

MUMBAI OF UNIVERSITY



CSS

BY:

PRIYUSH B.K.

NEW EDITION
2022

✔ What is CSS?

CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can also be used with any kind of XML documents including plain XML, SVG and XUL.

- CSS is used to control the style of a web document in a simple and easy way.
- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
- CSS is used to design HTML tags.
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files.

Why to Learn CSS?

- Create Stunning Web site
- Become a web designer.
- Control web
- Solves a big problem
- Saves a lot of time
- Learn other languages.

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is document title</title>
    <style>
      h1 {
        color: #36CFFF;
      }
    </style>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

Hello World!

✔ Applications of CSS

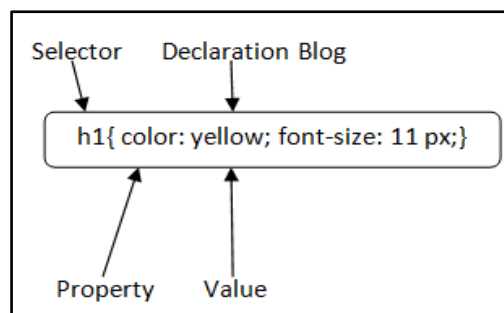
As mentioned before, CSS is one of the most widely used style language over the web. I'm going to list few of them here:

1. **CSS saves time** - You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.
2. **Pages load faster** - If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So, less code means faster download times.
3. **Easy maintenance** - To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
4. **Superior styles to HTML** - CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.



CSS Syntax Rule:

A CSS rule set contains a selector and a declaration block.



Selector: Selector indicates the HTML element you want to style. It could be any tag like `<h1>`, `<title>` etc.

Declaration Block: The declaration block can contain one or more declarations separated by a semicolon. For the above example, there are two declarations:

1. `color: yellow;`
2. `font-size: 11 px;`

Each declaration contains a property name and value, separated by a colon.

Property: A Property is a type of attribute of HTML element. It could be color, border etc.

Value: Values are assigned to CSS properties. In the above example, value "yellow" is assigned to color property.



How to add CSS:

CSS is added to HTML pages to format the document according to information in the style sheet. There are three ways to insert CSS in HTML documents.

- ✓ Inline CSS
- ✓ Internal CSS
- ✓ External CSS

Inline CSS:

- An inline style may be used to apply a unique style for a single element.
- The inline CSS is also a method to insert style sheets in HTML document.
- If you want to use inline CSS, you should use the style attribute to the relevant tag.



Syntax:

```
1. <htmltag style="cssproperty1:value; cssproperty2:value;"> </htmltag>
```

EX.

```
1. <!DOCTYPE html>
2. <html>
3. <body>
4. <h1 style="color:red;margin-left:40px;">Inline CSS.</h1>
5. <p>This paragraph is not affected.</p>
6. </body>
7. </html>
```

Output:

Inline CSS



Disadvantages of Inline CSS

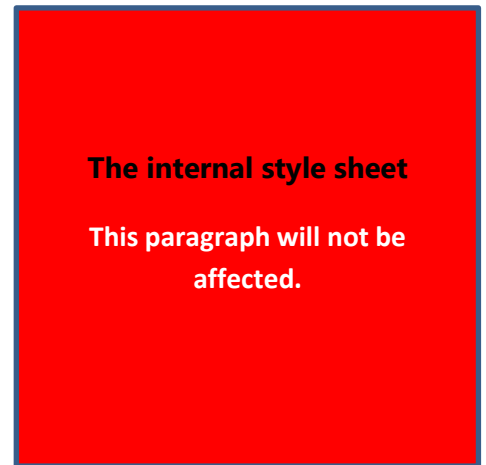
- You cannot use quotations within inline CSS. If you use quotations the browser will interpret this as an end of your style value.
- These styles cannot be reused anywhere else.
- These styles are tough to be edited because they are not stored at a single place.
- It is not possible to style pseudo-codes and pseudo-classes with inline CSS.
- Inline CSS does not provide browser cache advantages.

Internal CSS:

- The internal style sheet is used to add a unique style for a single document.
- The internal style is defined inside the <style> element, inside the head section.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5.   body {
6.     background-color: red;
7.   }
8.   h1 {
9.     color: black;
10.    margin-left: 80px;
11.  }
12. </style>
13. </head>
14. <body>
15. <h1>The internal style sheet </h1>
16. <p>This paragraph will not be affected.</p>
17. </body>
18. </html>
```

Output:



External CSS:

- The external style sheet is generally used when you want to make changes on multiple pages.
- It is ideal for this condition because it facilitates you to change the look of the entire web site by changing just one file.
- It uses the <link> tag on every pages and the <link> tag should be put inside the head section.

1. `<head>`
2. `<link rel="stylesheet" type="text/css" href="mystyle.css">`
3. `</head>`

The external style sheet may be written in any text editor but must be saved with a .css extension. This file should not contain HTML elements.

Let's take an example of a style sheet file named "mystyle.css".

File: mystyle.css

```
1. body {  
2.     background-color: lightblue;  
3. }  
4. h1 {  
5.     color: navy;  
6.     margin-left: 20px;  
7. }
```

Inheritance:

- The inherit keyword specifies that a property should inherit its value from its parent element.
 - The inherit keyword can be used for any CSS property, and on any HTML element.
 - The value of inherit keyword is used to inherit a property to an element from parent element property value.
 - Inheritance is always from the parent element in the document tree.
1. `<!DOCTYPE html>`
 2. `<html>`
 3. `<head>`
 4. `<style>`
 5. `span {`
 6. `color: blue;`
 7. `border: 1px solid black;`
 8. `}`
 9. `.extra span {`
 10. `color: inherit;`
 11. `}`

```
12. </style>
13. </head>
14. <body>
15. <div>
16. Here is <span>a span element</span> which is blue, as span elements are set to be.
17. </div>
18. <div class="extra" style="color:green">
19. Here is <span>a span element</span> which is green, because it inherits from its
    parent.
20. </div>
21. <div style="color:red">
22. Here is <span>a span element</span> which is blue, as span elements are set to be.
23. </div>
24. </body>
25. </html>
```

Output:

Here is a span element which is blue, as span elements are set to be.
Here is a span element which is green, because it inherits from its parent.
Here is a span element which is blue, as span elements are set to be.

CSS Background

CSS background property is used to define the background effects on element. There are 5 CSS background properties that affects the HTML elements:

1. background-color
2. background-image
3. background-repeat
4. background-attachment
5. background-position

CSS background-color:

The background-color property is used to specify the background color of the element.

background-color: #b0d4de;

CSS background-image

The background-image property is used to set an image as a background of an element. By default the image covers the entire element. You can set the background image for a page like this.

```
background-image: url("paper.gif");
```

CSS background-repeat

The background-image property repeats the background image horizontally and vertically. Some images are repeated only horizontally or vertically.

The background looks better if the image repeated horizontally only.

```
1. body {  
2.   background-image: url("gradient-bg.png");  
3.   background-repeat: repeat-x;  
4. }
```

CSS background-attachment

The background-attachment property is used to specify if the background image is fixed or scroll with the rest of the page in browser window.

If you set fixed the background image then the image will not move during scrolling in the browser.

```
1. <style>  
2. body {  
3.   background: white url('bbb.gif');  
4.   background-repeat: no-repeat;  
5.   background-attachment: fixed;  
6.   margin-left:200px;  
7. }  
8. </style>
```


CSS background-position

The background-position property is used to define the initial position of the background image. By default, the background image is placed on the top-left of the webpage.

You can set the following positions:

- center
- top
- bottom
- left
- right

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. body {
6. background: white url('good-morning.jpg');
7. background-repeat: no-repeat;
8. background-attachment: fixed;
9. background-position: center;
10. }
11. </style>
12. </head>
13. <body>
14. <p> background.</p>
15. <p> background.</p>
16. <p>This is a fixed background-image. Scroll down the page.</p>
17. <p>This is a fixed background-image. Scroll down the page.</p>
18. </body>
19. </html>
```

CSS Border

The CSS border is a shorthand property used to set the border on an element.

The CSS border properties are used to specify the style, color and size of the border of an element. The CSS border properties are given below

- border-style- (dotted ,dash)
- border-color (color red:))
- border-width
- border-radius (curve)

Shadow CSS

Shadow CSS is a great method to make our website more appealing. CSS Shadows are a visual effect that consists of a drawing that resembles an object's shadow; these shadows provide a 3-dimensional appearance to the object.

In CSS, mainly the **text-shadow** and the **box-shadow** property is used to add shadows to texts and elements.

CSS text-shadow property:

Syntax:

```
text-shadow: x-label, y-label, blur radius, color/none/initial;
```

Values:

- **X- label:-** It is a compulsory field that specifies the horizontal shadow position. Negative x-label values are also allowed. To shift the shadow in right side put the positive x-value e.g: 3px, 3rem, 3em. Similarly to shift the shadows in left side enter the negative x-values e.g: -3px, -3rem, -3em etc.
- **Y-label:-** It is also a compulsory field which specifies the vertical shadow position. Negative y-label values are allowed. To shift the shadow on the lower side, put the positive y-value, e.g.: 3px, 3rem, 3em. Similarly, to shift the shadows on the upper side, enter the negative y-values, e.g.: -3px, -3rem, -3em etc.
- **Blur Radius:-** It is an optional field that blurs the shadow. Its default value is 0.
- **Color:-** It is also an optional field that specifies the color of the shadow. Any of the CSS colors can be used to change the color of the shadow.
- **None:-** It ensures that there is no shadow.
- **Initial:-** It sets the properties to the default values

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):

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. h1 {
6. text-shadow: 2px 2px red;
7. }
8. </style>
9. </head>
10. <body>
11. <h1>Text-shadow effect!</h1>
12. </body>
13. </html>
```

Text-shadow effect!

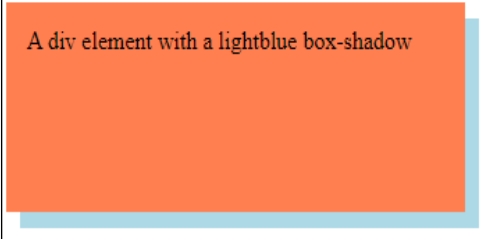
CSS box-shadow Property:

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

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. div {
6. width: 300px;
7. height: 100px;
8. padding: 15px;
9. background-color: coral;
10. box-shadow: 10px 10px lightblue;
11. }
12. </style>
13. </head>
14. <body>
15. <h1>The box-shadow Property</h1>
16. <div>A div element with a lightblue box-shadow</div>
17. </body>
18. </html>
```

The box-shadow Property

A div element with a lightblue box-shadow



CSS Text Effects

We can apply different effects on the text used within an HTML document. Some properties can be used for adding the effects on text.

Using CSS, we can style the web documents and affects the text. The properties of the text effect help us to make the text attractive and clear. There are some text effect properties in CSS that are listed below:

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

word-break

It specifies how words should break at the end of the line. It defines the line break rules.

CSS Text Overflow

It specifies the representation of overflowed text, which is not visible to the user. It signals the user about the content that is not visible. This property helps us to decide whether the text should be clipped or show some dots (ellipsis).

word-wrap

CSS word-wrap property is used to break the long words and wrap onto the next line. This property is used to prevent overflow when an unbreakable string is too long to fit in the containing box.

writing-mode

It specifies whether the text will be written in the horizontal or vertical direction. If the writing direction is vertical, then it can be from left to right (vertical-lr) or from right to left (vertical-rl).

CSS Text Overflow code

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. p.test1 {
6. white-space: nowrap;
7. width: 200px;
8. border: 1px solid #000000;

```
9. overflow: hidden;
10. text-overflow: clip;
11. }
12. p.test2 {
13. white-space: nowrap;
14. width: 200px;
15. border: 1px solid #000000;
16. overflow: hidden;
17. text-overflow: ellipsis;
18. }
19. </style>
20. </head>
21. <body>
22. <h1>The text-overflow Property</h1>
23. <p>The following two paragraphs contains a long text that will not fit in the box.</p>
24. <h2>text-overflow: clip:</h2>
25. <p class="test1">This is some long text that will not fit in the box</p>
26. <h2>text-overflow: ellipsis:</h2>
27. <p class="test2">This is some long text that will not fit in the box</p>
28. </body>
29. </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 ...

CSS TEXT

✓ Text Color

✓ Text Alignment and Text Direction

1. text-align
2. text-align-last
3. direction
4. Unicode-bidi
5. vertical-align

✓ Text Decoration

1. text-decoration-line
2. text-decoration-color
3. text-decoration-style
4. text-decoration-thickness
5. text-decoration

✓ text-transform property

- 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:

✓ Text Spacing

- text-indent
- letter-spacing
- line-height
- word-spacing
- white-space

CSS Transforms:

- CSS3 supports transform property. This transform property facilitates you to translate, rotate, scale, and skews elements.
- Transformation is an effect that is used to change shape, size and position.
- There are two type of transformation i.e. 2D and 3D transformation supported in CSS3.

CSS 2D Transforms

The CSS 2D transforms are used to re-change the structure of the element as translate, rotate, scale and skew etc.

Following is a list of 2D transforms methods:

- **translate(x,y):** It is used to transform the element along X-axis and Y-axis.
- **translateX(n):** It is used to transform the element along X-axis.

- **translateY(n):** It is used to transform the element along Y-axis.
- **rotate():** It is used to rotate the element on the basis of an angle.
- **scale(x,y):** It is used to change the width and height of an element.
- **scaleX(n):** It is used to change the width of an element.
- **scaleY(n):** It is used to change the height of an element.
- **skewX():** It specifies the skew transforms along with X-axis.
- **skewY():** It specifies the skew transforms along with Y-axis.
- **matrix():** It specifies matrix transforms.

The translate() method:

- The CSS translate() method is used to move an element from its current position according to the given parameters i.e. X-axis and Y-axis.

The rotate() method:

- The CSS rotate() method is used to rotate an element clockwise or anti-clockwise according to the given degree.

The scale() method:

- The CSS scale() method is used to increase or decrease the size of an element according to the given width and height.

The skewX() method:

- The CSS skewX() method is used to skew an element along the X axis according to the given angle.

The skewY() method:

- The CSS skewY() method is used to skew an element along the Y axis according to the given angle.

The skew() method:

- The CSS skew() method is used to skew an element along with X-axis and Y-axis according to the given angle.

The matrix() method:

- The CSS matrix() method combines all the six 2D transform methods altogether. It allows you to rotate, scale, translate, and skew elements.

Ex.

📌 CSS Border:

The CSS border is a shorthand property used to set the border on an element.

The CSS border properties are used to specify the style, color and size of the border of an element. The CSS border properties are given below

- border-style
- border-color
- border-width
- border-radius

📌 CSS Border Images:

The CSS border-image property allows you to specify an image to be used instead of the normal border around an element.

The property has three parts:

1. The image to use as the border
2. Where to slice the image
3. Define whether the middle sections should be repeated or stretched

The border-image property takes the image and slices it into nine sections, like a tic-tac-toe board. It then places the corners at the corners, and the middle sections are repeated or stretched as you specify.

Note: For border-image to work, the element also needs the border property set!

Syntax:

`border-image: source slice width outset repeat | initial |`

The border-image property can be applied to all elements except the elements of the internal table (such as tr, th, td) when border-collapse is set to collapse.

It is the shorthand property for border-image-source, border-image-slice, border-image-width, border-image-outset, and border-image-repeat. We can set all these properties at once using the border-image property

Ex

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <title>
5. CSS border-image Property
6. </title>
7. <style>
8. p{
9.   border: 10px solid transparent;
10. padding: 15px;
11. text-align:center;
12. font-size: 25px;
13. color: darkviolet;
14. }
15. #border1 {
16. border-image: url('diamond.png') 43 / 10px 15px round stretch;
17. }
18. </style>
19. </head>
20. <body>
21. <h2>border-image property</h2>
22. <p id = "border1">
23. Welcome to the javaTpoint.com
24. </p>
25. </body>
26. </html>
```

OUTPUT:

border-image property



CSS Colors

The color property in CSS is used to set the color of HTML elements. Typically, this property is used to set the background color or the font color of an element.

In CSS, we use color values for specifying the color. We can also use this property for the border-color and other decorative effects.

We can define the color of an element by using the following ways:

- RGB format.
- RGBA format.
- Hexadecimal notation.
- HSL.
- HSLA.
- Built-in color.

RGB Format:

RGB format is the short form of 'RED GREEN and BLUE' that is used for defining the color of an HTML element simply by specifying the values of R, G, B that are in the range of 0 to 255.

The color values in this format are specified by using the `rgb()` property. This property allows three values that can either be in percentage or integer (range from 0 to 255).

Syntax: `color: rgb(R, G, B);`

RGBA Format:

It is almost similar to RGB format except that RGBA contains A (Alpha) that specifies the element's transparency. The value of alpha is in the range 0.0 to 1.0, in which 0.0 is for fully transparent, and 1.0 is for not transparent.

Syntax: `color: rgba(R, G, B);`

Hexadecimal notation:

Hexadecimal can be defined as a six-digit color representation. This notation starts with the # symbol followed by six characters ranges from 0 to F. In hexadecimal notation, the first two digits represent the red (RR) color value, the next two digits represent the green (GG) color value, and the last two digits represent the blue (BB) color value.

Syntax: `color: # (0-F)(0-F)(0-F)(0-F)(0-F)(0-F);`

Short Hex codes:

It is a short form of hexadecimal notation in which every digit is recreated to arrive at an equivalent hexadecimal value.

For example, #7B6 becomes #77BB66 in hexadecimal.

HSL:

It is a short form of **Hue, Saturation, and Lightness**. Let's understand them individually.

Hue: It can be defined as the degree on the color wheel from 0 to 360.

Saturation: It takes value in percentage in which 100% represents fully saturated, i.e., no shades of gray, 50% represent 50% gray, but the color is still visible, and 0% represents fully unsaturated, i.e., completely gray, and the color is invisible.

Lightness: The lightness of the color can be defined as the light that we want to provide the color in which 0% represents black (there is no light), 50% represents neither dark nor light, and 100% represents white (full lightness).

Syntax: color:hsl(H, S, L);

HSLA:

It is entirely similar to HSL property, except that it contains A (alpha) that specifies the element's transparency. The value of alpha is in the range 0.0 to 1.0, in which 0.0 indicates fully transparent, and 1.0 indicates not transparent.

Syntax: color: hsla(H, S, L, A);

Built-in Color:

As its name implies, built-in color means the collection of previously defined colors that are used by using a name such as red, blue, green, etc.

Syntax: color: color-name;

EX:

```
<!DOCTYPE html>

<html>

<body>

<h1 style="background-color:DodgerBlue;">Hello World</h1>

<p style="background-color:Tomato;">

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exercitation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

</p>

</body>

</html>
```

OUTPUT:

Hello World

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exercitation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

CSS Transition

The CSS transitions are effects that are added to change the element gradually from one style to another, without using flash or JavaScript.

You should specify two things to create CSS transition.

- The CSS property on which you want to add an effect.
- The time duration of the effect.

Properties:

- transition
- transition-delay
- transition-duration
- transition-property
- transition-timing-function

EX.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. div {
6. width: 100px;
7. height: 100px;
8. background: red;
9. transition-property: width;
10. transition-duration: 2s;
11. transition-timing-function: linear;
12. transition-delay: 1s;
13. }
14. div:hover {
15. width: 300px;
16. }
17. </style>
18. </head>
19. <body>
20. <h1>The transition Properties Specified One by One</h1>
21. <p>Hover over the div element below, to see the transition effect:</p>
22. <div></div>
23. </body>
```

CSS Animation:

CSS Animation property is used to create animation on the webpage. It can be used as a replacement of animation created by Flash and JavaScript.

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.

Properties:

- @keyframes
- animation-name
- animation-duration
- animation-delay
- animation-iteration-count
- animation-direction
- animation-timing-function
- animation-fill-mode
- animation

CSS3 @keyframes Rule

The animation is created in the @keyframe rule. It is used to control the intermediate steps in a CSS animation sequence.

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.

The **animation-delay property** specifies a delay for the start of an animation.

The **animation-iteration-count** property specifies the number of times an animation should run.

The animation-direction property specifies whether an animation should be played forwards, backwards or in alternate cycles.

The animation-direction property can have the following values:

- **normal** - The animation is played as normal (forwards). This is default
- **reverse** - The animation is played in reverse direction (backwards)
- **alternate** - The animation is played forwards first, then backwards
- **alternate-reverse** - The animation is played backwards first, then forwards.

The animation-timing-function property specifies the speed curve of the animation.

The animation-timing-function property can have the following values:

- **ease** - Specifies an animation with a slow start, then fast, then end slowly (this is default)
- **linear** - Specifies an animation with the same speed from start to end
- **ease-in** - Specifies an animation with a slow start
- **ease-out** - Specifies an animation with a slow end
- **ease-in-out** - Specifies an animation with a slow start and end
- **cubic-bezier(n,n,n,n)** - Lets you define your own values in a cubic-bezier function.

The animation-fill-mode property specifies a style for the target element when the animation is not playing (before it starts, after it ends, or both).

The animation-fill-mode property can have the following values:

- **none** - Default value. Animation will not apply any styles to the element before or after it is executing
- **forwards** - The element will retain the style values that is set by the last keyframe (depends on animation-direction and animation-iteration-count)
- **backwards** - The element will get the style values that is set by the first keyframe (depends on animation-direction), and retain this during the animation-delay period
- **both** - The animation will follow the rules for both forwards and backwards, extending the animation properties in both directions.

EX.

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5.   div {
6.     width: 100px;
7.     height: 100px;
8.     background-color: red;
9.     position: relative;
10.    animation-name: example;
11.    animation-duration: 5s;
12.    animation-timing-function: linear;
13.    animation-delay: 2s;
14.    animation-iteration-count: infinite;
15.    animation-direction: alternate;
16.  }
17. @keyframes example {
18.  0% {background-color:red; left:0px; top:0px;}
19.  25% {background-color:yellow; left:200px; top:0px;}
20.  50% {background-color:blue; left:200px; top:200px;}
21.  75% {background-color:green; left:0px; top:200px;}
22.  100% {background-color:red; left:0px; top:0px;}
23. }
24. </style>
25. </head>
26. <body>
27. <h1>CSS Animation</h1>
28. <p>This example uses six of the animation properties:</p>
29. <div>
30. </div>
31. </body>
32. </html>
```