



# CSS – FORMATTING TEXT, BACKGROUND AND ANIMATION

---

**SUBJECT: BASICS OF WEB DESIGN  
(09CE2102)**

**DEPARTMENT: CE-DIPLO**

**BY: ZEEL MEHTA**

---

➤ **CSS Gradients**

➤ **Text effect**

➤ **Text-Transformations**

➤ **Shadow Effects**

➤ **Animations**

# CSS Gradients

---

- CSS gradients let you display smooth transitions between two or more specified colors.
- CSS defines three types of gradients:
  - Linear Gradients (goes down/up/left/right/diagonally)
  - Radial Gradients (defined by their center)
  - Conic Gradients (rotated around a center point)



# CSS Linear Gradients

---

To create a linear gradient you must define at least two color stops. Color stops are the colors you want to render smooth transitions among. You can also set a starting point and a direction (or an angle) along with the gradient effect.

## **Syntax:**

`background-image: linear-gradient(direction, color-stop1, color-stop2, ...);`

## **Direction - Top to Bottom (this is default)**

The following example shows a linear gradient that starts at the top. It starts red, transitioning to yellow:

### **Example:**

```
#grad {  
  background-image: linear-gradient(red, yellow);  
}
```

top to bottom (default)



---

## Direction - Left to Right

The following example shows a linear gradient that starts from the left. It starts red, transitioning to yellow:

Example

```
#grad {  
  background-image: linear-gradient(to right, red , yellow);  
}
```



---

## Direction - Diagonal

You can make a gradient diagonally by specifying both the horizontal and vertical starting positions.

The following example shows a linear gradient that starts at top left (and goes to bottom right). It starts red, transitioning to yellow:

### Example

```
#grad {  
  background-image: linear-gradient(to bottom right, red, yellow);  
}
```

top left to bottom right



---

## Using Angles

If you want more control over the direction of the gradient, you can define an angle, instead of the predefined directions (to bottom, to top, to right, to left, to bottom right, etc.). A value of 0deg is equivalent to "to top". A value of 90deg is equivalent to "to right". A value of 180deg is equivalent to "to bottom".

### Syntax:

background-image: linear-gradient(angle, color-stop1, color-stop2);

### Example

180deg

```
#grad {  
  background-image: linear-gradient(180deg, red, yellow);  
}
```

---

## Using Multiple Color Stops

The following example shows a linear gradient (from top to bottom) with multiple color stops:



```
#grad {  
  background-image: linear-gradient(red, yellow, green);  
}
```



```
#grad {  
  background-image: linear-gradient(to right,  
red,orange,yellow,green,blue,indigo,violet);  
}
```

# Rainbow Background



# CSS Radial Gradients

---

A radial gradient is defined by its center.

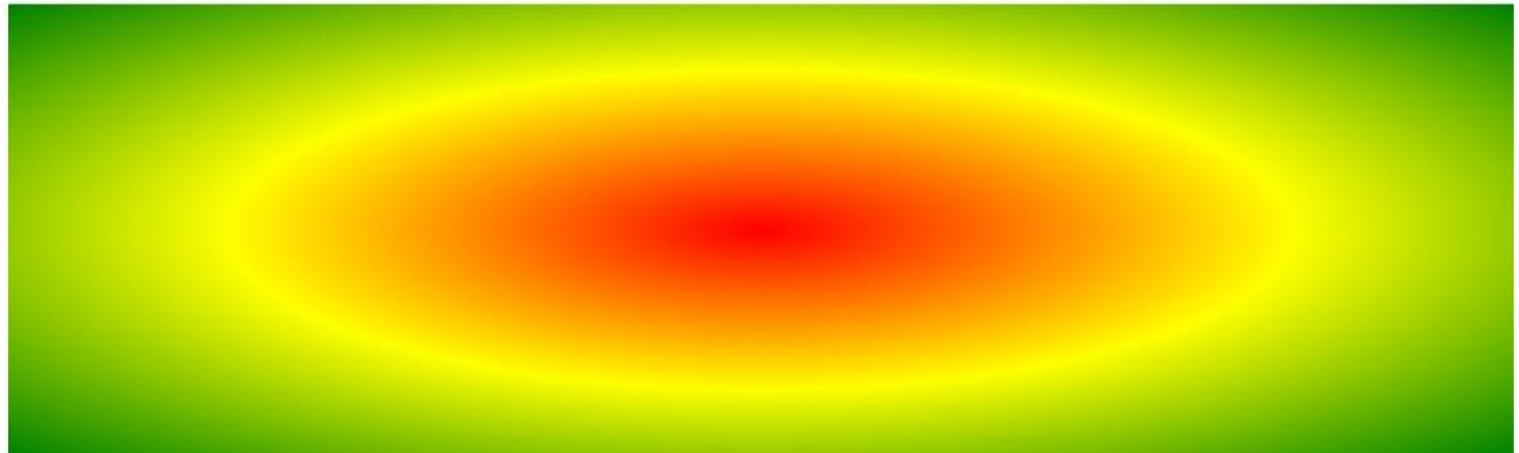
To create a radial gradient you must also define at least two color stops.

**Syntax:**

background-image: radial-gradient(shape size at position, start-color, ..., last-color);

Example

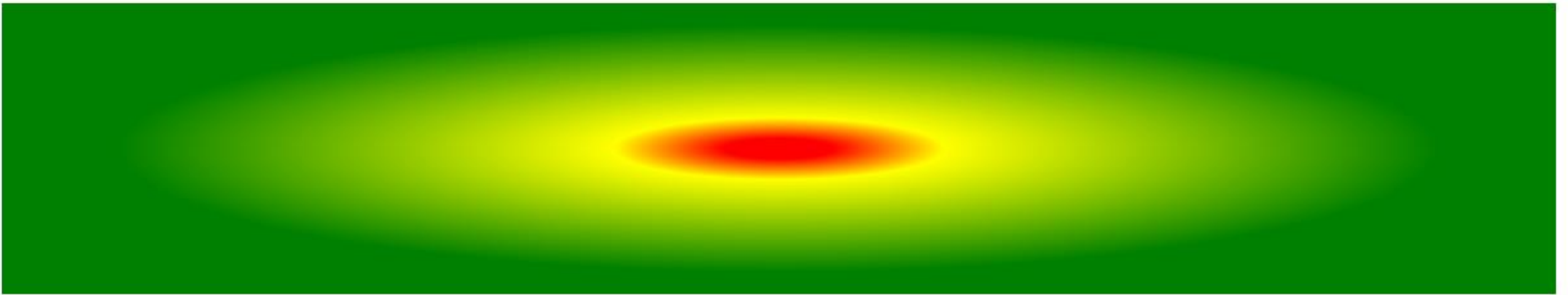
```
#grad {  
  background-image:  
  radial-gradient(red, yellow,  
  green);  
}
```



---

## Radial Gradient - Differently Spaced Color Stops

```
#grad {  
  background-image: radial-gradient(red 5%, yellow 15%, green  
60%);  
}
```



---

## Set Shape:

The shape parameter defines the shape. It can take the value circle or ellipse. The default value is ellipse.

The following example shows a radial gradient with the shape of a circle:

### Example:

```
#grad {  
  background-image: radial-gradient(circle,  
    red, yellow, green);  
}
```



# CSS Conic Gradients

---

A conic gradient is a gradient with color transitions rotated around a center point.

To create a conic gradient you must define at least two colors.

**Syntax:**

```
background-image: conic-gradient([from angle] [at position,]  
color [degree], color [degree], ...);
```

By default, angle is 0deg and position is center.

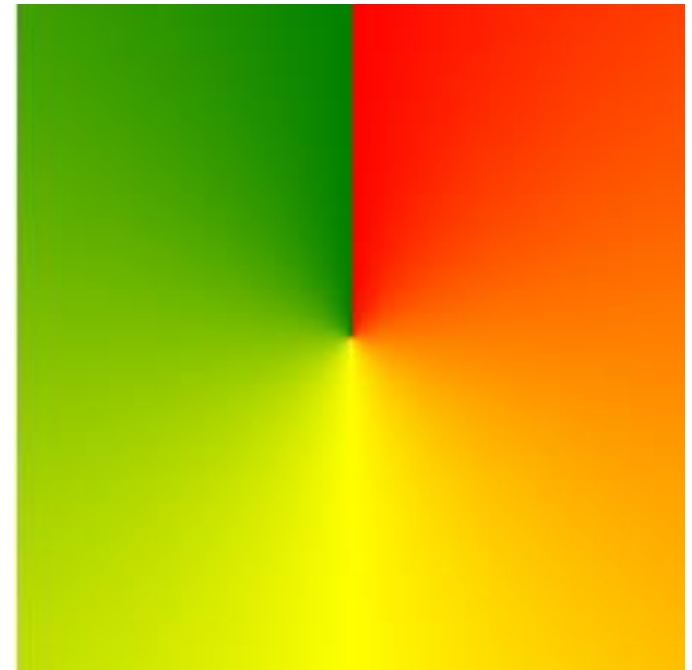
If no degree is specified, the colors will be spread equally around the center point.

---

## Conic Gradient: Three Colors

The following example shows a conic gradient with three colors:

```
#grad {  
  background-image: conic-  
gradient(red, yellow, green);  
}
```



---

## Conic Gradient: Colors and Degrees

The following example shows a conic gradient with colors and a degree for each color:

```
#grad {  
  background-image: conic-gradient(red  
45deg, yellow 90deg, green 210deg);  
}
```

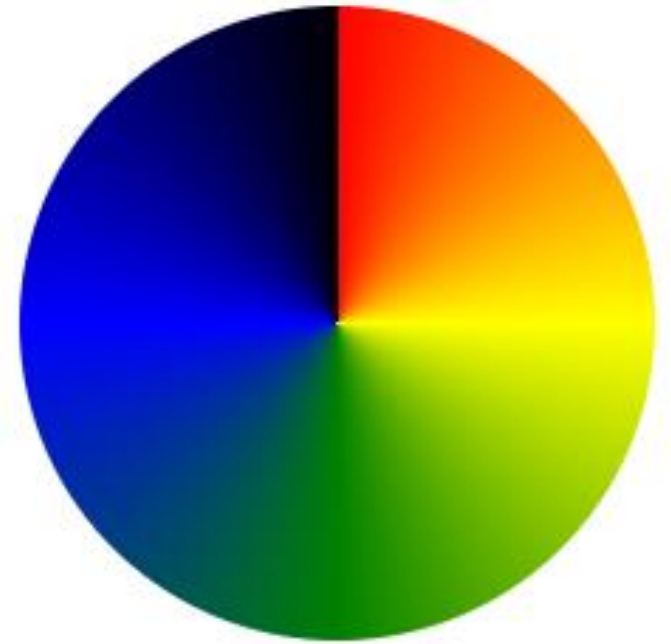


---

## Create Pie Charts

Just add border-radius: 50% to make the conic gradient look like a pie:

```
#grad {  
  background-image: conic-  
gradient(red, yellow, green, blue,  
black);  
  border-radius: 50%;  
}
```

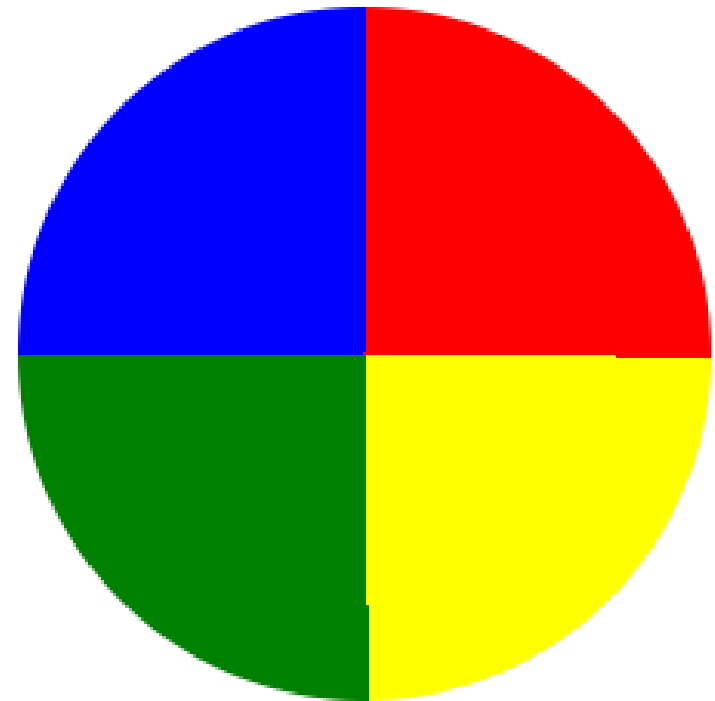


---

Here is another pie chart with defined degrees for all the colors:

**Example**

```
#grad {  
  background-image: conic-  
gradient(red 0deg, red 90deg,  
yellow 90deg, yellow 180deg,  
green 180deg, green 270deg, blue  
270deg);  
  border-radius: 50%;  
}
```





# Text effect

---

In this chapter you will learn about the following properties:

- text-overflow
- word-wrap
- word-break
- writing-mode

---

## CSS Text Overflow:

The CSS text-overflow property specifies how overflowed content that is not displayed should be signaled to the user.

### **It can be clipped:**

This is some long text that

Or

### **it can be rendered as an ellipsis (...):**

This is some long text t...

The CSS code is as follows:

```
p.test1 {  
  white-space: nowrap;
```

```
  width: 200px;  
  border: 1px solid #000000;  
  overflow: hidden;  
  text-overflow: clip;  
}
```

```
p.test2 {  
  white-space: nowrap;  
  width: 200px;  
  border: 1px solid #000000;  
  overflow: hidden;  
  text-overflow: ellipsis;  
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
p.test1 { white-space: nowrap; width: 200px; border: 1px solid #000000;
  overflow: hidden; text-overflow: clip;}
p.test2 { white-space: nowrap; width: 200px; border: 1px solid #000000;
  overflow: hidden; text-overflow: ellipsis;}
</style>
</head>
<body>
<h1>The text-overflow Property</h1>
<p>The following two paragraphs contains a long text that will not fit in
the box.</p>
<h2>text-overflow: clip:</h2>
<p class="test1">This is some long text that will not fit in the box</p>
<h2>text-overflow: ellipsis:</h2>
<p class="test2">This is some long text that will not fit in the box</p>
</body>
</html>
```

## The text-overflow Property

The following two paragraphs contains a long text that will not fit in the box.

### text-overflow: clip:

This is some long text that will

### text-overflow: ellipsis:

This is some long text that ...

---

The following example shows how you can display the overflowed content when hovering over the element:

### Example

`div.test:hover { overflow: visible; }`

```
<!DOCTYPE html>
<html>
<head>
<style>
div.test {white-space: nowrap; width: 200px; overflow: hidden;
  border: 1px solid #000000;}
div.test:hover { overflow: visible;}
</style>
</head>
<body>
<p>Hover over the two divs below, to see the entire text.</p>
<div class="test" style="text-overflow:ellipsis;">This is some long text
that will not fit in the box</div>
<br>
<div class="test" style="text-overflow:clip;">This is some long text that
will not fit in the box</div>
</body>
</html>
```

Hover over the two divs below, to see the entire text.

This is some long text that will not fit in the box

This is some long text that will

```
<!DOCTYPE html>
<html>
<head>
<style>
div.test {white-space: nowrap; width: 200px; overflow: hidden;
  border: 1px solid #000000;}
div.test:hover { overflow: visible;}
</style>
</head>
<body>
<p>Hover over the two divs below, to see the entire text.</p>
<div class="test" style="text-overflow:ellipsis;">This is some long text
that will not fit in the box</div>
<br>
<div class="test" style="text-overflow:clip;">This is some long text that
will not fit in the box</div>
</body>
</html>
```

Hover over the two divs below, to see the entire text.

This is some long text that ...

This is some long text that will not fit in the box

---

## CSS Word Wrapping

The CSS word-wrap property allows long words to be able to be broken and wrap onto the next line.

If a word is too long to fit within an area, it expands outside:

This paragraph  
contains a very long  
word:

thisisaveryveryveryveryveryveryverylongword. wrap onto the next line:

The long word will  
break and wrap to  
the next line.

### Example

Allow long words to be  
able to be broken and

wrap onto the next line:

```
p { word-wrap: break-  
word; }
```

```
<!DOCTYPE html>
<html>
<head>
<style>
p.test {width: 11em; border: 1px solid
#000000; word-wrap: break-word;}
</style>
</head>
<body>
<h1>The word-wrap Property</h1>
<p class="test">This paragraph contains a
very long word:
thisisaveryveryveryveryveryverylongword.
The long word will break and wrap to the
next line.</p>
</body>
</html>
```

## The word-wrap Property

This paragraph contains a very long word:  
thisisaveryveryveryveryveryveryveryveryveryveryverylongword. The long word will break and wrap to the next line.


---

## CSS Word Breaking

The CSS word-break property specifies line breaking rules.


This paragraph contains some text. This line will break at hyphens.

```
p.test1 {  
  word-  
break: keep-all;  
}
```



This paragraph contains some text. The lines will break at any character.

```
p.test2 {  
  word-  
break: break-all;  
}
```





```
<!DOCTYPE html>
<html>
<head>
<style>
p.test1 {width: 30px; border: 1px solid
#000000; word-break: keep-all;}
p.test2 {width: 100px; border: 1px solid
#000000; word-break: break-all;}
</style>
</head>
<body>
<h1>The word-break Property</h1>
<p class="test1">This paragraph contains
some text. This line will-break-at-
hyphens.</p>
<p class="test2">This paragraph contains
some text. The lines will break at any
character.</p>
</body>
</html>
```

## The word-break Property

This  
paragraph  
contains  
some  
text.  
This  
line  
will-  
break-  
at-  
hyphens.

This paragraph  
contains some t  
ext. The lines  
will break at an  
y character.

---

## CSS Writing Mode

The CSS writing-mode property specifies whether lines of text are laid out horizontally or vertically.

Some text with a span element with a vertical-rl writing-mode.

The following example shows some different writing modes:

### Example

```
p.test1 { writing-mode: horizontal-tb; }
```

```
span.test2 { writing-mode: vertical-rl; }
```

```
p.test2 { writing-mode: vertical-rl; }
```

```

<!DOCTYPE html>
<html>
<head>
<style>
p.test1 { writing-mode: horizontal-tb;}
span.test2 { writing-mode: vertical-rl;}
p.test2 { writing-mode: vertical-rl;}
</style>
</head>
<body>
<h1>The writing-mode Property</h1>
<p class="test1">Some text with default writing-
mode.</p>
<p>Some text with a span element with a <span
class="test2">vertical-rl</span> writing-
mode.</p>
<p class="test2">Some text with writing-mode:
vertical-rl.</p>
</body>
</html>

```

## The writing-mode Property

Some text with default writing-mode.

Some text with a span element with a vertical-rl writing-mode.

Some text with writing-mode: v

# Text Transformation

---

The **text-transform** property is used to specify uppercase and lowercase letters in a text.

It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word:

## Example

**p.uppercase** { **text-transform**: uppercase; }

THIS TEXT IS TRANSFORMED TO UPPERCASE.

**p.lowercase** { **text-transform**: lowercase; }

this text is transformed to lowercase.

**p.capitalize** { **text-transform**: capitalize; }

This Text Is Capitalized.

# CSS Shadow Effects

---

## **CSS Shadow Effects**

With CSS you can add shadow to text and to elements.

In these chapters you will learn about the following properties:

- text-shadow

- box-shadow

---

## CSS Text Shadow

The CSS text-shadow property applies shadow to text.

In its simplest use, you only specify the horizontal shadow (2px) and the vertical shadow (2px):

### Example

```
h1 {  
  text-shadow: 2px 2px;  
}
```

————→ **Text shadow effect!**

add a color to the shadow:

### Example

```
h1 {  
  text-shadow: 2px 2px red;  
}
```

————→ **Text shadow effect!**

---

Add a blur effect to the shadow:

**Example**

```
h1 { text-shadow: 2px 2px 5px red; }
```

The following example shows a white text with black shadow:

**Example**

```
h1 { color: white; text-shadow: 2px 2px 4px #000000; }
```

The following example shows a red neon glow shadow:

**Example**

```
h1 { text-shadow: 0 0 3px #FF0000; }
```

→ Text shadow effect!

→ Text shadow effect!

→ Text shadow effect!

---

## Multiple Shadows

To add more than one shadow to the text, you can add a comma-separated list of shadows.

The following example shows a red and blue neon glow shadow:

### Example

```
h1 {  
  text-shadow: 0 0 3px #FF0000, 0 0 5px #0000FF;  
}
```

Text shadow effect!



---

## CSS box-shadow Property:

The CSS box-shadow property is used to apply one or more shadows to an element.

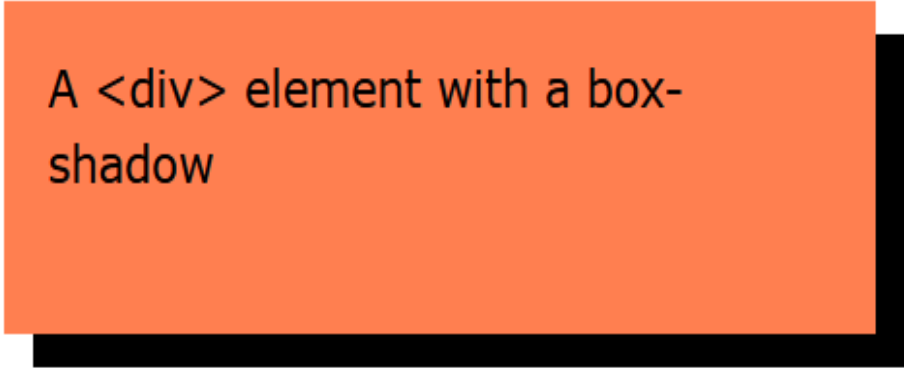
### Specify a Horizontal and a Vertical Shadow

In its simplest use, you only specify a horizontal and a vertical shadow. The default color of the shadow is the current text-color.

### Example

Specify a horizontal and a vertical shadow:

```
div {  
  box-shadow: 10px 10px ;  
}
```



A <div> element with a box-shadow

---

## Specify a Color for the Shadow

The color parameter defines the color of the shadow.

### Example

```
div {  
  box-shadow: 10px 10px lightblue;  
}
```



A <div> element with a lightblue box-shadow

### Add a Blur Effect to the Shadow

The blur parameter defines the blur radius.

The higher the number, the more blurred the shadow will be.

### Example

```
div {  
  box-shadow: 10px 10px 5px lightblue;  
}
```



A <div> element with a 5px blurred, lightblue box-shadow

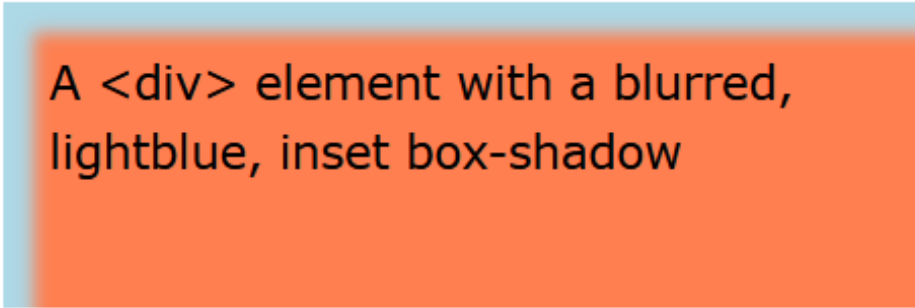
---

## Set the inset Parameter:

The inset parameter changes the shadow from an outer shadow (outset) to an inner shadow.

### Example

```
div { box-shadow: 10px 10px 5px  
  lightblue inset; }
```



A <div> element with a blurred, lightblue, inset box-shadow

### Add Multiple Shadows

An element can also have multiple shadows:

### Example

```
div { box-shadow: 5px 5px blue, 10px  
  10px red, 15px 15px green;}
```

# CSS Animations

---

CSS allows animation of HTML elements without using JavaScript or Flash!

In this chapter you will learn about the following properties:

- @keyframes
- animation-name
- animation-duration

---

## **What are CSS Animations?**

An animation lets an element gradually change from one style to another.

You can change as many CSS properties you want, as many times as you want.

To use CSS animation, you must first specify some keyframes for the animation.

Keyframes hold what styles the element will have at certain times.

---

# The `@keyframes` Rule

When you specify CSS styles inside the `@keyframes` rule, the animation will gradually change from the current style to the new style at certain times.

To get an animation to work, you must bind the animation to an element.

The following example binds the "example" animation to the `<div>` element.

The animation will last for 4 seconds, and it will gradually change the background-color of the `<div>` element from "red" to "yellow":

---

## Example

```
/* The animation code */
@keyframes example {
  from {background-color: red;}
  to {background-color: yellow;}
}

/* The element to apply the animation to */
div { width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
```

## Output:

[https://www.w3schools.com/css/tryit.asp?filename=trycss3\\_animation1](https://www.w3schools.com/css/tryit.asp?filename=trycss3_animation1)

**Note:** The **animation-duration** property defines how long an animation should take to complete. If the **animation-duration** property is not specified, no animation will occur, because the default value is 0s (0 seconds).

---

In the example above we have specified when the style will change by using the keywords "from" and "to" (which represents 0% (start) and 100% (complete)).

It is also possible to use percent. By using percent, you can add as many style changes as you like.

The following example will change the background-color of the <div> element when the animation is 25% complete, 50% complete, and again when the animation is 100% complete.



---

## Example:

```
/* The animation code */
@keyframes example {
  0% {background-color: red;}
  25% {background-color: yellow;}
  50% {background-color: blue;}
  100% {background-color: green;}
}
/* The element to apply the animation to */
div { width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
```

## Output:

[https://www.w3schools.com/css/tryit.asp?filename=trycss3\\_animation2](https://www.w3schools.com/css/tryit.asp?filename=trycss3_animation2)

---

The following example will change both the background-color and the position of the <div> element when the animation is 25% complete, 50% complete, and again when the animation is 100% complete:

```
/* The animation code */
```

```
@keyframes example {
```

```
  0% {background-color:red; left:0px; top:0px;}
```

```
  25% {background-color:yellow; left:200px; top:0px;}
```

```
  50% {background-color:blue; left:200px; top:200px;}
```

```
  75% {background-color:green; left:0px; top:200px;}
```

```
  100% {background-color:red; left:0px; top:0px;}
```

```
}
```

---

```
/* The element to apply the animation to */
div {
  width: 100px;
  height: 100px;
  position: relative;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
```

## Output:

[https://www.w3schools.com/css/tryit.asp?filename=trycss3\\_animation3](https://www.w3schools.com/css/tryit.asp?filename=trycss3_animation3)