CSS Properties

Background

CSS **background** property is used to define the background effects on Html element. There are 5 CSS background properties that affect the Html elements.

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

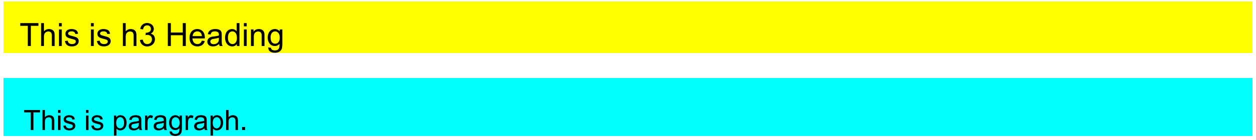
background-color

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

Example

```
<html>
<head>
<style>
h3{     background:yellow;      }
p{      background:cyan;        }
</style>
</head>
<body>
<h3>This is h3 Heading</h3>
<p>This is paragraph</p>
</body>
</html>
```

Output:



This is h3 Heading

This is paragraph.

In above example we set the background color cyan for <p> and yellow for <h3>.

background-image

The background-image property is used to specify the image in background of the Html element.

Example

```
<html>
<head>
<style>
p{background-image:url(car2.png);}
</style>
</head>
<body>
<p>This is paragraph</p>
</body>
</html>
```

background-repeat

By default, the background-image property repeats the background image horizontally and vertically. Some images are repeated only horizontally or vertically. If you do not want to repeat background image then set **no-repeat**.

Example: repeat in x-axis

```
<html>
<head>
```

```
<style>
p{
background-image:url(car2.png);
background-repeat: repeat-x;
}
</style>
</head>
<body>
<p>This is paragraph</p>
</body>
</html>
```

Example: repeat in y-axis

```
<html>
<head>
<style>
p{
background-image:url(car2.png);
background-repeat: repeat-y;
}
</style>
</head>
<body>
<p>This is paragraph</p>
</body>
</html>
```

background-position

background-position is use only in case of when you use image in background of any Html elements. It 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 position of background, all background image position are given below;

- top
- bottom
- center
- left
- right

example:

```
p{
background-image:url(car3.png);
background-repeat: no-repeat;
background-position: bottom;
}
```

CSS Border

The CSS border properties allow you to specify the style, size, and color of an Html element border. Following CSS border properties are used for Html elements,

- border-style
- border-width
- border-color
- border-radius
- border-shadow : HposVerposblurradius spread shadowcolor

Border Style

The border-style property specifies what kind of border to display.

Value	Description
none	It is used for does not define any border.
dotted	It is used to define a dotted border.
dashed	It is used to define a dashed border.
solid	It is used to define a solid border.
double	It defines two borders with the same border-width value.
groove	It defines a 3D grooved border. effect is generated according to border-color value.
ridge	It defines a 3D ridged border. effect is generated according to border-color value.
inset	It defines a 3D inset border. effect is generated according to border-color value.
outset	It defines a 3D outset border. effect is generated according to border-color value.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p.none {border-style: none;}
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
```

```
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.hidden {border-style: hidden;}
</style>
</head>
<body>
<p class="none">No border.</p>
<p class="dotted">A dotted border.</p>
<p class="dashed">A dashed border.</p>
<p class="solid">A solid border.</p>
<p class="double">A double border.</p>
<p class="groove">A groove border.</p>
<p class="ridge">A ridge border.</p>
<p class="inset">An inset border.</p>
<p class="outset">An outset border.</p>
<p class="hidden">A hidden border.</p>
</body>
</html>
```

Output

No border.

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border.

A ridge border.

An inset border.

An outset border.

Border Width

The **border-width** property is used to set the border width. It is set in pixels. You can also set the width of the border by using pre-defined values, thin, medium or thick.

Example: css border-color

```
<!DOCTYPE html>
<html>
<head>
<style>
p.para1 {
border-style: solid;
width:200px;
border-color: green;
}
p.para2{
border-style: solid;
width:100px;
border-color: red;
}
</style>
```

```
</head>
<body>
<p class="para1">This is a solid red border</p>
<p class="para2">This is a solid green border</p>
</body>
</html>
```

Output

border-radius 50px

border-radius 10px

Border Color

This properties are used for set the color of Html elements border. There are three method to set the color of border.

- **Name:** It specifies the color name. For example: "red".
- **RGB:** It specifies the RGB value of the color. For example: "rgb(255,0,0)".
- **Hex:** It specifies the hex value of the color. For example: "#ff0000".

Example: css border-color

```
<!DOCTYPE html>
<html>
<head>
<style>
p.para1 {
border-style: solid;
border-color: green;
}
p.para2 {
```

```
border-style: solid;
border-color: red;
}
</style>
</head>
<body>
<p class="para1">This is a solid red border</p>
<p class="para2">This is a solid green border</p>
</body>
</html>
```

Output

This is a solid red border

This is a solid green border

Border Radius

border-radius are used for give the radius for border, using this you can make a circle.

Example: css border-color

```
<!DOCTYPE html>
<html>
<head>
<style>
p.parar1 {
border-style: solid;
border-radius: 100%;
border-color: green;
}
p.parar2 {
```

```
border-style: solid;
border-radius: 50%;
        height:100px;
        width:100px;
border-color: red;
}
</style>
</head>
<body>
<p class="parar1">This is a solid red border</p>
<p class="parar2">This is a solid green border</p>
</body>
</html>
```

CSS Display

CSS display is the most important property of CSS which is used to display Html elements on web page. Every Html element on the webpage is a rectangular box and the CSS display property specifies the type of box used for an Html element.

CSS display values

Values	Description
inline	It is used to displays an Html element in same line (like <span>).
none	It is used to hide the Html element.
block	It is used to displays an element as a block element (like <p>).

list-item	Let the element behave like a <li> element.

display:inline

It is used to display an Html elements in same line without any line break. In below example three paragraph display in same line.

```
<!DOCTYPE html>
<html>
<head>
<style>
p{
display:inline;
}
</style>
</head>
<body>
<p>Inline display</p>
<p>Inline display</p>
<p>Inline display</p>
</body>
</html>
```

Output:

```
Inline display Inline display Inline display
```

display:none

It is used for hide text on browser but it does not take any space. In below example we hide three paragraph texts.

```
<!DOCTYPE html>
<html>
<head>
<style>
p{
display:none;
}
</style>
</head>
<body>
<p>Text not display</p>
<p>Text not display</p>
<p>Text not display</p>
</body>
</html>
```

display:block

It is used to displays an element as a block element. It displayselements same like <p> tag. In below example we display text by using <span> tag. It take some space and also line break same like paragraph.

```
<!DOCTYPE html>
<html>
<head>
<style>
span
{
display:block;
}
</style>
```

```
</head>
<body>
<span>Block display elements</span>
<span>Block display elements</span>
<span>Block display elements</span>
</body>
</html>
```

Output

```
Inline-Block elements
inline-Block elements
```

display:inline-block

It is used to displays an element as a block element. It display an elements same like <p> tag. In below example we display text by using <span> tag. It take some space but it display all element in same line.

```
<!DOCTYPE html>
<html>
<head>
<style>
p{
display:inline-block;
}
</style>
</head>
<body>
<span>Inline-block elements</span>
<span>Inline-block elements</span>
<span>Inline-block elements</span>
```

```
</body>

</html>
```

Output

```
Inline-Block elements Inline-Block elements
```

CSS Float

The CSS **float** property is a positioning property. It is used to push an element to the left or right, allowing other element to wrap around it. It is generally used with images and layouts.

How it works

Elements are floated only horizontally. So it is possible only to float elements left or right, not up or down.

- A floated element may be moved as far to the left or the right as possible. Simply, it means that a floated element can display at extreme left or extreme right.
- The elements after the floating element will flow around it.
- The elements before the floating element will not be affected.
- If the image floated to the right, the texts flow around it, to the left and if the image floated to the left, the text flows around it, to the right.

CSS Float Properties

Property	Description	Values
clear	The clear property is used to avoid elements after the floating elements which flow around it.	left, right, both, none, inherit

float	It specifies whether the box should float or not.	left, right, none, inherit

```
<!DOCTYPE html>
<html>
<head>
<style>
input{
float:right;
}
</style>
</head>
<body>
<p>Float textbox's from right</p>
<input><input><input><input>
</body>
</html>
```

CSS Font

CSS font is used for design text or font for display on web page. CSS font properties define the font family, boldness, size, and the style of a text.

Font-style property:

Font styles are used for set font style. Font style property has three values they are;

- **normal:** The text is shown normally
- **italic:** The text is shown in italics
- **oblique:** The text is "leaning" (oblique is very similar to italic, but less supported)

Example: font style

```
<html>
<head>
<style>
h1 {
font-style: normal;
}
h2 {
font-style: italic;
}
p {
font-style: oblique;
}
</style>
</head>
<h1>This is h1 Heading</h1>
<h2>This is h2 Heading</h2>
<p>This is Paragraph</p>
</body>
</html>
```

Output

This is h1 Heading

This is h2 Heading

This is Paragraph

Font-Size property:

We can define size of font in following way

With Em

With Percent

With Pixels

If you set the text size with pixels then you can gives full control over the text size.

Example:

```
<html>
<head>
<style>
h1 {
font-size: 40px;
}
h2 {
font-size: 30px;
}
p {
font-size: 14px;
}
</style>
</head>
<h1>This is h1 Heading</h1>
<h2>This is h2 Heading</h2>
<p>This is Paragraph</p>
</body>
</html>
```

Font size in Em

The em size unit is recommended by the W3C. 1em is equal to default text size in browsers (16px).

The size convert pixels to em using this formula: pixels/16=em

Example:

```
<html>
<head>
<style>
h1 {
font-size: 2.5em; * 40px/16=2.5em */
}
h2 {
font-size: 1.875em; /* 30px/16=1.875em */
}
p {
font-size: 0.875em; /* 14px/16=0.875em */
}
</style>
</head>
<h1>This is h1 Heading</h1>
<h2>This is h2 Heading</h2>
<p>This is Paragraph</p>
</body>
</html>
```

Font size in percent

Text size work on all browsers, you can use default font-size in percent for the <body> element.

Example:

```
<html>
<head>
<style>
body {
font-size: 100%;
}
h1 {
```

```
font-size: 2.5em;
}
h2 {
font-size: 1.875em;
}
p {
font-size: 0.875em;
}
</style>
</head>
<h1>This is h1 Heading</h1>
<h2>This is h2 Heading</h2>
<p>This is Paragraph</p>
</body>
</html>
```

CSS Margin

margins are used for give sufficient space around an element (outside the border). The margin does not have a background color, and it is completely transparent. Margin have four properties which is given below;

- top
- bottom
- left
- right

Example

```
<html>
<body>
<p style="margin-top:150px;"> Margin from top</p>
```

```
<p style="margin-bottom:100px;">Margin from bottom</p>
<p style="margin-right:50px;">Margin from right</p>
<p style="margin-left:50px;">Margin from left</p>
</body>
</html>
```

CSS Padding

The padding clears an area around the content (inside the border) of an element. The padding is affected by the background color of the element. Padding have four properties which is given below;

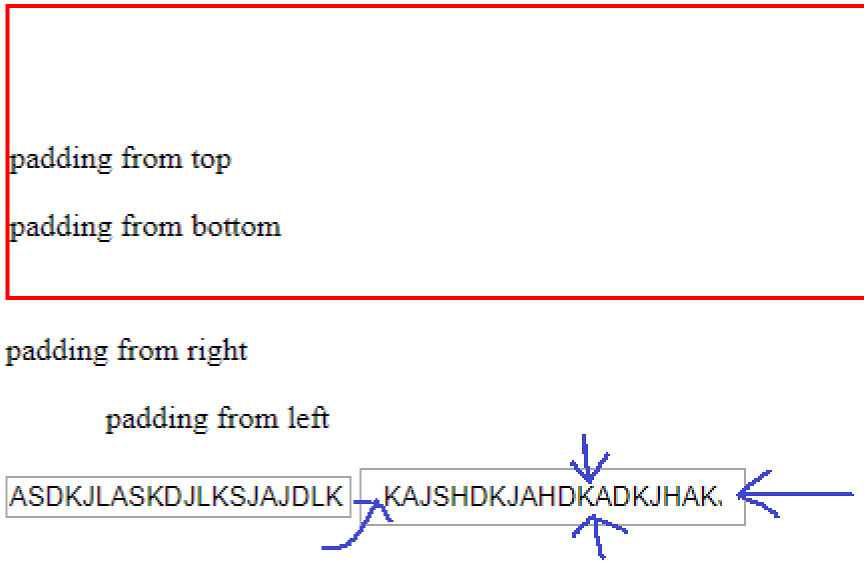
- top
- bottom
- left
- right

Example

```
<html>
<body>
<div style="border:2px solid red;">
        <p style="padding-top:50px;">padding from top</p>
        <p style="padding-bottom:10px;">padding from bottom</p>
</div>
<p style="padding-right:50px;">padding from right</p>
<p style="padding-left:50px;">padding from left</p>
<!—NORMAL TEXTFIELD →
<input type="text" name="t1"/>
<!—TEXTFIELD WITH PADDING SETTINGS →
<input type="text" name="t1" style="padding-right:10px;padding-left:10px;padding-top:5px;padding-bottom:5px;"/>
```

```
</body>
</html>
```

Output



padding from top

padding from bottom

padding from right

padding from left

ASDKJLASKDJLKSJAJDLK KAJSHDKJAHDKADKJHAK

CSS Overflow

CSS overflow property used to handle the content when it overflows its block level container.

Why need of Overflow?

Suppose if you do not set the height and width of any box then it goes larger as content, but if you do not set height or width of box but content cannot fit inside box then it goes overflow. The CSS overflow property is used to overcome this problem.

Overflow property values

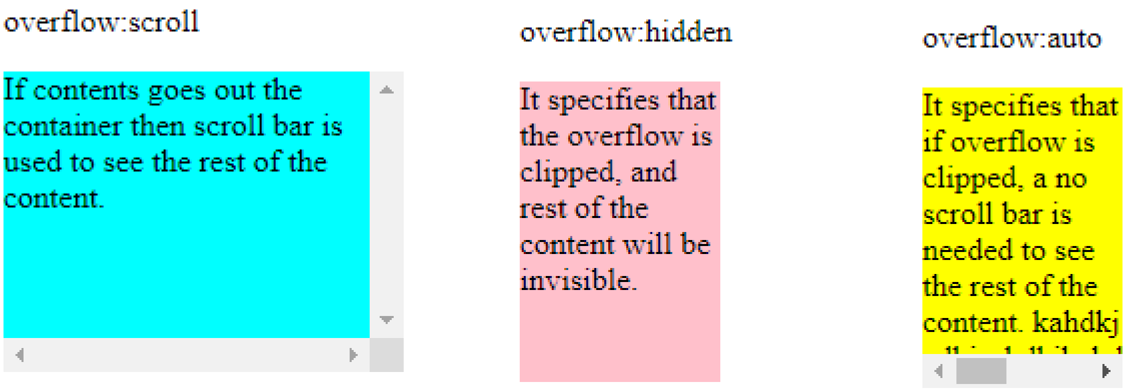
Values	Description
visible	It specifies that overflow is not clipped. It renders outside the element's box. This is a default value.

hidden	It specifies that the overflow is clipped, and rest of the content will be invisible.
scroll	It specifies that the overflow is clipped, and a scroll bar is used to see the rest of the content.
auto	It specifies that if overflow is clipped, a scroll bar is needed to see the rest of the content.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
div.scroll {
background-color: cyan;
width: 200px;
height: 150px;
overflow: scroll;
}
div.hidden {
background-color: pink;
width: 100px;
height: 150px;
overflow: hidden;
}
div.auto {
background-color: yellow;
width: 100px;
height: 150px;
overflow: auto;
}
</style>
</head>
<body>
<p>overflow:scroll</p>
<div class="scroll">
If contents goes out the container then scroll bar is used to see the rest of the content.
</div>
<p>overflow:hidden</p>
<div class="hidden">
It specifies that the overflow is clipped, and rest of the content will be invisible.
</div>
<p>overflow:auto</p>
<div class="auto">
It specifies that if overflow is clipped, a scroll bar is needed to see the rest of the content.
</div>
</body>
</html>
```

Output



overflow:scroll

If contents goes out the container then scroll bar is used to see the rest of the content.

overflow:hidden

It specifies that the overflow is clipped, and rest of the content will be invisible.

overflow:auto

It specifies that if overflow is clipped, a no scroll bar is needed to see the rest of the content. kahdkj

CSS Position

position properties are used for set the position of Html elements. css position properties are used for set the position of text, image for display on web page.

CSS have following position properties;

- **Relative**
- **Absolute**

Fixed

These properties are used for fixed position of elements, when you scroll down any web page that time these elements display on same position.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p.pos_fixed {
position: fixed;
top: 40px;
right: 10px;
color: green;
}
</style>
</head>
<body>
<p class="pos_fixed">This is my first html with csscode.</p>
<p>This is my first html with csscode.</p>
</body>
</html>
```

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p.pos_relative {
position: relative;
top: 20px;
right: 10px;
color: green;
}
</style>
</head>
<body>
<p class="pos_relative">This is my first html with csscode.</p>
<p>This is my first html with csscode.</p>
</body>
</html>
```

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p.pos_absolute {
position: absolute;
top: 140px;
right: 50px;
color: green;
}
</style>
</head>
<body>
<p class="pos_absolute">This is my first html with csscode.</p>
<p>This is my first html with csscode.</p>
</body>
</html>
```

CSS Opacity

In CSS we can easily create transparent images very easily by using **Opacity**.The **CSS Opacity** property is used to specify the transparency of any Html element. In simple word, you can say that it specifies the clarity of the image.

Opacity value should be range 0 to 1 only.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
img.trans {
opacity: 0.4;
filter: alpha(opacity=40); /* this one for For IE8 and earlier */
}
</style>
</head>
<body>
<imgsrc="car3.png" alt="logo">
```

```
<img class="trans" src="car3.png" alt="logo">
</body>
</html>

<!DOCTYPE html>
<html>
<head>
<style>
        #i{
        background-image:url("w.jpg");
        }
   #div1{
     width:500px;
     height:300px;
     background-color:#3399ff;
     opacity:0.6;
    }
</style>
</head>
<body id="i">
<div id="div1">
    hello world
</div>
</body>
</html>
```

CSS TextStyles

CSS text styles are used specify space between letters, words, lines etc…

Value	Description
Letter-spacing	This property specifies gap between letters.
Word-spacing	This property specifies gap between words.
Line-height	This property specifies height of line of text.
Text-decoration	This property specifies underline/overline/strikeout/overline/line-through/none. Default value is none.
Text-transform	This property specifies case of text. Those are uppercase / lowercase / capitalize / none. Default value is none.
Text-align	Used to specify alignment of text. Alignments are left / right / center / justify. Default is left.
Text-shadow	Used to apply shadow foe the text.

Example

```
<!-- text spaces -->

<!DOCTYPE html>

<html>

<head>

<style>

        p{ font-size:20px; }

        .p1{ letter-spacing: 5px; }

        .p2{ letter-spacing: -1px; }

        .p3{ word-spacing: 10px; }

        .p4{ line-height: 40px; }

        .p5{ text-decoration: line-through; }

        .p6{text-transform:uppercase; }

        .p7{ text-align: center; }

</style>

</head>

<body>

        <p class="p1">This line Demonstrate letter spacing </p>

        <p class="p2">This line Demonstrate letter spacing -ve value</p>

        <p class="p3">This line Demonstrate word spacing </p>

        <p class="p4">This line Demonstrate line height </p>

        <p class="p5">This line Demonstrate text-decoration </p>

        <p class="p6">This line Demonstrate text-transform </p>

        <p class="p7">This line Demonstrate text-align </p>

</body>

</html>
```

Output

This line Demonstrate letter spacing

This line Demonstrate letter spacing -ve value

This line Demonstrate word spacing

This line Demonstrate line height

~~This line Demonstrate text-decoration~~

THIS LINE DEMONSTRATE TEXT-TRANSFORM

This line Demonstrate text-align

CSS Word Wrap

CSS **Word Wrap properties** are used for breaks the long words and wrap onto the next line. The main use of these properties is to prevent overflow when a long string unable to fit into containing box.

word wrap values

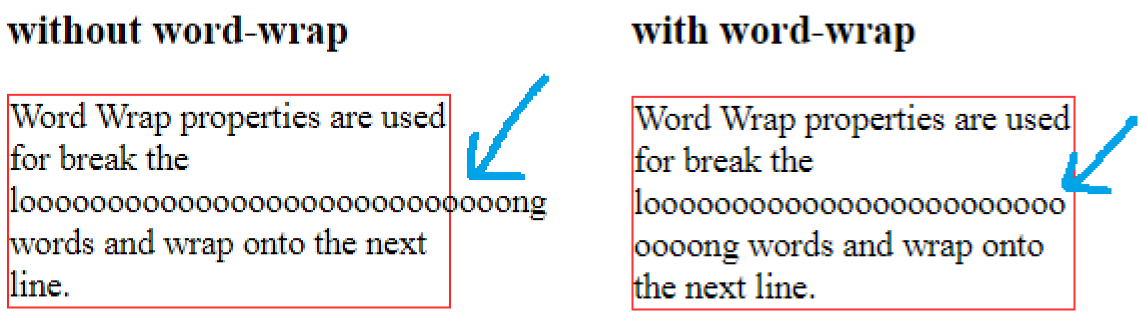
Value	Description
normal	This property is used to break words only at allowed break points.
break-word	It is used to break unbreakable words.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
         .p1 {
        width: 200px;
        border: 1px solid #ff3333;
         }
        .p2{
        width: 200px;
        border: 1px solid #ff3333;
        word-wrap: break-word;
         }
</style>
</head>
<body>
<h3>without word-wrap</h3>
<p class="p1">Word Wrap properties are used for break the looooooooooooooooooooooooooooong
words and wrap onto the next line. </p>

<h3>with word-wrap</h3>
<p class="p2">Word Wrap properties are used for break the looooooooooooooooooooooooooooong
words and wrap onto the next line. </p>
</body>
</html>
```

Output



without word-wrap

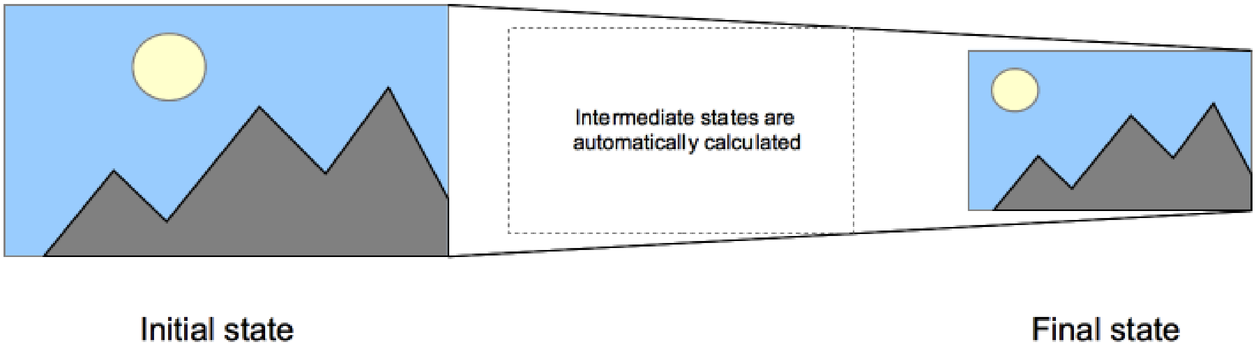
Word Wrap properties are used
for break the
looooooooooooooooooooooooooooong
words and wrap onto the next
line.

with word-wrap

Word Wrap properties are used
for break the
looooooooooooooooooooooooo
oooong words and wrap onto
the next line.

CSS Transitions

Normally when the value of a CSS property changes, the rendered result is instantly updated, with the affected elements immediately changing from the old property value to the new property value. This section describes a way to specify transitions using new CSS properties. These properties are used to animate smoothly from the old state to the new state over time.



Note: Transition will we started when you move cursor on it. You must need to specify time duration otherwise Transition have no effect. The default time value is zero.

You need to specify two things for create CSS transition.

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

Transitions support only pixels based and colour based properties only.

Ex: width, height, opacity, font-size, border-width, background-colour, colour, border-colour etc.

```
selector{
        property : old-value;
        property : old-value;
          …
        transition: property  time (sec/ms), property  time (sec/ms), …
}
Selector:hover{
        property :new-value;
        property : new-value;
          ....
}
```

Property	Description
transition	A shorthand property for setting the four transition properties into a single property
transition-delay	Specifies when the transition effect will start.
transition-duration	Specifies how many seconds or milliseconds a transition effect takes to complete
transition-property	Specifies the name of the CSS property the transition effect is for.
transition-timing-function	Specifies the speed curve of the transition effect. • **ease** specifies a transition effect with a slow start, then fast, then end slowly (this is default) • **linear** specifies a transition effect with the same speed from start to end • **ease-in** specifies a transition effect with a slow start • **ease-out** specifies a transition effect with a slow end • **ease-in-out** specifies a transition effect with a slow start and end

Example1

```
<!DOCTYPE html>
<html>
<head>
<style>
img {
background: red;
transition: width 2s;
}
img:hover {
width: 500px;
}
</style>
</head>
<body>
<h1>The transition Property</h1>
<imgsrc="car1.jpg" width="100" height="100">
<p>Hover over the img element, to see the transition effect</p>
</body>
</html>
```

Example2

```
<!DOCTYPE html>
<html>
<head>
<style>
        .d1 {
        width: 100px;
        height: 100px;
        background: #ff33cc;
        transition: width 500ms;
        }
        .d1:hover {
        width: 500px;
        }
        .d2 {
        width: 100px;
        height: 100px;
        background: red;
        transition: height 2s;
        }
        .d2:hover {
        height: 300px;
        }
</style>
</head>
```

```
<body>
        <h1>The transition Property</h1>
        <p>get mouse over the div element, to see the transition width effect</p>
        <div class="d1"></div>

        <br><br>

        <p>get mouse over the div element, to see the transition height effect</p>
        <div class="d2"></div>
</body>
</html>
```

Example3

```
<!DOCTYPE html>
<html>
<head>
<style>
        div {
        width: 100px;
        height: 100px;
        background: #ff3300;
position: relative;
        color: #00ff33;
        font-size: 25px;
        transition: background-color 2s, font-size 2s, width 2s;
        }
        div:hover {
        background-color: blue;
        font-size: 100px;
        width: 400px;
        }
</style>
</head>
<body>
        <h1>The transition with mutiple Properties</h1>

        <div>See effect</div>
</body>
</html>
```

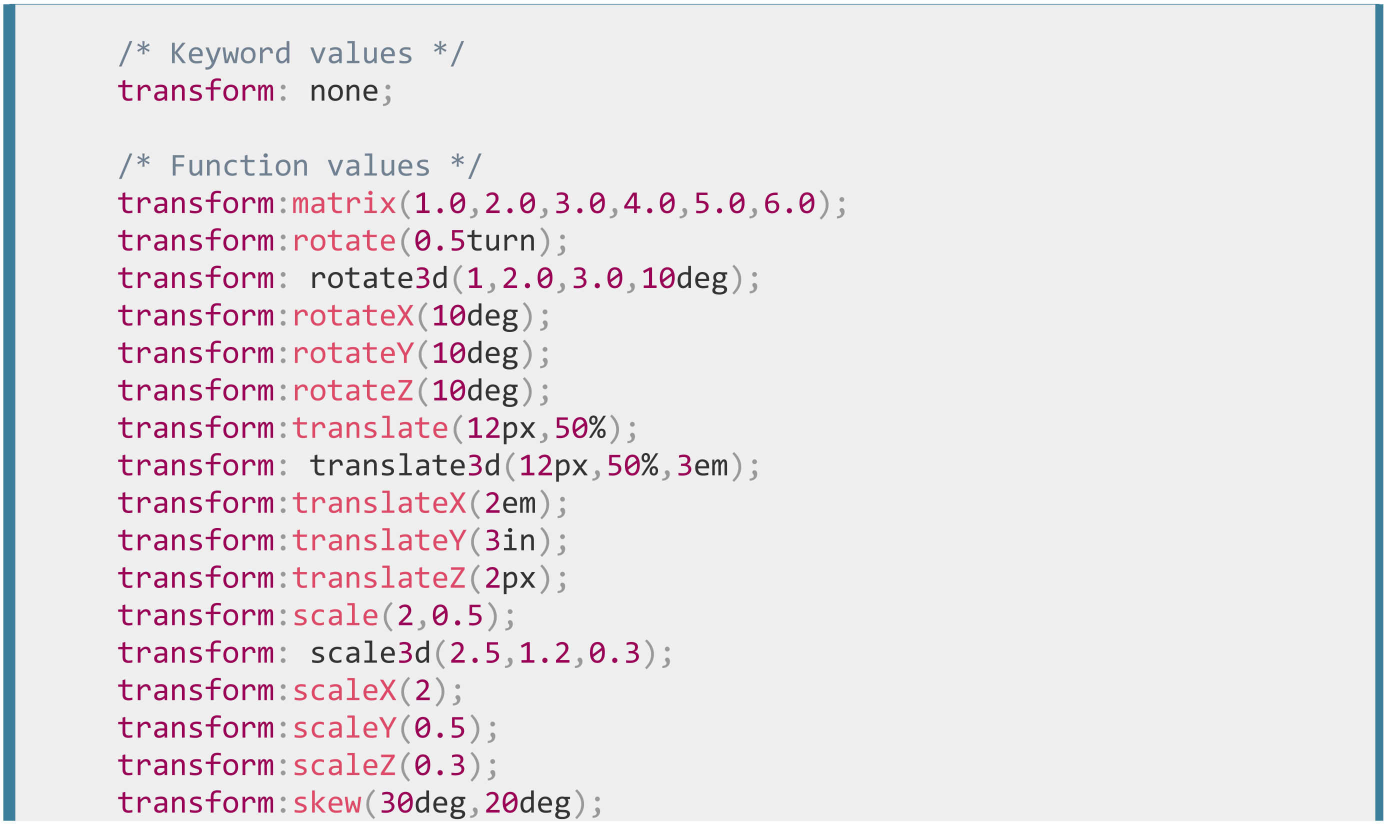
CSS Transforms

Transforms are used to display the element in a different visual dimension.

Types of Transforms:

- matrix(1, 2, 3, 4, 5, 6…);
- translate(x, y) translateX(x) translateY(y) translateZ(z)
- scale(1,1); scaleX(x) scaleY(y) scaleZ(z)
- rotate(+-deg) rotateX(x-deg) rotateY(y-deg) rotateZ(z-deg)
- skew(Xdeg, Ydeg) skewX(deg) skewY(deg) skewZ(deg)

- scale(0.5) translate(-100%, -100%);



```
/* Keyword values */
transform: none;

/* Function values */
transform:matrix(1.0,2.0,3.0,4.0,5.0,6.0);
transform:rotate(0.5turn);
transform: rotate3d(1,2.0,3.0,10deg);
transform:rotateX(10deg);
transform:rotateY(10deg);
transform:rotateZ(10deg);
transform:translate(12px,50%);
transform: translate3d(12px,50%,3em);
transform:translateX(2em);
transform:translateY(3in);
transform:translateZ(2px);
transform:scale(2,0.5);
transform: scale3d(2.5,1.2,0.3);
transform:scaleX(2);
transform:scaleY(0.5);
transform:scaleZ(0.3);
transform:skew(30deg,20deg);
```

```
transform:skewX(30deg);
transform:skewY(1.07rad);

/* Multiple function values */
transform:translateX(10px)rotate(10deg)translateY(5px);
transform:translate(10px,0,20px)rotateY(3deg);
```

Example1

```
<!DOCTYPE html>
<html>
<head>
<style>
        img {
        border: solid red;
        transform: rotate(50deg);
        width: 140px;
        height: 100px;
        }

        div {
        border: solid red;
        transform: translate(30px, 20px) rotate(20deg);
        width: 140px;
        height: 60px;
        }
</style>
</head>
<body>
        <h1>The transition with mutiple Properties</h1>

        <div>See effect</div><br>
        <imgsrc="car3.png">

</body>
</html>
```

Example2

```
<!DOCTYPE html>
<html>
<head>
<style>
        div {
        width: 200px;
        height: 200px;
        background-color:#00ccff;
        margin-left: 150px;
```

```
	box-shadow: 10px 10px10px10px #ff0000;
	}
	div:hover{
	background-color:#0099ff;
	transform: translate(30px, 30px); /* scale(1.5); */
	}
</style>
</head>
<body>
	<h1>The transform Properties</h1>
	<div>See effect</div>
</body>
</html>
```

Example3

```
<!DOCTYPE html>
<html>
<head>
<style>
	div {
	width: 200px;
	height: 200px;
	background-color:#6633ff;
	margin-left: 100px;
	box-shadow: 5px 5px5px5px #ff0000;
	transition: transform 1s;
	}
	div:hover{
	background-color:#0099ff;
	transform: rotate(90deg) scale(1.5);
	}
</style>
</head>
<body>
	<h1>Transform with Transition Properties</h1>
	<div>See effect</div>
</body>
</html>
```

Animations

- Animations are "group of transitions/transform/properties", which will be performed one after another.
- Transition contains two points only (starting point, ending point).
- Animation contains multiple points of milestones.

Syn:

```
    @keyframes Ani-name
    {
        from{ prop:value; prop:value; prop:value;…}← old status
        to{prop:new-value; prop:new-value; prop:new-value;…}← new status
    }

    @keyframesAni-name
    {
        0%{ prop:value; prop:value; prop:value;…}
        25%{ prop:value; prop:value; prop:value;…}
        33%{ prop:value; prop:value; prop:value;…}
        75%{ prop:value; prop:value; prop:value;…}
        100%{ prop:value; prop:value; prop:value;…}
    }
```

Mapping to tags:

```
    Selector{
            reg attributes
            animation-name:ani-name;
            animation-duration: Ns;
            animation-direction: normal/reverse/alter;
            animation-delay:Ns;
            animation-iteration-count:count/infinite;    default is 1
    }
```