

Transform :- Allow us to
→ translate
→ Rotate
+ Scale
→ Skew

1) translate() :- moves an element from its current position to x_2, y_2

div {
-ms-transform: translate(50, 100);
-webkit-transform: translate(50, 100);
transform: translate(50px, 100px);

[-ms- → IE
-webkit → Safari]

2) Rotate() :- Rotate(20deg)

+ve degree :- clockwise
-ve degree :- anticlockwise

3) Scale() :- increases or decreases the size

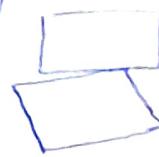
scale(2, 3)

$\frac{1}{2}$ h (three times increase in height)

(double the width)

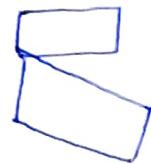
scale(.5, .5) → half

4) SkewX() :- Skews an element along the x-axis by the given angle.



transform: `SkewX(20deg);`

SkewY() → Along y axis



Skew(x angle, y angle)

5) `matrix (scaleX(), scaleY(), skewX, scaleY, translateX(), translateY())`

transform: `matrix(1, -0.3, 0, 1, 0, 0)`

(Combines 2D transform methods into one)

3D Transform

- 1) `rotateX()`
- `rotateY()`
- `rotateZ()`

transform: `rotateX(150deg)`

- Transitions :- Must specify two things
 - 1) CSS property on which effect has to be added
 - 2) Duration of effect

Example transition: width 2s

Transition effect will start when the specified CSS property changes value

Example:-

<style>

```
div {
    width: 100px;
    height: 100px;
    background: red;
    -webkit-transition: width 2s;
    transition: width 2s;
}
```

}

div:hover

```
{ width: 300px; }
```

}

</style>

Multiple properties :-

```
transition: width 2s, height 4s;
```

Speed curve :-

transition-timing-function :-

- (def.) ease → slow start, then fast, end
linear → same speed
ease-in → slow start
ease-out → slow end
ease-in-out → slow start & end

Example :-

div {

width: 100; height: 100; background-color: red;
-webkit-transition: width 2s;
transition: width 2s;

}

#div { -webkit-transition-timing: linear }

div: hover

{ width: 300px

}

transition-delay :- delays in seconds

transition + transform

transition: width 2s, height 2s, transform 2s;

div: hover { width: 300px;
height: 300px;
transform: rotate(180deg);
}

Animation

Gradually changes from one style to another

@keyframes :-

@keyframes ex :-

```
from { background-color: red; }
      to { background-color: yellow; }
```

div

```
{
    width: 100px;
    height: 100px;
    animation-name: ex;
    animation-duration: 4s;
    animation-iteration-count: 3;
    animation-direction: reverse;
```

@-webkit-keyframes ex :-

```
from { background-color: red; }
      to { background-color: yellow; }
```

3

animation-direction:- reverse / alternate

% format :-

@keyframes ex :-

```
0% { background-color: red; left: 0; top: 0; }
     : yellow; left: 200px; top: 100px}
20% {           : green; left: 200px; top: 100px}
50% {           : blue; left: 0; top: 0; }
75% {           : }
```

(3)

animation-name

animation-duration

animation-iteration-count: <not> / infinite

animation-direction: reverse / alternate

animation-timing-function: ease / linear /
ease-in / ease-out /
ease-in-out

Image

border-radius :- 8px

: 50%

Thumbnail images :-

img { border: 1px solid #ddd;

border-radius: 4px;

padding: 5px;

width: 150px;

}

img:hover { box-shadow: 0 0 2px 1px

rgba(0, 140, 186,
0.5) }

margin: auto

img { height: auto (responsive)
display: block; } + center
width: 50%
}

border-image: url("image") 30 stretch;
border-image-source, border-image-width

border-image-repeat: repeat /
repeat-x / stretch

box-shadow: none

h-shadow vshadow blur spread color
size
(of shadow)

inset

(changes the
shadow from an
offset outer to
an inner shadow)

Filter:-

filter: grayscale(100%)

blur(px) → default: 0

brightness (%) 0 → black 100% original

contrast (%) → 0-black 100% original
>100% less contrast.

hue-rotate(deg) hue-rotate(90deg)

invert (%) invert(100%)

0% transparent opacity(%) opacity(30%)

100% original sepia(%) sepia(100%)

saturate(<100%) → desaturate(%)

drop-shadow(8px 8px 10px red,
h-shadow v-shadow)

flipH

flipV

(optional
blue effect,
0+def.)

text-shadow: 2px 2px red;

textbox-shadow: 2px 2px 4px black, 2px 3px red

<style>

* div { width: 100px; height: 100px;

background: red;

transition: width 2s, background-color 5s}

#div1 { transition-timing-function: linear }

div : hover { width: 300px; background-color: green }

~~</div>~~

</style>

<body>

<div id="div1" > linear transition </div>

Style Sheet

Background Properties

Property

background

background-attachment

background-color

background-image

background-position

Description

Sets all the background properties in one declaration

Sets whether a background image is fixed or scrolls with the rest of the page

Sets the background color of an element

Sets the background image for an element

Sets the starting position of a background image

top left If you only specify one keyword, the second value will be "center".

top Default value is: 0% 0%

center

top right

center

left

center

center

center

right

bottom

left

bottom

center

bottom

right

x% y% The first value is the horizontal position and the second value is the vertical. The top left corner is 0% 0%. The right bottom corner is 100% 100%. If you only specify one value, the other value will be 50%.

xpos The first value is the horizontal position and the second value is the vertical. The top left corner is 0 0. Units can be pixels (0px 0px) or any other CSS units. If you only specify one value, the other value will be 50%. You can mix % and positions

background-repeat Sets how a background image will be repeated

Border and Outline Properties

Property

border

Description

Sets all the border properties in one declaration

border-width Specifies the width of the border. See border-width for possible values

border-style Specifies the style of the border. See border-style for possible values

border-color Specifies the color of the border. See border-color for possible values

border-bottom Sets all the bottom border properties in one declaration

border-bottom-color Sets the color of the bottom border

border-bottom-style Sets the style of the bottom border

<u>border-bottom-width</u>	Sets the width of the bottom border
<u>border-color</u>	Sets the color of the four borders
<u>border-left</u>	Sets all the left border properties in one declaration
<u>border-left-color</u>	Sets the color of the left border
<u>border-left-style</u>	Sets the style of the left border
<u>border-left-width</u>	Sets the width of the left border
<u>border-right</u>	Sets all the right border properties in one declaration
<u>border-right-color</u>	Sets the color of the right border
<u>border-right-style</u>	Sets the style of the right border
<u>border-right-width</u>	Sets the width of the right border
<u>border-style</u>	Sets the style of the four borders
none	Specifies no border
hidden	The same as "none", except in border conflict resolution for table elements
dotted	Specifies a dotted border
dashed	Specifies a dashed border
solid	Specifies a solid border
double	Specifies a double border
groove	Specifies a 3D grooved border. The effect depends on the border-color value
ridge	Specifies a 3D ridged border. The effect depends on the border-color value
inset	Specifies a 3D inset border. The effect depends on the border-color value
outset	Specifies a 3D outset border. The effect depends on the border-color value
<u>border-top</u>	Sets all the top border properties in one declaration
<u>border-top-width</u>	Specifies the width of the top border. See <u>border-top-width</u> for possible values
<u>border-top-style</u>	Specifies the style of the top border. See <u>border-top-style</u> for possible values
<u>border-top-color</u>	Specifies the color of the top border. See <u>border-top-color</u> for possible values
<u>border-top-color</u>	Sets the color of the top border
<u>border-top-style</u>	Sets the style of the top border
<u>border-top-width</u>	Sets the width of the top border
<u>border-width</u>	Sets the width of the four borders
<u>outline</u>	Sets all the outline properties in one declaration
<u>outline-color</u>	Sets the color of an outline
<u>outline-style</u>	Sets the style of an outline
<u>outline-width</u>	Sets the width of an outline

Dimension Properties

Property

height

max-height

max-width

min-height

min-width

width

Font Properties

Property

font

font-family

font-size

font-style

font-variant

font-weight

List Properties

Property

list-style

list-style-image

list-style-position

list-style-type

Description

Sets the height of an element

Sets the maximum height of an element

Sets the maximum width of an element

Sets the minimum height of an element

Sets the minimum width of an element

Sets the width of an element

Description

Sets all the font properties in one declaration

Specifies the font family for text

Specifies the font size of text

Specifies the font style for text

normal The browser displays a normal font style. This is default

italic The browser displays an italic font style

oblique The browser displays an oblique font style

Specifies whether or not a text should be displayed in a small-caps font

~~Specifies the weight and style~~

Description

Sets all the properties for a list in one declaration

Specifies an image as the list-item marker

Specifies if the list-item markers should appear inside or outside the content flow

Specifies the type of list-item marker

none No marker

circle The marker is a circle

disc The marker is a filled circle. This is default

square The marker is a square

armenian The marker is traditional Armenian numbering

decimal The marker is a number

decimal-leading-zero The marker is a number padded by initial zeros (01, 02, 03, etc.)

georgian The marker is traditional Georgian numbering (an, ban, gan, etc.)

lower-alpha The marker is lower-alpha (a, b, c, d, e, etc.)

lower-greek The marker is lower-greek (alpha, beta, gamma, etc.)

lower-latin	The marker is lower-latin (a, b, c, d, e, etc.)
lower-roman	The marker is lower-roman (i, ii, iii, iv, v, etc.)
upper-alpha	The marker is upper-alpha (A, B, C, D, E, etc.)
upper-latin	The marker is upper-latin (A, B, C, D, E, etc.)
upper-roman	The marker is upper-roman (I, II, III, IV, V, etc.)

Margin Properties

Property

margin
margin-bottom
margin-left
margin-right
margin-top

Description

Sets all the margin properties in one declaration
 Sets the bottom margin of an element
 Sets the left margin of an element
 Sets the right margin of an element
 Sets the top margin of an element

Padding Properties

Property

padding
~~padding-bottom~~
padding-left
padding-right
padding-top

Description

Sets all the padding properties in one declaration
length Specifies the padding in px, pt, cm, etc. Default value is 0px
% Specifies the padding in percent of the width of the containing element
~~Sets the bottom padding of an element.~~
 Sets the left padding of an element
 Sets the right padding of an element
 Sets the top padding of an element

Positioning Properties

Property

bottom
clear
clip
cursor

Description

Sets the bottom margin edge for a positioned box
 Specifies which sides of an element where other floating elements are not allowed
 Clips an absolutely positioned element
 Specifies the type of cursor to be displayed
URL A comma separated of URLs to custom cursors. **Note:** Always specify a generic cursor at the end of the list, in case none of the URL-defined cursors can be used
 auto Default. The browser sets a cursor
 crosshair The cursor render as a crosshair
 default The default cursor
 e-resize The cursor indicates that an edge of a box is to be moved right (east)
 help The cursor indicates that help is available
 move The cursor indicates something that should be moved

<u>n-resize</u>	The cursor indicates that an edge of a box is to be moved up (north)
<u>ne-resize</u>	The cursor indicates that an edge of a box is to be moved up and right (north/east)
<u>nw-resize</u>	The cursor indicates that an edge of a box is to be moved up and left (north/west)
<u>pointer</u>	The cursor render as a pointer
<u>progress</u>	The cursor indicates that the program is busy (in progress)
<u>s-resize</u>	The cursor indicates that an edge of a box is to be moved down (south)
<u>se-resize</u>	The cursor indicates that an edge of a box is to be moved down and right (south/east)
<u>sw-resize</u>	The cursor indicates that an edge of a box is to be moved down and left (south/west)
<u>text</u>	The cursor indicates text
<u>w-resize</u>	The cursor indicates that an edge of a box is to be moved left (west)
<u>wait</u>	The cursor indicates that the program is busy
<u>display</u>	Specifies the type of box an element should generate : <i>inline, block, flex</i>
<u>float</u>	Specifies whether or not a box should float
<u>left</u>	Sets the left margin edge for a positioned box
<u>overflow</u>	Specifies what happens if content overflows an element's box
	<u>visible</u> The overflow is not clipped. It renders outside the element's box. This is default
	<u>hidden</u> The overflow is clipped, and the rest of the content will be invisible
	<u>scroll</u> The overflow is clipped, but a scroll-bar is added to see the rest of the content
	<u>auto</u> If overflow is clipped, a scroll-bar should be added to see the rest of the content
<u>position</u>	Specifies the type of positioning for an element
<u>right</u>	Sets the right margin edge for a positioned box
<u>top</u>	Sets the top margin edge for a positioned box
<u>visibility</u>	Specifies whether or not an element is visible
	<u>visible</u> The element is visible. This is default
	<u>hidden</u> The element is invisible (but still takes up space)
<u>z-index</u>	Sets the stack order of an element

Print Properties

Property

orphans

Description

Sets the minimum number of lines that must be left at the bottom of a page when a page break occurs inside an element

page-break-after

Sets the page-breaking behavior after an element

		auto Default. Insert a page break after the element if necessary
	always	Insert a page break after the element
	avoid	Avoid inserting a page break after the element
	left	Insert page breaks after the element until it reaches a blank left page
	right	Insert page breaks after the element until it reaches a blank right page
<u>page-break-before</u>		Sets the page-breaking behavior before an element
<u>page-break-inside</u>		Sets the page-breaking behavior inside an element
widows		Sets the minimum number of lines that must be left at the top of a page when a page break occurs inside an element

Table Properties

Property	Description
<u>border-collapse</u>	Specifies whether or not table borders should be collapsed collapse Borders are collapsed into a single border when possible (border-spacing and empty-cells properties will be ignored) separate Borders are detached (border-spacing and empty-cells properties will not be ignored). This is default
<u>border-spacing</u>	Specifies the distance between the borders of adjacent cells
<u>caption-side</u>	Specifies the placement of a table caption top Puts the caption above the table. This is default bottom Puts the caption below the table
<u>empty-cells</u>	Specifies whether or not to display borders and background on empty cells in a table hide No background or borders are shown on empty cells show Background and borders are shown on empty cells. This is default
<u>table-layout</u>	Sets the layout algorithm to be used for a table auto Automatic table layout algorithm (this is default): <ul style="list-style-type: none"> • The column width is set by the widest unbreakable content in the cells • Can be slow, since it needs to read through all the content in the table, before determining the final layout

fixed Fixed table layout algorithm:

- The horizontal layout only depends on the table's width and the width of the columns, not the contents of the cells
- Allows a browser to lay out the table faster than the automatic table layout
- The browser can begin to display the table once the first row has been received

Text Properties

Property
color
direction

letter-spacing
line-height
text-align

text-decoration

text-indent

text-transform

vertical-align
white-space
word-spacing

Description

Sets the color of text

Specifies the text direction/writing direction

ltr The writing direction is left-to-right. This is default

rtl The writing direction is right-to-left

Increases or decreases the space between characters in a text

Sets the line height

Specifies the horizontal alignment of text

left Aligns the text to the left

right Aligns the text to the right

center Centers the text

justify Stretches the lines so that each line has equal width (like in newspapers and magazines)

Specifies the decoration added to text

none Defines a normal text. This is default

underline Defines a line below the text

overline Defines a line above the text

line-through Defines a line through the text

blink Defines a blinking text

Specifies the indentation of the first line in a text-block

Controls the capitalization of text

none No capitalization. The text renders as it is. This is default

capitalize Transforms the first character of each word to uppercase

uppercase Transforms all characters to uppercase

lowercase Transforms all characters to lowercase

Sets the vertical alignment of an element

Specifies how white-space inside an element is handled

Increases or decreases the space between words in a text

CSS Pseudo-classes/elements

Property

:active
:after
:before
:first-child
:first-letter
:first-line
:focus

Description

Adds a style to an element that is activated

Adds content after an element

Adds content before an element

Adds a style to an element that is the first child of another element

Adds a style to the first character of a text

Adds a style to the first line of a text

Adds a style to an element that has keyboard input focus

:hover
:lang
:link
:visited

Adds a style to an element when you mouse over it
Adds a style to an element with a specific lang attribute
Adds a style to an unvisited link
Adds a style to a visited link

Example:

```
h1:after
{
content:url(beep.wav);
}

a:link {color:#FF0000} /* unvisited link */
a:visited {color:#00FF00} /* visited link */
a:hover {color:#FF00FF} /* mouse over link */
a:active {color:#0000FF} /* selected link */

div
{
direction:rtl;
}

p
{
border:5px solid red;
}
```

- **padding:10px 5px 15px 20px;**
 - top padding is 10px
 - right padding is 5px
 - bottom padding is 15px
 - left padding is 20px
- **padding:10px 5px 15px;**
 - top padding is 10px
 - right and left padding are 5px
 - bottom padding is 15px
- **padding:10px 5px;**
 - top and bottom padding are 10px
 - right and left padding are 5px
- **padding:10px;**
 - all four paddings are 10px

; Valid ; Invalid ; nth-child
; nth-last-child ; first-of-type
; nth-of-type ; last-of-type
; checked ; enabled
; disabled ; dir-langs

Content

a string
an image
nothing ""
Content(6)

div::after
{ Content;"he"; }

A E :- Any E element that is a descendant of an A element (i.e. child or a child of child)

A7E Any G Element that is a child (direct or descendent)
E::first-child Any E Element that is a child of A
B+E :- Any E Element that is next sibling of B element

nth-child ($a_n + b$)

$a_n + b - 1$ (siblings)

$n + 0$ or simply $n \rightarrow$ match
every child element 1, 2, 3, ...

$2n + 0 \rightarrow 2n \rightarrow$ match elements
2, 4, 6, 8, ...

$2n + 1 \rightarrow$ match 1, 3, 5, 7, ...

$3n + 4 \rightarrow$ match 4, 7, 10, 13, ...

Hover (with child display)

P {

display: none;

background-color: yellow;

padding: 20px;

}

div: hover P

{

display: block;

}

<div> Hover on me </div>

<p> Hello!! ... </p>

</div>

<div>

 one → li: first-child

 two → li: nth-child(2)

 three → li: last-child

<div> abc </div> → div div: first-of-type

<p> para </p>

<div> def </div> → div div: last-of-type

</div> <p> para2 </p> → p: nth-of-type(2)

p: nth-of-type(2n+1) { text-align: left }

p: nth-of-type(2n) { text-align: right }

<div>

<p> first para </p>

<p> second </p>

<p> third </p>

</div>

first-

second

third

P i: first-child { ... }

1st <i> element in all <p>

P: first-child { ... }

all <i> elements in <p> that are
1st child of another element

invalid

=====
<style>
input: invalid { border: 2px solid
red; }
</style>

<input type="email" value="Support">

Counter body::before { counter-increment: my-counter; }

body { counter-reset: my-counter; }

p::before { counter-increment: my-counter; content: counter(my-counter); }

OR

p::before { counter: "Table" my-counter; counter-increment: my-counter; }

display :- inline :- Default : as an inline element

block :- as a block like <p>

flex :-

inline-block

inline-flex

inline-table

list-item

run-in

table

table-column-group

table-header-group (as ~~thead~~)

table-row-group as <tbody>

table-cell as <td>

table-column <as >

none :- will not display

(attribute = value)

[^ = value] → Starts with

[\$ = value] → Ends with

[* = "value"] - Substring

(~ = "Value"] Containing value

All CSS Pseudo Classes

Selector	Example	Example description
<u>:active</u>	a:active	Selects the active link
<u>:checked</u>	input:checked	Selects every checked <input> element
<u>:disabled</u>	input:disabled	Selects every disabled <input> element
<u>:empty</u>	p:empty	Selects every <p> element that has no children
<u>:enabled</u>	input:enabled	Selects every enabled <input> element
<u>:first-child</u>	p:first-child	Selects every <p> elements that is the first child of its parent
<u>:first-of-type</u>	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
<u>:focus</u>	input:focus	Selects the <input> element that has focus
<u>:hover</u>	a:hover	Selects links on mouse over
<u>:in-range</u>	input:in-range	Selects <input> elements with a value within a specified range
<u>:invalid</u>	input:invalid	Selects all <input> elements with an invalid value
<u>:lang(<i>language</i>)</u>	p:lang(it)	Selects every <p> element with a lang attribute value starting with "it"
<u>:last-child</u>	p:last-child	Selects every <p> elements that is the last child of its parent
<u>:last-of-type</u>	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
<u>:link</u>	a:link	Selects all unvisited links

<u>:valid</u>	input:valid	Selects all <input> elements with a valid value
<u>:visited</u>	a:visited	Selects all visited links

All CSS Pseudo Elements

Selector	Example	Example description
::after	p::after	Insert content after every <p> element
::before	p::before	Insert content before every <p> element
::first-letter	p::first-letter	Selects the first letter of every <p> element
::first-line	p::first-line	Selects the first line of every <p> element
::selection	p::selection	Selects the portion of an element that is selected by a user

```
p::after {
    content: " - Remember this";
    background-color: yellow;
    color: red;
    font-weight: bold;
}
```

```
input:required {
    background-color: yellow;
}
```

Specify a background color for every <p> element that is the only child of its parent:

```
p:only-child {
    background: #ff0000;
```

Specify a background color for every <p> element that is the second p element of its parent:

```
p:nth-of-type(2) {
    background: red;
```

Specify a background color for every <p> element that is the second child of its parent:

```
p:nth-child(2) {
    background: red;
```

<u>:not(selector)</u>	:not(p)	Selects every element that is not a <p> element
<u>:nth-child(n)</u>	p:nth-child(2)	Selects every <p> element that is the second child of its parent
<u>:nth-last-child(n)</u>	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
<u>:nth-last-of-type(n)</u>	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
<u>:nth-of-type(n)</u>	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
<u>:only-of-type</u>	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
<u>:only-child</u>	p:only-child	Selects every <p> element that is the only child of its parent
<u>:optional</u>	input:optional	Selects <input> elements with no "required" attribute
<u>:out-of-range</u>	input:out-of-range	Selects <input> elements with a value outside a specified range
<u>:read-only</u>	input:read-only	Selects <input> elements with a "readonly" attribute specified
<u>:read-write</u>	input:read-write	Selects <input> elements with no "readonly" attribute
<u>:required</u>	input:required	Selects <input> elements with a "required" attribute specified
<u>:root</u>	root	Selects the document's root element
<u>:target</u>	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)