

# OKLCH - SCSS/SASS - Dark/Light Mode

Wenn deine Farben bereits als `oklch(...)` vorliegen, kannst du eine Funktion bauen, die einfach den **Lightness-Wert umkehrt**:

```
$$  
L_{\text{neu}} = 1 - L  
$$
```

Also aus:

```
oklch(0.3 0.15 30)
```

wird:

```
oklch(0.7 0.15 30)
```

## Einfache Sass-Funktion

```
@use "sass:color";  
@use "sass:math";  
  
@function invert-oklch-lightness($c) {  
  $space: color.space($c);  
  
  @if $space != oklch {  
    @error "invert-oklch-lightness erwartet eine OKLCH-Farbe, bekommen: #{$space}";  
  }  
  
  $l: color.channel($c, "lightness", $space: oklch);  
  $chroma: color.channel($c, "chroma", $space: oklch);  
  $hue: color.channel($c, "hue", $space: oklch);  
  
  $new-l: 1 - $l;
```

```
@return oklch($new-l $chroma $hue);
}
```

## Verwendung

```
$color: oklch(0.3 0.15 30);
$dark-mode: invert-oklch-lightness($color);
```

Ergebnis:

```
oklch(0.7 0.15 30)
```

## Beispiel im Einsatz

```
@use "sass:color";

@function invert-oklch-lightness($c) {
  @if color.space($c) != oklch {
    @error "Nur OKLCH-Farben sind erlaubt.";
  }

  $l: color.channel($c, "lightness", $space: oklch);
  $chroma: color.channel($c, "chroma", $space: oklch);
  $hue: color.channel($c, "hue", $space: oklch);

  @return oklch(1 - $l $chroma $hue);
}

$bg-light: oklch(0.92 0.03 240);
$bg-dark: invert-oklch-lightness($bg-light);

$saccent-light: oklch(0.3 0.15 30);
$saccent-dark: invert-oklch-lightness($saccent-light);

:root {
  --bg: #{$bg-light};
```

```
--accent: #{$accent-light};
}

@media (prefers-color-scheme: dark) {
  :root {
    --bg: #{$bg-dark};
    --accent: #{$accent-dark};
  }
}
```

## Wichtiger Hinweis

Das ist **mathematisch simpel**, aber **gestalterisch nicht immer optimal** ☐

Denn ein „guter“ Dark-Mode ist oft **nicht nur**:

1. Lightness invertieren
2. Chroma gleich lassen
3. Hue gleich lassen

Sondern häufig will man zusätzlich:

- **Chroma etwas reduzieren**, weil kräftige Farben auf dunklem Hintergrund schnell zu intensiv wirken
- bei sehr hellen Farben ein anderes Mapping verwenden
- Kontraste für Text gezielt anpassen

## Etwas robustere Variante

Wenn du willst, kannst du beim Invertieren gleichzeitig die Chroma leicht dämpfen:

```
@use "sass:color";
@use "sass:math";

@function dark-mode-oklch($c, $chroma-factor: 0.85) {
  @if color.space($c) != oklch {
    @error "Nur OKLCH-Farben sind erlaubt.";
  }
}
```

```
$l: color.channel($c, "lightness", $space: oklch);
$chroma: color.channel($c, "chroma", $space: oklch);
$hue: color.channel($c, "hue", $space: oklch);

$new-l: 1 - $l;
$new-chroma: $chroma * $chroma-factor;

@return oklch($new-l $new-chroma $hue);
}
```

Beispiel:

```
$color: oklch(0.3 0.15 30);
$dark: dark-mode-oklch($color);
```

Ergibt ungefähr:

```
oklch(0.7 0.1275 30)
```

---

# Falls du nur rohe Werte übergeben willst

Wenn du lieber direkt mit Zahlen statt mit einem Farbwert arbeitest:

```
@function invert-oklch($l, $c, $h) {
  @return oklch(1 - $l $c $h);
}
```

Verwendung:

```
$color: invert-oklch(0.3, 0.15, 30);
```

---

# Fazit

## Ja, absolut.

Für deinen konkreten Fall reicht im Prinzip genau das:

```
@function invert-oklch-lightness($c) {  
  $l: color.channel($c, "lightness", $space: oklch);  
  $chroma: color.channel($c, "chroma", $space: oklch);  
  $hue: color.channel($c, "hue", $space: oklch);  
  
  @return oklch(1 - $l $chroma $hue);  
}
```

Wenn du möchtest, kann ich dir als Nächstes auch noch eine Version schreiben, die:

1. **auch normale CSS-Strings wie `oklch(0.3 0.15 30)` verarbeitet**, oder
2. **eine ganze Sass-Map mit Theme-Farben automatisch in Dark-Mode-Werte umwandelt.**

---

Revision #3

Created 2026-05-23 17:21:25 UTC by art10m

Updated 2026-05-23 17:23:46 UTC by art10m