

CSS Gradients

CSS gradients let you display smooth transition b/w 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)

Linear Gradient

To create a linear gradient you must define at least two color stops. Color stops are the colors you want to render smooth transition 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, color1, color2, ...);

Directions:

* Top to Bottom (Default)

Linear gradient that starts at the top

and ends at the bottom.

* Left to Right

If starts at the left and ends at the bottom.

* Diagonal

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

It starts at the top left and goes

to bottom right.

* Using Angles

You can also define the angles instead of predefined direction.

Syntax

background-image: linear-gradient(angle, color1, color2);

Radial Gradient

Example:

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, color1, color2 ---);

⇒ By default shape is ellipse.

Repeating radial gradient

By using this you can also repeat the radial gradient.

Syntax

background-image: repeating-radial-gradient(color1, color2);

Conic Gradient

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

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

Syntax

background-image: conic-gradient(color1, color2--);

⇒ By default, angle is 0° Odeg and position is center.

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

CSS Shadow Effects

With CSS you can add shadow to text

and to elements.

* text-shadow

* box-shadow

Text Shadow

The CSS `text-shadow` property applies shadow to text.

Syntax

`text-shadow: 2px 2px value1 value2 value3;`

horizontal shift vertical shift blur effect

Ex P of

text-shadow: 2px 2px red;

y

at the end of shadow applied on

Hello

⇒ By default shadow color is same as text color.

Box Shadow

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

Syntax

box-shadow: value1 value2 value3,
 [horizontal shift] [vertical shift] [Blur]
 [shift] [shift] [effect]

⇒ By default shadow color is same as text color.

Q. How can we add border using shadows?

We can add border using shadows by putting horizontal shift value and vertical shift value as 1px.

- ⇒ Multiple shadows can be added using comma.
- ⇒ Color of the shadow can be changed.
- ⇒ Spread radius can be changed.

CSS Dimension Properties:

* Height

The height property is used to set the height of the box.

* Width

The width property is used to set the width of a box.

* Max-height

It is used to set maximum height that a box can attain.

* Min-height

It is used to set minimum height that a box can be.

* Max-width

It is used to set maximum width that a box can be.

* Min-width

It is used to set minimum width that a box can be.

NOTE - Do practical for better understanding.

Overflow Property

The overflow property specifies what should happen if content overflows an element's box.

This property specifies whether to clip content or to add scrollbar when an element's content is too big to fit in a specified area.

NOTE: The overflow property only works for block elements with a specified height.

Syntax

`overflow: visible | hidden | clip | scroll | auto;`

Values in Overflow Property

* Visible

* Hidden

* Clip

* Scroll

* Auto

CSS Position Property

The position property specifies the type of positioning method used for an element.

* Static (Default value)

An element with position: static; is not positioned in any special way; it is always positioned according to the normal flow of page.

* Relative

An element with position: relative; is positioned relative to its normal position.

Setting the top, right, bottom and left properties of a relative-positioned element will cause it to be adjusted away from its normal position.

* Fixed

An element with position: fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled.

The top, right, bottom and left properties are used to position the element.

* Absolute

An element with position: absolute; is positioned relative to the nearest positioned ancestor.

However, if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with the page scroll.

NOTE: Absolute positioned elements are removed from the normal flow, and can overlap elements.

* Sticky

The element is positioned based on the user's scroll position.

A sticky element toggles b/w relative & fixed, depending on the scroll position.

IMP ⇒ Do practical for better understanding.

2D Transform

CSS transform allow you to move, rotate, scale, and skew elements.

* With transform property you can use the following methods:

* translate()

* rotate()

* scaleX()

lowercase sans etchings benefitized

* scaleY()

uppercase sans etchings benefitized

* scale()

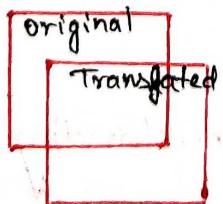
* skewX()

* skewY()

* skew()

* matrix()

* The translate() method

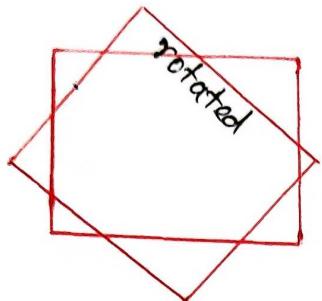


The translate() method moves an element from its current position according to its given parameters.

Syntax

transform: translate (value1, value2);

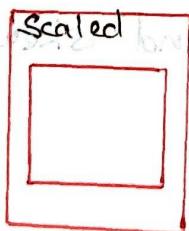
* The rotate() Method



The rotate() method rotates an element clockwise or counter-clockwise according to a given degree.

Ex `div { transform: rotate(20deg); }`

* The scale() Method



The scale() method increases or decreases the size of an element (according to the given parameters for the width and height).

Ex `div {`

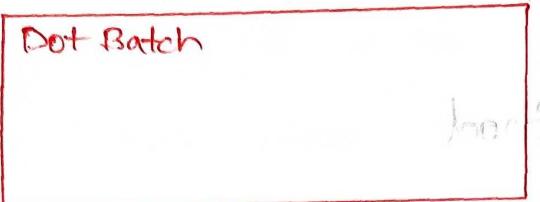
`transform: scale(2, 3);`

`scaleX scaleY`

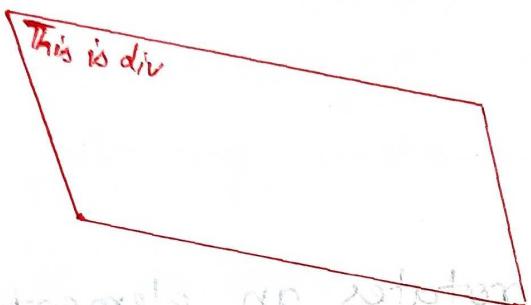
We can also use scaleX and scaleY separately.

Notes 2:

* The skew() Method



Transforms the element off-axis



Elements are rotated/bent on one or more axes.

The skew() method skews an element along with x and y-axis by the given angles.

Ex div {

transform: skew(20deg, 10deg);

y
↑
skewx ↑
 skewy

We can also use skewX and skewY separately.

* The matrix() Method

The matrix() method combines all the 2D transform methods into one.

The parameters are as follows: matrix(scaleX(), skewY(), skewX(), scaleY(), translateX(), translateY())

div {

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

3D Transforms (Refer to any video explanation)

It works on the z-axis like 2D transforms which works on x-axis and y-axis. Not possible to explain in 2D Notebook.