Gradients

- Linear gradient
- Radial gradient
- Conic gradient
- Repeating and mixing
- Resources

Imagine you've got a site to build and at the top, there's an intro with a heading, summary and a button. The designer has handed over a design which has a purple background for this intro. The only problem is the background features two shades of purple as a gradient. How do you do this?

{% Img src="image/VbAJIREinuYvovrBzzvEyZOpw5w1/0T5kGhQWJTLUclxSWipH.svg", alt="A dark to light purple gradient background with heading, paragraph and link", width="800", height="411" %}

You might initially think that you're going to need to export an image from your design tool for this, but you can use a `linear-gradient` instead.

{% Codepen { user: 'web-dot-dev', id: 'XWpzdoV' } %}

A gradient is an image and can be used anywhere images can be used, but it's created with CSS and is made up with colors, numbers and angles. CSS gradients allow you to create anything from a smooth gradient between two colors, right up to impressive artwork by mixing and repeating multiple gradients.

Linear gradient

{% BrowserCompat 'css.types.image.gradient.linear-gradient' %}

The `linear-gradient()` function generates an image of two or more colors, progressively. It takes multiple arguments, but in its simplest configuration, you can pass some colors like this and it will automatically split them evenly, while blending them.

```
.my-element {
  background: linear-gradient(black, white);
```

```
}
```

{% Codepen { user: 'web-dot-dev', id: 'mdRqPoZ', tab: 'css,result', height: 400 } %}

You can also pass an angle or keywords that represent an angle. If you choose to use keywords, specify a direction after the `to` keyword. This means if you want a gradient that is black and white, that runs from left (black) to right (white), you would specify the angle as `to right` as the first argument.

```
.my-element {
  background: linear-gradient(to right, black, white);
}
```

{% Codepen { user: 'web-dot-dev', id: 'zYNPqXW', tab: 'css,result', height: 400 } %}

A color stop value defined where a color stops and mixes with its neighbors. For a gradient starting with a dark shade of red running at a 45deg angle, at 30% of the size of the gradient changing to a lighter red: it looks like this.

```
.my-element {
  background: linear-gradient(45deg, darkred 30%, crimson);
}
```

{% Codepen { user: 'web-dot-dev', id: 'NWdwNZr', tab: 'css,result', height: 400 } %}

You can add as many colors and color stops as you like in a `linear-gradient()`, and you can layer gradients on top of each other by separating each gradient with a comma.

{% Codepen { user: 'web-dot-dev', id: 'abpVZbj', tab: 'css,result', height: 400 } %}

Radial gradient

{% BrowserCompat 'css.types.image.gradient.radial-gradient' %}

To create a gradient that radiates in a circular fashion, the `radial-gradient()` function steps in to help. It's similar to `linear-gradient()`, but instead of specifying an angle, you optionally specify a position and ending shape. If you just specify colors, the `radial-gradient()` will auto-select the position as `center` and select either a circle or ellipse, depending on the size of the box.

```
.my-element {
  background: radial-gradient(white, black);
}
```

{% Codepen { user: 'web-dot-dev', id: 'LYxOZEq', tab: 'css,result', height: 400 } %}

The gradient's position is similar to `background-position` using keywords and/or number values. The size of the radial gradient determines the size of the gradient's ending shape (circle or ellipse) and by default will be `farthest-corner`, which means it exactly meets the farthest corner of the box from the center. You can also use the following keywords:

- `closest-corner` will meet the closest corner to the center of the gradient.
- `closest-side` will meet the side of the box closest to the center of the gradient.
- `farthest-side` will do the opposite to `closest-side`.

You can add as many color stops as you like, just like with the `linear-gradient`. Likewise, you can add as many `radial-gradients` as you like too.

{% Codepen { user: 'web-dot-dev', id: 'MWJOepV', tab: 'css,result', height: 400 } %}

Conic gradient

{% BrowserCompat 'css.types.image.gradient.conic-gradient' %}

A conic gradient has a center point in your box and starts from the top (by default), and goes around in a 360 degree circle.

```
.my-element {
background: conic-gradient(white, black);
}
```

{% Codepen { user: 'web-dot-dev', id: 'qBRVNXG', tab: 'css,result', height: 400 } %}

The `conic-gradient()` function accepts position and angle arguments.

By default, the angle is 0 degrees which starts at the top, in the center. If you were to set the angle to be `45deg`, it would be the top right corner. The angle argument accepts any type of angle value, like the linear and radial gradients.

The position is center by default. As with radial and linear gradients, positioning can be keyword-based, or it can be defined with numeric values.

{% Codepen { user: 'web-dot-dev', id: 'vYgWKpO', tab: 'css,result', height: 400 } %}

You can add as many color stops as you want, like with other gradient types. A good use case for this capability, with conic gradients is rendering pie charts with CSS.

{% Codepen { user: 'web-dot-dev', id: 'mdRqExP', tab: 'css,result', height: 400 } %}

Repeating and mixing

Each type of gradient has a repeating type, too. These are `repeating-linear-gradient()`, `repeating-radial-gradient()` and `repeating-conic-gradient()`. They are similar to the non-repeating functions and take the same arguments. The difference is that if the defined gradient can be repeated to fill the box, based on both of their sizes, it will.

If your gradient doesn't appear to be repeating, you probably haven't set a length for one of the color stops. For example, you can create a striped background with a `repeating-linear-gradient` by setting color stop lengths.

```
.my-element {
  background: repeating-linear-gradient(
    45deg,
    red,
    red 30px,
    white 30px,
    white 60px
  );
}
```

{% Codepen { user: 'web-dot-dev', id: 'ExZbgdy', tab: 'css,result', height: 400 } %}

You can also mix gradient functions on `background` properties, as well as defining as many gradients as you like, just like you would with a background image. For example, you can mix multiple linear-gradients together, or two linear-gradients with a radial gradient.

{% Codepen { user: 'web-dot-dev', id: 'ExZqGxP', tab: 'css,result', height: 400 } %}

Resources

- Conic.css - a useful collection of conic gradients
- MDN guide to gradients
- Gradient generator

{% Assessment 'gradients' %}

□□□ #2
□ □□□ □□□ 2026-10-31 22:45:44 UTC
□ □□□ □□□ 2026-10-31 22:45:45 UTC