Color Contrast for Accessibility

Liz Marley

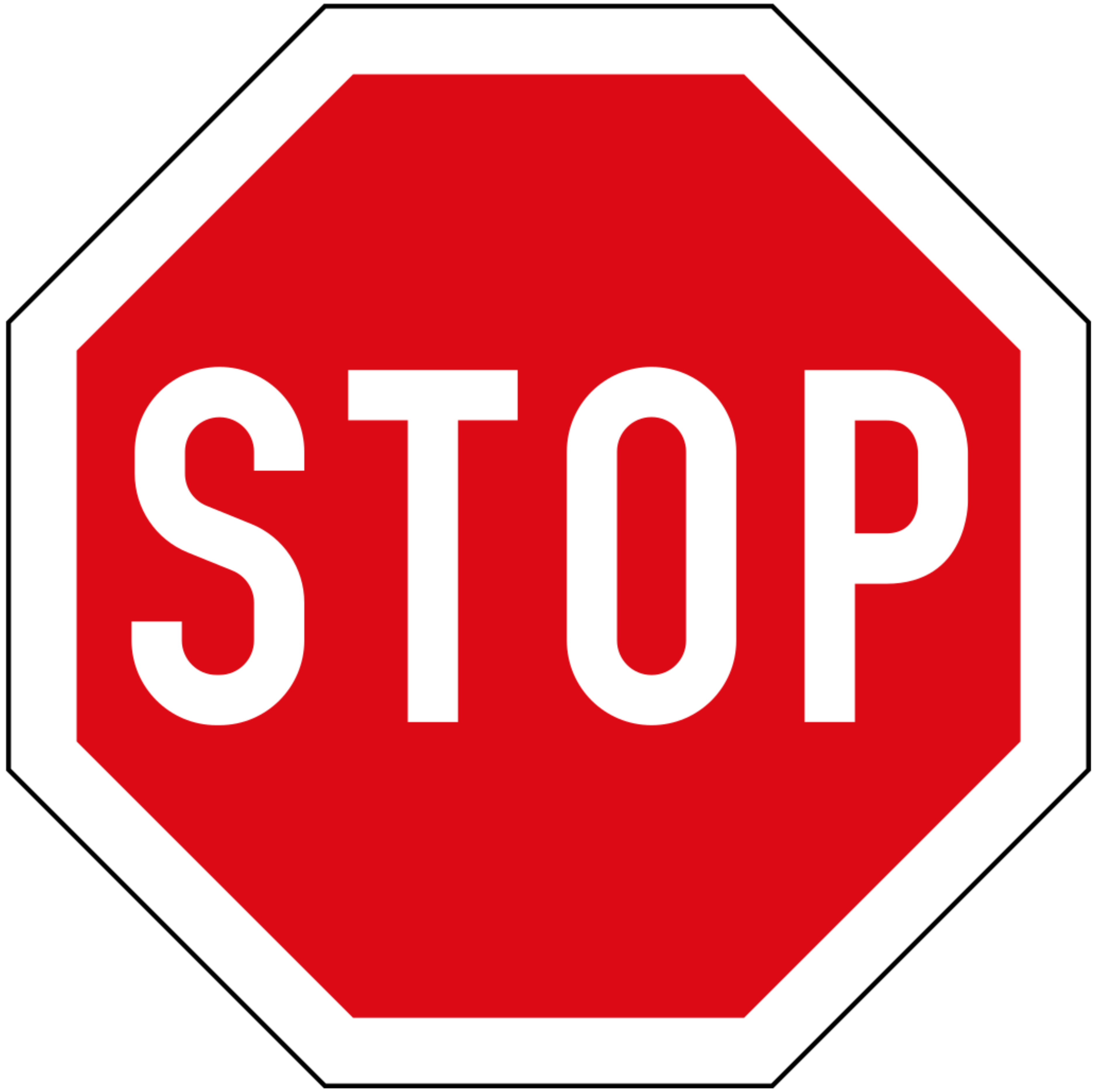
Color Contrast for Accessibility

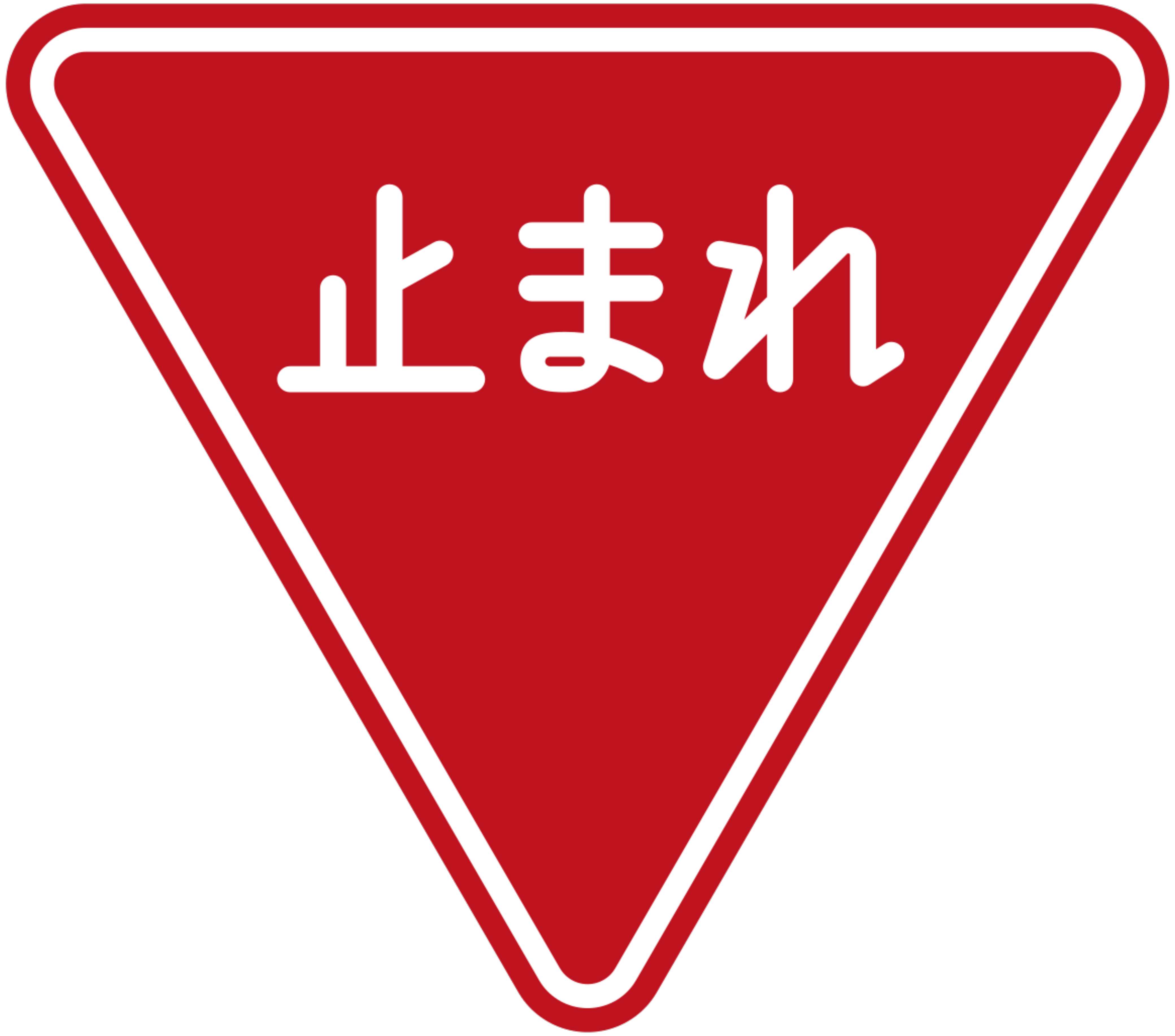
Liz Marley

Main Content

Secondary Text

Placeholder


STOP


止まれ

#FFD1D1 15.3:1	#FFC5C5 14:1	#FF6060 7:1	#FF0000 5.25:1	#EA0000 4.5:1	#B70000 3:1	#9E0000 2.4:1	#600000 1.5:1
#00FF00 15.3:1	#00F400 14:1	#00AD00 7:1	#009000 5.25:1	#008800 4.5:1	#006800 3:1	#005900 2.4:1	#003400 1.5:1
#D9D9FF 15.3:1	#CFCFFF 14:1	#8888FF 7:1	#6C6CFF 5.25:1	#5D5DFF 4.5:1	#3131FF 3:1	#0000FF 2.4:1	#00009F 1.5:1
#DCDCDC 15.3:1	#D3D3D3 14:1	#959595 7:1	#7F7F7F 5.25:1	#747474 4.5:1	#595959 3:1	#4B4B4B 2.4:1	#2C2C2C 1.5:1

The HIG (Apple's Human Interface Guidelines)

https://developer.apple.com/design/human-interface-guidelines/

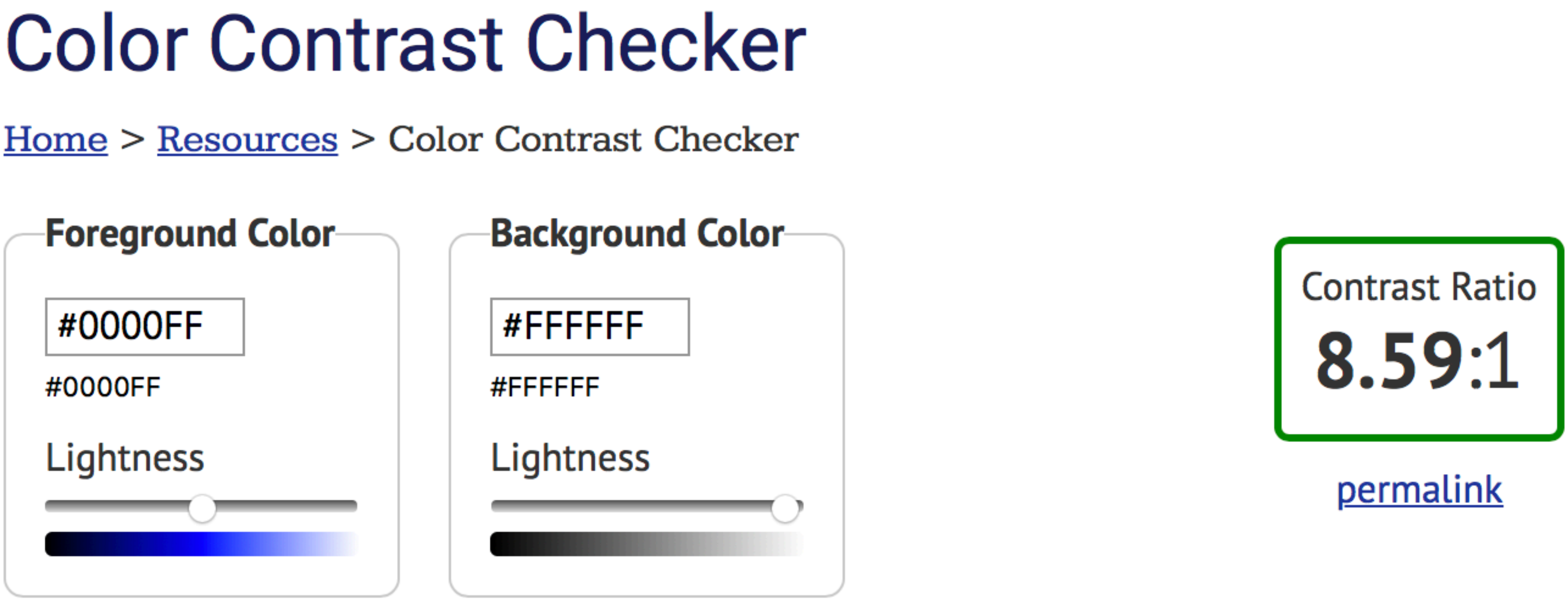
Web Content Accessibility Guidelines

https://www.w3.org/TR/WCAG20/

#FFD1D1 15.3:1	#FFC5C5 14:1	#FF6060 7:1	#FF0000 5.25:1	#EA0000 4.5:1	#B70000 3:1	#9E0000 2.4:1	#600000 1.5:1
#00FF00 15.3:1	#00F400 14:1	#00AD00 7:1	#009000 5.25:1	#008800 4.5:1	#006800 3:1	#005900 2.4:1	#003400 1.5:1
#D9D9FF 15.3:1	#CFCFFF 14:1	#8888FF 7:1	#6C6CFF 5.25:1	#5D5DFF 4.5:1	#3131FF 3:1	#0000FF 2.4:1	#00009F 1.5:1
#DCDCDC 15.3:1	#D3D3D3 14:1	#959595 7:1	#7F7F7F 5.25:1	#747474 4.5:1	#595959 3:1	#4B4B4B 2.4:1	#2C2C2C 1.5:1
		7		4.5			

#FFD1D1 1.4:1	#FFC5C5 1.5:1	#FF6060 3:1	#FF0000 4:1	#EF0000 4.5:1	#B70000 7:1	#9E0000 8.6:1	#600000 14:1
#00FF00 1.4:1	#00F400 1.5:1	#00AD00 3:1	#009400 4:1	#008A00 4.5:1	#006800 7:1	#005900 8.6:1	#003400 14:1
#D9D9FF 1.4:1	#CFCFFF 1.5:1	#8888FF 3:1	#6C6CFF 4:1	#6161FF 4.5:1	#3131FF 7:1	#0000FF 8.6:1	#00009F 14:1
#DCDCDC 1.4:1	#D3D3D3 1.5:1	#959595 3:1	#7F7F7F 4:1	#777777 4.5:1	#595959 7:1	#4B4B4B 8.6:1	#2C2C2C 14:1

4.5 7



Color Contrast Checker

Home > Resources > Color Contrast Checker

Foreground Color

#0000FF

#0000FF

Lightness

Background Color

#FFFFFF

#FFFFFF

Lightness

Contrast Ratio

8.59:1

permalink

https://webaim.org/resources/contrastchecker/

```swift
struct ColorContrastResponse: Decodable { let ratio: CGFloat }

extension UIColor {
    func hex() -> String {
        let components = cgColor.converted(to: sRGBColorspace, intent: .defaultIntent, options: nil)!.components!
        let hexStrings = components.map { String(format: "%02x", Int(255 * $0)) }
        return hexStrings[0] + hexStrings[1] + hexStrings[2]
    }
}

func contrast(_ foreground: UIColor, _ background: UIColor, completion: @escaping (CGFloat?) -> ()) {
    let url = URL(string: "https://webaim.org/resources/contrastchecker/?fcolor=\(foreground.hex())&bcolor=\
(background.hex())&api")!
    let task = URLSession.shared.dataTask(with: url) { (data, _, _) in
        guard let data = data else {
            completion(nil)
            return
        }
        let response = try? JSONDecoder().decode(ColorContrastResponse.self, from: data)
        completion(response?.ratio)
    }
    task.resume()
}
```

$$(R, G, B)component = \begin{cases} \dfrac{CC_{sRGB}}{12.92}, & if\ CC_{sRGB} \leq 0.03928 \\ \left(\dfrac{CC_{sRGB} + 0.055}{1.055}\right)^{2.4}, & otherwise \end{cases}$$

$$luminance = 0.2126 \times R + 0.7152 \times G + 0.0722 \times B$$

$$contrast = \frac{L_1 + 0.05}{L_2 + 0.05}$$

```
func relativeLuminance(_ color: CGColor) -> CGFloat {
    let sRGBColorSpace = CGColorSpaceMakeSRGB // ???
    let sRGB = color.converted(to: sRGBColorSpace,
                               intent: .defaultIntent,
                               options: nil)!.components!
    let adjusted = sRGB.map { (c) -> CGFloat in
        if c <= 0.03928 { return c / 12.92
        } else { return pow((c + 0.055)/1.055, 2.4) }
    }
    return 0.2126 * adjusted[0] +
           0.7152 * adjusted[1] +
           0.0722 * adjusted[2]
}

func contrastRatio(_ foreground: CGColor, _ background: CGColor) -> CGFloat {
    let ratio = (foreground + 0.05) / (background + 0.05)
    return ratio > 1 ? ratio : 1 / ratio
}
```

```
let sRGBColorSpace = CGColorSpaceMakeSRGB
let sRGB = color.converted(to: sRGBColorSpace,
                           intent: .defaultIntent,
                           options: nil)!.components!
```

```
let adjusted = sRGB.map { (c) -> CGFloat in
    if c <= 0.03928 {
        return c / 12.92
    }
    else {
        return pow((c + 0.055)/1.055, 2.4)
    }
}
```

```
return 0.2126 * adjusted[0] +        // red
       0.7152 * adjusted[1] +        // green
       0.0722 * adjusted[2]          // blue
```

```
func contrastRatio(c1: CGColor, c2: CGColor) -> CGFloat {
    let ratio = (c1 + 0.05) / (c2 + 0.05)
    return ratio > 1 ? ratio : 1 / ratio
}
```

#FFD1D1 15.3:1	#FFC5C5 14:1	#FF6060 7:1	#FF0000 5.25:1	#EA0000 4.5:1	#B70000 3:1	#9E0000 2.4:1	#600000 1.5:1
#00FF00 15.3:1	#00F400 14:1	#00AD00 7:1	#009000 5.25:1	#008800 4.5:1	#006800 3:1	#005900 2.4:1	#003400 1.5:1
#D9D9FF 15.3:1	#CFCFFF 14:1	#8888FF 7:1	#6C6CFF 5.25:1	#5D5DFF 4.5:1	#3131FF 3:1	#0000FF 2.4:1	#00009F 1.5:1
#DCDCDC 15.3:1	#D3D3D3 14:1	#959595 7:1	#7F7F7F 5.25:1	#747474 4.5:1	#595959 3:1	#4B4B4B 2.4:1	#2C2C2C 1.5:1
		7		4.5			

🤔

Where is Apple Leading?

I don't know but I'm going to guess anyways…

Dark Mode

NSColor.labelColor

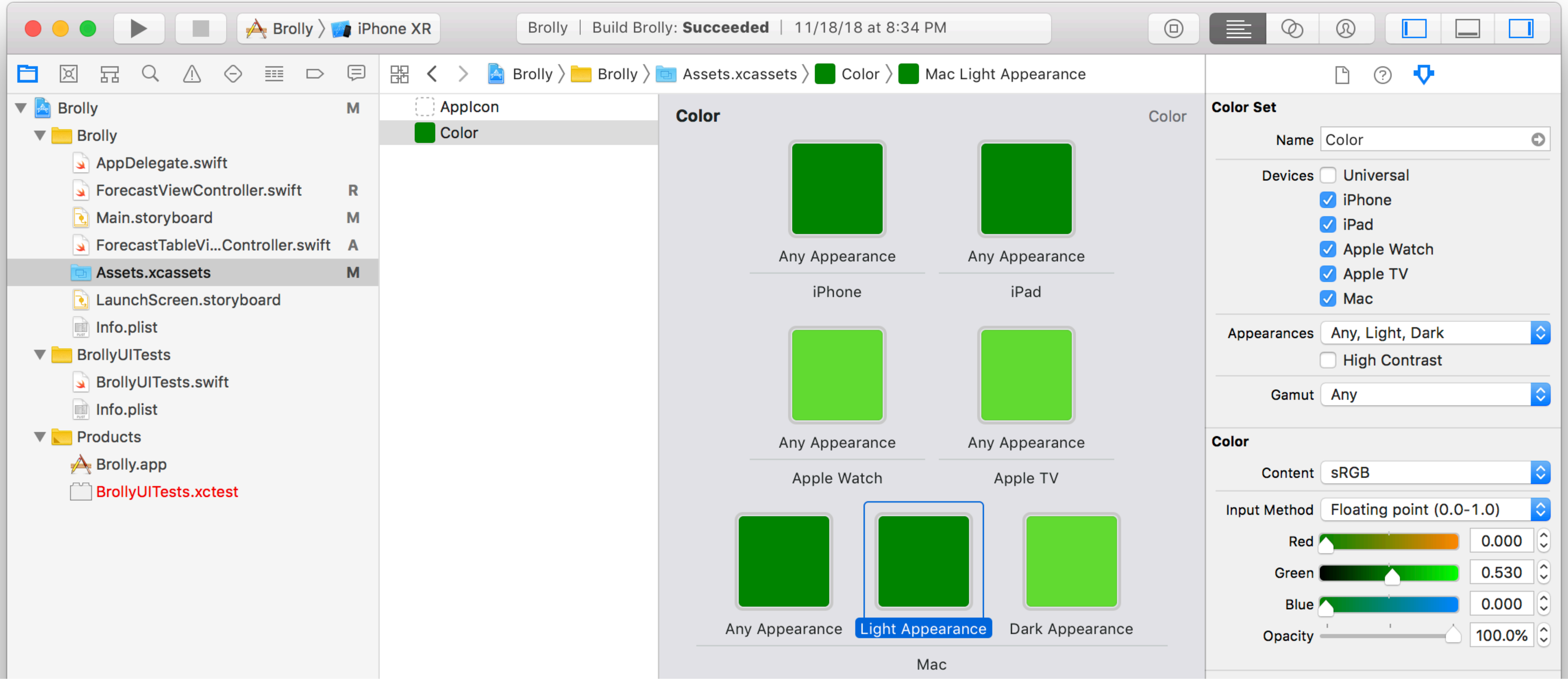
NSColor.secondaryLabelColor

NSColor.placeholderTextColor

UIColor.darkText

UIColor.lightText

UIColor.groupTableViewBackground



Color Set

Name: Color

Devices: ☐ Universal ☑ iPhone ☑ iPad ☑ Apple Watch ☑ Apple TV ☑ Mac

Appearances: Any, Light, Dark

☐ High Contrast

Gamut: Any

Color

Content: sRGB

Input Method: Floating point (0.0-1.0)

Red: 0.000

Green: 0.530

Blue: 0.000

Opacity: 100.0%

Color Assets

Use a color contrast ratio of 7 to help people access your content.

Accessibility is for Everyone

Thank you.