```

```
<div id="example1"></div>  
<div id="example2"></div>  
<div id="example3"></div>  
<div id="example4"></div>
```



[Feedback?](#)

PARTICIPATION
ACTIVITY

5.4.7: Linear gradients.



- 1) What direction creates the gradient below?

Correct



```
background: linear-  
gradient(_____,  
orange, red);
```

to right

The gradient is orange on the left edge and becomes red when moving to the right.



to right

[Check](#)

[Show answer](#)

- 2) What angle (direction of red arrow) creates the gradient below?

Correct



225deg

```
background: linear-
gradient(_____, blue,
green);
```

The 225 degree linear gradient goes from blue in the top-right corner to green as the gradient moves down and to the left.

**Check****Show answer**

- 3) What color and percent creates the repeating linear gradient that ends in white?

```
background: repeating-
linear-gradient(black,
_____);
```

Correct

white 25%

The black to white gradient repeats 4 times, so each gradient is 25% of the gradient's total length.



white 25%

Check**Show answer****Feedback?**

Radial gradients

A radial gradient is created with the CSS function **radial-gradient(color1, color2)**, which creates an ellipse-shaped gradient that begins with **color1** in the center and ends with **color2** on the perimeter. More than two colors may be specified. A percentage or length can be placed after a color to give more emphasis to the color. Ex:

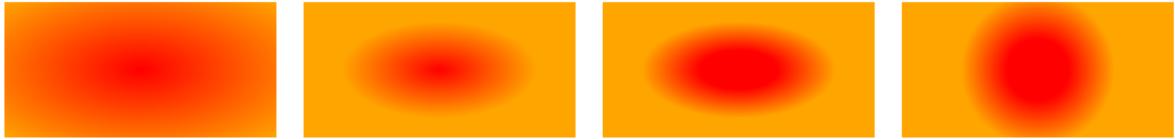
radial-gradient(red 10%, yellow 30%) gives more emphasis to red and yellow than the default rendering.

The ellipse shape of a radial gradient fits the gradient's bounding rectangle. However, a circular radial gradient can be created with the **circle** argument. Ex:

radial-gradient(circle, red, yellow) creates a circle gradient.

Figure 5.4.4: Examples of radial gradients.

```
#example1 {  
    background: radial-gradient(red, orange);  
}  
#example2 {  
    background: radial-gradient(red, orange 50%);  
}  
#example3 {  
    background: radial-gradient(red 20%, orange 50%);  
}  
#example4 {  
    background: radial-gradient(circle, red 20%, orange 50%);  
}
```

[Feedback?](#)**PARTICIPATION ACTIVITY****5.4.8: Radial gradient.**

- 1) A radial gradient is always an ellipse or circle.

True
 False

Correct

No other shapes are supported.

- 2) The radial gradient below has a blue interior and a green exterior.

Correct

The interior is green and the exterior blue.



```
radial-  
gradient(green,  
blue);
```

True
 False



3) What arguments to `radial-gradient()` create the radial gradient below?



Correct

The black interior is made larger with the "40%".

- black, white, red
- black 40%, white, red

[Feedback?](#)

Positioning radial gradients

A radial gradient's ellipse or circle is centered by default in the enclosing rectangle, but the center position can be specified using "`at centerX centerY`" where `centerX` and `centerY` specify a distance or percentage. Ex:

`radial-gradient(at 50px 10px, yellow, green)` specifies a center that is 50px from the left edge and 10px from the top.

By default, a radial gradient's shape reaches to the farthest corner of the containing rectangle. An extent keyword describes the size of the radial gradient's shape:

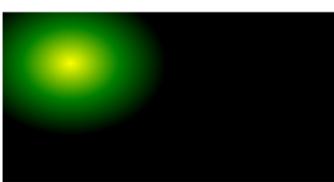
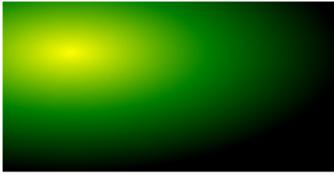
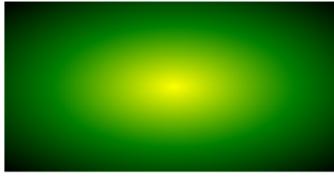
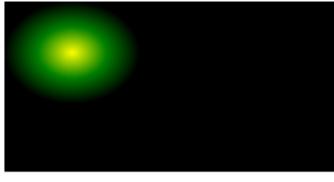
- **closest-side** - Circle touches the rectangle's side closest to the circle's center.
Ellipse touches the vertical and horizontal sides closest to the ellipse's center.
- **farthest-side** - Circle touches the rectangle's side farthest from the circle's center.
Ellipse touches the vertical and horizontal sides farthest from the ellipse's center.
- **closest-corner** - Circle or ellipse touches the corner closest to the shape's center.
- **farthest-corner** - Circle or ellipse touches the corner farthest from the shape's center. (Default behavior.)

PARTICIPATION ACTIVITY

5.4.9: Positioned radial gradients.



Match the background with the radial gradient CSS that produced the background.

- | | | |
|--|--|---|
| A.  | B.  | C.  |
| D.  | E.  | F.  |

If unable to drag and drop, refresh the page.

C	<p><code>radial-gradient(at 60px 100px, yellow, green, black)</code></p> <p>The ellipse is centered at (60,100) and touches the top-right corner.</p>	Correct
E	<p><code>radial-gradient(yellow, green, black)</code></p> <p>The ellipse is centered in the rectangle and touches all four sides.</p>	Correct
A	<p><code>radial-gradient(closest-corner at 20% 30%, yellow, green, black)</code></p> <p>The ellipse is centered at 20% of the rectangle's width and 30% of the rectangle's height and touches the top-left corner, which is closest to the ellipse's center.</p>	Correct
D	<p><code>radial-gradient(farthest-side at 20% 30%, yellow, green, black)</code></p> <p>The ellipse is centered at 20% of the rectangle's width and 30% of the rectangle's height and touches the right and bottom edges, which are furthest from the ellipse's center.</p>	Correct
F	<p><code>radial-gradient(closest-side at 20% 30%, yellow, green, black)</code></p> <p>The ellipse is centered at 20% of the rectangle's width and 30% of the rectangle's height and touches the left and top edges, which are closest to the ellipse's center.</p>	Correct
	<p><code>radial-gradient(farthest-corner at 20% 30%, yellow, green, black)</code></p>	Correct

B

The ellipse is centered at 20% of the rectangle's width and 30% of the rectangle's height and touches the bottom-right corner, which is farthest from the ellipse's center.

Reset**Feedback?****PARTICIPATION
ACTIVITY**

5.4.10: Gradient practice.



The webpage below displays an advertisement with a background produced by the CSS function `repeating-radial-gradient()`. Make the following modifications to the HTML and CSS so the rendered webpage resembles the expected webpage:

1. Add a radial gradient background to the `<body>` using any colors you prefer, and position the ellipse close to the bottom-right corner.
2. Create two more advertisements like the ads in the expected webpage. Choose whatever fonts and colors you prefer. One ad should have a linear gradient background and the other a repeating linear gradient background.

HTML**CSS**

```
1 <div id="ad1">Vote this Tuesday!</div>
```

```
2
```

Render webpage**Reset code**

Your webpage**Expected webpage**

Vote this Tuesday!

Vote this Tuesday!

Buy some chocolate!

UPGRADE YOUR PHONE!

▼ View solution

 Explain

--- START FILE: HTML ---

```
<div id="ad1">Vote this Tuesday!</div>
<div id="ad2">Buy some chocolate!</div>
<div id="ad3">Upgrade your phone!</div>
```

--- END FILE: HTML ---

--- START FILE: CSS ---

```
div {
    text-align: center;
    width: 350px;
    height: 60px;
    line-height: 60px;
    font-weight: bold;
    margin-bottom: 10px;
}

body {
    background: radial-gradient(at 80% 80%, gray, white);
}

#ad1 {
    font-size: 28pt;
    background: repeating-radial-gradient(pink, yellow,
```

```
lightblue 15%);  
}  
  
#ad2 {  
    font-size: 18pt;  
    font-family: arial;  
    font-style: italic;  
    color: white;  
    background: linear-gradient(to right, red, black);  
}  
  
#ad3 {  
    font-size: 24pt;  
    font-family: tahoma;  
    font-variant: small-caps;  
    color: black;  
    background: repeating-linear-gradient(to top left,  
    DodgerBlue, Gainsboro 10%);  
}
```

--- END FILE: CSS ---

[Feedback?](#)

CHALLENGE
ACTIVITY

5.4.1: Special effects.



530096.4000608.qx3zqy7

Start



1



2



3



4

CSS HTML

```
1 p {  
2     /* Your solution goes here */  
3  
4  
5 }
```

1

2

3

4

[Check](#)[Next](#)[View your last submission](#) ▾[Feedback?](#)

Exploring further:

- [CSS3 text-shadow Property](#) from W3Schools
- [CSS3 box-shadow Property](#) from W3Schools
- [CSS3 Rounded Corners](#) from W3Schools
- [CSS3 border-image Property](#) from W3Schools
- [CSS3 Gradients](#) from W3Schools

How was
this
section?

[Provide section feedback](#)