CSS Multiple Column

Property	Syntax	Example	What it does
column-count:	column-count:number;	column-count:4;	Specifies the number of columns an element should be divided into
column-gap	column-gap:number;	column-gap:20px;	Specifies the gap between the columns
column-rule-style:	column-rule-style:value	column-rule-style:dashed;	Specifies the style of the rule between columns. **(none, dotted, dashed, solid, double)** **default: none**
column-rule-color:	column-rule-color:color	column-rule-color:rgb(219, 12, 47);	Specifies the color of the rule between columns
column-rule-width:	column-rule-width:value	column-rule-width:7px;	Specifies the width of the rule between columns
Shorthand column-rule	Column-rule: width style color	Column-rule:1px solid cyan	Specify width,style and color of rule

CSS style images

Property	Syntax	Example	What it does
border-radius	border-radius:radius;	border-radius: 20px;	Sets the corner rounding for **all four corners**.
border-top-left-radius	border-top-left-radius: radius;	border-top-left-radius: 15px;	Rounds only the **top-left** corner.
border-top-right-radius	border-top-right-radius;	border-top-right-radius: **15px**;	Rounds only the **top-right** corner.
border-bottom-right-radius	border-bottom-right-radius: radius;	border-bottom-right-radius: 15px;	Rounds only the **bottom-right** corner.
border-bottom-left-radius	border-bottom-left-radius: radius;	border-bottom-left-radius: 15px;	Rounds only the **bottom-left** corner.
padding	padding:value	padding:5px	The border property and padding property are used to make a thumbnail image.
opacity	opacity:value	opacity:0.3	To make an image transparent, we have to use the opacity property. The value of this property lies between **(transparent) 0.0 to 1.0 (Solid).**
Responsive	max-width: %; height:auto	max-width:100%; height:auto	The image becomes flexible — it scales down when the screen is smaller, but won't stretch beyond its natural size.

2D Transform

Property	Syntax	Example	What it does
translate(x, y)	transform: translate(x, y);	transform: translate(50px, 30px);	Moves an element 50px right and 30px down.
translateX(n)	transform: translateX(n);	transform: translateX(100px);	Moves an element 100px horizontally (right).
translateY(n)	transform: translateY(n);	transform: translateY(-50px);	Moves an element 50px upward.
rotate(angle)	transform: rotate(angle);	transform: rotate(45deg);	Rotates the element 45° clockwise.
scale(x, y)	transform: scale(x, y);	transform: scale(1.5, 0.8);	Increases width 1.5× and decreases height to 0.8×.
scaleX(n)	transform: scaleX(n);	transform: scaleX(2);	Doubles the element’s width.
scaleY(n)	transform: scaleY(n);	transform: scaleY(0.5);	Reduces the element’s height by half.
skew(x, y)	transform: skew(x, y);	transform: skew(20deg, 10deg);	Slants an element 20° horizontally and 10° vertically.
skewX(angle)	transform: skewX(angle);	transform: skewX(20deg);	Skews an element horizontally by 20°.
skewY(angle)	transform: skewY(angle);	transform: skewY(20deg);	Skews an element vertically by 20°.

Note:

- The transform: scaleX(-1) property is used to flip the image horizontally.
- The transform: scaleY(-1) - mirror image vertically.
- The transform: scale(-1) property create a mirror image vertically as well as horizontally.

CSS Transitions Properties

Property	Syntax	Example	What it does
transition-property	transition-property: all | property | property1, property2, ...;	transition-property: background-color, transform;	Specifies which CSS properties will be animated during the transition.
transition-duration	transition-duration: time;	transition-duration: 0.5s;	Specifies how long the transition takes to complete.
transition-delay	transition-delay: time;	transition-delay: 0.3s;	Specifies how long to wait before starting the transition.
transition-timing-function	transition-timing-function: linear | ease | ease-in | ease-out | ease-in-out;	transition-timing-function: ease-in-out;	Controls the speed curve of the transition — how the animation progresses over time.
transition (shorthand)	transition: property duration timing-function delay;	transition: background-color 0.5s ease-in-out 0.2s;	Sets all transition properties in a single line (shorthand form).

Note: If the duration part is not specified, the transition will have no effect (default duration is 0s).

Transition Shorthand Property Breakdown

Property	Required in shorthand?	Default value
transition-property	No	all
transition-duration	Yes	0s
transition-timing-function	No	ease
transition-delay	No	0s

CSS Animation

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.

Note: In CSS keyframe, **from** = **0%** (start) and **to** = **100%** (end); using **percentages** lets you define multiple stages within the animation.

CSS Animation Properties

Property	Syntax	Example	What it does
animation-name	animation-name: keyframename;	animation-name: moveBox;	Specifies the name of the @keyframes to link to the element.
animation-duration	animation-duration: time;	animation-duration: 2s;	Specifies the time it takes for one animation cycle to complete.
animation-delay	animation-delay: time;	animation-delay: 1s;	Specifies the delay before the animation starts.
animation-timing-function	linear | ease | ease-in | ease-out | ease-in-out	animation-timing-function: ease-in-out;	Controls the speed curve of the animation over time.
animation-iteration-count	number | infinite	animation-iteration-count: infinite;	Specifies how many times the animation should repeat.
animation-direction	normal | reverse | alternate | alternate-reverse	animation-direction: alternate;	Specifies whether animation should run forward, backward, or alternate directions.

CSS gradient

Linear gradient

type	Syntax	Example	What it does
Linear Gradient (Default – Top to Bottom)	**background-image:** linear-gradient(color1, color2);	background-image: linear-gradient(purple, yellow);	Creates a smooth color transition from top to bottom (default direction).
Bottom to Top	background-image: linear-gradient(to top, color1, color2);	background-image: linear-gradient(to top, purple, yellow);	Creates a gradient starting from bottom to top.
Left to Right	background-image: linear-gradient(to right, color1, color2);	background-image: linear-gradient(to right, purple, yellow);	Creates a gradient moving from left to right.
Right to Left	background-image: linear-gradient(to left, color1, color2);	background-image: linear-gradient(to left, purple, yellow);	Creates a gradient moving from right to left.
Diagonal Gradient (to top left)	background-image: linear-gradient(to top left, color1, color2);	background-image: linear-gradient(to top left, purple, yellow);	Creates a diagonal gradient moving from bottom right to top left.
Diagonal Gradient (to bottom right)	background-image: linear-gradient(to bottom right, color1, color2);	background-image: linear-gradient(to bottom right, purple, yellow);	Creates a diagonal gradient from top left to bottom right.
Angle-based Gradient	background-image: linear-gradient(angle, color1, color2);	background-image: linear-gradient(90deg, pink, lightblue);	Uses degrees (0°, 90°, 180°,-90° etc.) to control gradient direction.
Evenly spaced multiple colors	linear-gradient(color1, color2, color3, ...)	linear-gradient(red, yellow, green);	Colors are evenly spaced between start and end points.
Non-evenly spaced colors	linear-gradient(color1 %, color2 %, ...)	linear-gradient(pink 20%, lightgreen 30%, lightblue 50%);	Uses percentage values to control how far each color extends.
Rainbow Gradient Example	---	linear-gradient(to right, red, orange, yellow, green, blue, indigo, violet);	Creates a rainbow-style horizontal gradient.

Radial gradient

type	Syntax	Example	What it does
Basic Radial Gradient	background-image: radial-gradient(color1, color2, color3);	background-image: radial-gradient(red, yellow, blue);	Creates a circular gradient radiating outward from the center.
Different Color Stops	background-image: radial-gradient(color1 %, color2 %, color3 %);	background-image: radial-gradient(yellow 10%, red 25%, pink 40%);	Defines uneven spacing for color transitions.
Ellipse (default)	background-image: radial-gradient(ellipse, color1, color2);	background-image: radial-gradient(purple, yellow, pink);	Creates an oval-shaped gradient that fills the container.
Circle Shape	background-image: radial-gradient(circle, color1, color2);	background-image: radial-gradient(circle, blue, yellow, pink);	Creates a perfectly circular gradient.

Conic Gradients

type	Syntax	Example	What it does
Basic Conic Gradient	background-image: conic-gradient(color1, color2, color3);	background-image: conic-gradient(red, yellow, blue);	Creates a circular gradient where colors rotate around the center point.
Using Degrees for Each Color	background-image: conic-gradient(color1 deg, color2 deg, ...);	background-image: conic-gradient(red 45deg, yellow 90deg, green 180deg);	Defines starting angles for each color transition.
Defined Start and End Degrees (Pie Chart Style)	background-image: conic-gradient(color1 start end, color2 start end, ...);	background-image: conic-gradient(red 0deg 90deg, yellow 90deg 180deg, blue 270deg);	Creates solid color segments similar to pie chart slices.
Rounded Pie Chart	border-radius: 50%;	background-image: conic-gradient(red, yellow, green, blue); border-radius: 50%;	Turns conic gradient into a circular pie chart.
With “from” Angle	background-image: conic-gradient(from angle, color1, color2, ...);	background-image: conic-gradient(from 90deg, red, yellow, green);	Rotates where the gradient starts based on the specified angle.

CSS Variable

CSS variables store reusable values that can be updated from one place.
They can be global (defined in :root for whole document) or local (defined inside a selector).
Values are accessed using the var() function, e.g. color: var(--b);.

Syntax:

Define variable globally

:root{--b: blue; }

Use of variable

P{ color:var(--b) }

All the elements of the document can use the variable.

Define variable locally and use it

P{ --b: blue; color:var(--b) }

It has a scope only for the p element.

CSS Media Queries

Concept	Description	Example
@media Rule	Applies CSS based on device features (screen size, orientation, etc.).	@media (max-width: 600px) { body {background: yellow;} }
Purpose	Enables responsive design to adapt websites for all devices.	—
Syntax	@media not	only mediatype and (mediafeature) { /* CSS */ }`
Common Media Types	all (default, all devices), screen (monitors, tablets, phones).	—

Feature / Operator	Used For	Example
width, height	Set styles based on viewport or device size.	(max-width: 700px)
orientation	Detects landscape or portrait mode.	(orientation: portrait)
and	Combine multiple conditions.	@media screen and (max-width: 700px)
or	Apply if any condition is true.	@media (min-width: 500px), (orientation: portrait)
not	Exclude a condition.	@media not (orientation: landscape)
only	Apply only to specific media types.	@media only screen and (min-width: 500px)