

CSS content Image Test Page

Demonstrates url() and gradient functions in the CSS content property on ::before and ::after pseudo-elements. Image values require display: inline-block with explicit width/height.

1 — ::before linear-gradient



to right red → blue
::before { content: linear-gradient(to right, red, blue); }



to bottom green → yellow
::before { content: linear-gradient(to bottom, green, yellow); }



45deg multi-stop
::before { content: linear-gradient(45deg, red, yellow, blue); }



hard-stop stripes
::before { content: linear-gradient(to right, red 50%, blue 50%); }

2 — ::before radial-gradient



circle center
::before { content: radial-gradient(circle, white, navy); }



ellipse default
::before { content: radial-gradient(red, blue); }



at 30% 30%
::before { content: radial-gradient(circle at 30% 30%, yellow, orange, red); }



closest-side
::before { content: radial-gradient(closest-side circle at 50% 50%, lime, teal); }

3 — ::before conic-gradient



default sweep
::before { content: conic-gradient(red, blue); }



pie chart
::before { content: conic-gradient(#e74c3c 0 25%, #3498db 25% 65%, #2ecc71 65% 100%); }



from 90deg
::before { content: conic-gradient(from 90deg, red, yellow, blue); }



color wheel
::before { content: conic-gradient(red, yellow, lime, cyan, blue, magenta, red); }

4 — ::after gradients and URL fallback



::after linear
::after { content: linear-gradient(to bottom, purple, orange); }



::after radial
::after { content: radial-gradient(circle, gold, crimson); }



::after conic
::after { content: conic-gradient(red 0 33%, blue 33% 66%, green 66% 100%); }

missing URL (no crash)
::before { content: url("nonexistent.png"); }

5 — Repeating variants



repeating-linear
::before { content: repeating-linear-gradient(45deg, red, blue 10px); }



repeating-radial
::before { content: repeating-radial-gradient(circle, red, blue 10px); }



repeating-conic
::before { content: repeating-conic-gradient(red 0 10deg, blue 10deg 20deg); }



linear baseline
::before { content: linear-gradient(to right, teal, pink); }

6 — Mixed: text content regression and image side-by-side

Text content (string literals, counters) must still work correctly alongside the new image code path.

• Text

text bullet
::before { content: "• "; color: red; }



gradient + text
::before { content: linear-gradient(to right, red, blue); display: inline-block; width: 40px; height: 20px; }

Text ★

::after text
::after { content: "★"; color: orange; }

Text

none (no output)
::before { content: none; }

7 — SVG url() content image

A url() content-image source can also be an SVG, rendered as real vector content (not rasterized) - same as background-image and list-style-image. Note the image paints at the SVG's own intrinsic viewBox size (32x32 here), independent of the box size reserved by display:inline-block's width/height.



::before peach
::before { content: url("data:image/svg+xml;base64,..."); }



::after peach
::after { content: url("data:image/svg+xml;base64,..."); }

Text

peach + text (::after)
::after { content: url("data:image/svg+xml;base64,..."); }

missing SVG (no crash)
::before { content: url("nonexistent.svg"); }