

HSL lightness is not true perceived brightness ?

The **L** value in **HSL** is only a geometric quantity within that color model. It is **not** defined so that colors with the same **L** are perceived by humans as equally bright.

A typical example:

1. a strong **yellow**
2. a strong **blue**

can have the same HSL **L** value, yet yellow will still appear much brighter than blue.

Why is that?

HSL is ultimately based on the usual **RGB color space**. RGB is practical for displays, but it is not *perceptually uniform*.

That means:

1. equal numerical changes in RGB or HSL
2. do **not** produce equally large visual changes

In particular, our eyes are not equally sensitive across all hues:

1. **yellow/green** often appears very bright
2. **blue** often appears much darker
3. even when the technical values look similar

What you need instead

If you want colors to be *truly equally bright*, you need a **more perceptually uniform color space**. There are mathematical models for that.

The most important ones are:

1. **CIELAB / L*a*b***
 1. **L*** is much closer to perceived lightness

- 2. much better than HSL
- 2. **CIELUV**
 - 1. similar purpose
 - 2. slightly different focus
- 3. **OKLab / OKLCH**
 - 1. more modern
 - 2. often very well suited for UI, web, and design work
 - 3. in many cases, the best practical choice today

The core idea: “luminance” vs. “perceived lightness”

These are related, but not the same thing.

1. Physical or technical brightness: relative luminance

For an sRGB color, you can first compute the **relative luminance** Y .

If R , G , and B are **linear** RGB values in the range from 0 to 1, then:

$$Y = 0.2126 R + 0.7152 G + 0.0722 B$$

Important: these are **linear** RGB values, not the raw 8-bit sRGB values you typically use in CSS.

So first, sRGB must be linearized. For an sRGB channel c_{srgb} :

$$c_{\text{lin}} = \begin{cases} \frac{c_{\text{srgb}}}{12.92}, & c_{\text{srgb}} \leq 0.04045 \\ \left(\frac{c_{\text{srgb}} + 0.055}{1.055}\right)^{2.4}, & c_{\text{srgb}} > 0.04045 \end{cases}$$

This luminance is important for things like:

- 1. **contrast calculations**
- 2. WCAG
- 3. technical comparisons of brightness

But equal luminance is still not perfectly the same as “equally bright as perceived.”

2. Perceived brightness: perceptual lightness

For this, **CIELAB** uses the quantity L^* .

From the relative luminance Y and the reference white Y_n , it is approximately calculated as:

$$L^* = 116 \cdot \left(\frac{Y}{Y_n}\right)^{1/3} - 16$$

for larger values; more precisely, the full definition is piecewise. But the main idea is:

1. it is not linear
2. it is adjusted to human perception

Equal L^* values are much closer to “equally bright” than equal HSL L values.

Especially relevant for CSS today: OKLCH

For practical work on the web, **OKLCH** is often the best answer.

OKLCH consists of:

1. **L** = perceived lightness
2. **C** = chroma / colorfulness
3. **H** = hue angle

So it is similar to HSL in terms of intuition, but much better aligned with perception.

Advantage

If you use the same L value in OKLCH, colors are much more likely to appear equally bright.

Example in CSS:

```
color: oklch(0.7 0.15 30);  
color: oklch(0.7 0.15 120);  
color: oklch(0.7 0.15 260);
```

Here, the lightness set by `0.7` is much more consistent than in comparable HSL colors.

So: is there a mathematical way to do this?

Yes. In fact, there are several—depending on what exactly you mean by “equally bright”:

1. Equal technical luminance

1. using linear RGB and

\$\$

$$Y = 0.2126 R + 0.7152 G + 0.0722 B$$

\$\$

2. good for contrast and measurable brightness

2. Equal perceived lightness

1. using **CIELAB** `L*` or, better, **OKLab/OKLCH** `L`

2. good for design, UI, and color palettes

Practical recommendation

If your goal is:

“Colors should appear equally bright in an interface”

Use **OKLCH** instead of HSL.

“I want mathematically correct contrast checks”

Work with **relative luminance**.

“I want an older, well-established color model”

Then **CIELAB** is a good choice.

Why HSL is still often used

HSL is popular because it feels intuitive:

1. **H** = hue
2. **S** = saturation
3. **L** = light/dark

But that **L** is more of a *model parameter* than a true measure of visual lightness.

HSL can still be useful for:

1. simple color variations
2. quick prototypes
3. small adjustments

But not if you want colors that are precisely equal in perceived brightness.

Short version

Yes, there are mathematical models for this.

If colors should **really appear equally bright**, **HSL is not suitable**.

The best options are:

1. **OKLCH** for practical CSS and UI work
2. **CIELAB / OKLab** if you want to work in perceptually meaningful color spaces more rigorously

If you want, I can also show you next:

1. **why HSL lightness is defined the way it is**
2. **how to convert an HSL color into relative luminance**
3. **how to build an equal-lightness color palette in CSS using OKLCH**

Revision #1

Created 2026-05-23 17:41:56 UTC by art10m

Updated 2026-05-23 17:42:34 UTC by art10m