

DOKUMENTASI CSS

Cascading Style Sheets

Disusun oleh: Abiyya Hamdan Nurwandha

Tanggal: 29 Desember 2025

BAB I: PENDAHULUAN

1.1 Pengertian CSS

CSS (*Cascading Style Sheets*) adalah bahasa *stylesheet* yang digunakan untuk mengatur tampilan dan format dokumen HTML. CSS memisahkan konten (HTML) dari presentasi (styling), sehingga memudahkan pengelolaan dan pemeliharaan website.

1.2 Sejarah CSS

CSS pertama kali diusulkan oleh Håkon Wium Lie pada tahun 1994. CSS1 dirilis pada tahun 1996, diikuti CSS2 pada tahun 1998, dan CSS3 yang mulai dikembangkan pada tahun 1999 hingga sekarang. CSS3 dibagi menjadi modul-modul yang dikembangkan secara independen.

1.3 Fungsi CSS

CSS memiliki berbagai fungsi penting dalam pengembangan web:

- Mengatur warna, font, dan ukuran teks
- Mengatur layout dan positioning elemen
- Membuat responsive design untuk berbagai ukuran layar
- Menambahkan animasi dan transisi
- Meningkatkan user experience dengan visual yang menarik
- Mempercepat loading halaman dengan satu file CSS untuk banyak halaman

1.4 Keuntungan Menggunakan CSS

- **Konsistensi:** Satu file CSS dapat mengatur tampilan banyak halaman
 - **Maintenance:** Lebih mudah mengubah desain website
 - **Loading time:** Mengurangi ukuran file HTML
 - **Fleksibilitas:** Dapat membuat berbagai variasi tampilan
 - **Aksesibilitas:** Memisahkan struktur dari presentasi
-

BAB II: CARA PENERAPAN CSS

2.1 Inline CSS

CSS ditulis langsung pada atribut style elemen HTML:

```
<p style="color: blue; font-size: 16px;">Ini adalah paragraf dengan inline CSS</p>
```

Kelebihan: Cepat untuk styling satu elemen

Kekurangan: Sulit dimaintain, tidak efisien untuk banyak elemen

2.2 Internal CSS

CSS ditulis di dalam tag `<style>` pada bagian `<head>`:

```
<!DOCTYPE html>
<html>
<head>
<style>
  p {
    color: blue;
    font-size: 16px;
  }
  h1 {
    color: red;
    text-align: center;
  }
</style>
</head>
```

```
<body>
  <h1>Judul</h1>
  <p>Paragraf</p>
</body>
</html>
```

Kelebihan: Styling dalam satu file HTML

Kekurangan: Hanya berlaku untuk satu halaman

2.3 External CSS

CSS ditulis di file terpisah dengan ekstensi .css:

File: style.css

```
p {
  color: blue;
  font-size: 16px;
}
```

```
h1 {
  color: red;
  text-align: center;
}
```

File: index.html

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>Judul</h1>
  <p>Paragraf</p>
</body>
```

```
</html>
```

Kelebihan: Dapat digunakan untuk banyak halaman, mudah dimaintain

Kekurangan: Memerlukan HTTP request tambahan

BAB III: SELECTOR CSS

3.1 Selector Dasar

Element Selector

```
p {  
    color: blue;  
}
```

```
h1 {  
    font-size: 32px;  
}
```

Class Selector

```
.container {  
    width: 100%;  
    margin: 0 auto;  
}
```

```
.text-center {  
    text-align: center;  
}
```

ID Selector

```
#header {
```

```
background-color: navy;  
height: 80px;  
}
```

```
#footer {  
background-color: gray;  
}
```

Universal Selector

```
* {  
margin: 0;  
padding: 0;  
box-sizing: border-box;  
}
```

3.2 Selector Kombinasi

Descendant Selector (Spasi)

```
div p {  
color: red;  
}  
/* Memilih semua <p> di dalam <div> */
```

Child Selector (>)

```
div > p {  
color: blue;  
}  
/* Memilih <p> yang merupakan child langsung dari <div> */
```

Adjacent Sibling Selector (+)

```
h1 + p {
```

```
    font-weight: bold;  
}  
/* Memilih <p> yang tepat setelah <h1> */
```

General Sibling Selector (~)

```
h1 ~ p {  
    color: green;  
}  
/* Memilih semua <p> yang merupakan sibling dari <h1> */
```

3.3 Attribute Selector

```
/* Elemen dengan atribut tertentu */  
input[type="text"] {  
    border: 1px solid gray;  
}  
  
/* Elemen dengan nilai atribut yang dimulai dengan */  
a[href^="https"] {  
    color: green;  
}  
  
/* Elemen dengan nilai atribut yang diakhiri dengan */  
img[src$=".png"] {  
    border: 2px solid blue;  
}  
  
/* Elemen dengan nilai atribut yang mengandung */  
a[href*="example"] {  
    text-decoration: underline;  
}
```

3.4 Pseudo-class Selector

```
/* Hover state */
```

```
a:hover {  
    color: red;  
    text-decoration: underline;  
}
```

```
/* First child */  
p:first-child {  
    font-weight: bold;  
}
```

```
/* Last child */  
p:last-child {  
    margin-bottom: 0;  
}
```

```
/* Nth child */  
tr:nth-child(odd) {  
    background-color: #f2f2f2;  
}
```

```
tr:nth-child(even) {  
    background-color: white;  
}
```

```
/* Focus state */  
input:focus {  
    outline: 2px solid blue;  
}
```

```
/* Active state */  
button:active {  
    transform: scale(0.95);  
}
```

3.5 Pseudo-element Selector

```
/* Before */
p::before {
    content: "→";
    color: blue;
}

/* After */
p::after {
    content: " ✓";
    color: green;
}

/* First letter */
p::first-letter {
    font-size: 2em;
    font-weight: bold;
}

/* First line */
p::first-line {
    font-variant: small-caps;
}

/* Selection */
::selection {
    background-color: yellow;
    color: black;
}
```

BAB IV: PROPERTI CSS

4.1 Text Properties

```
.text-styling {
    /* Warna teks */
```

```
color: #333;

/* Font family */
font-family: 'Arial', sans-serif;

/* Ukuran font */
font-size: 16px;

/* Ketebalan font */
font-weight: bold;

/* Style font */
font-style: italic;

/* Align text */
text-align: center;

/* Dekorasi text */
text-decoration: underline;

/* Transform text */
text-transform: uppercase;

/* Jarak antar huruf */
letter-spacing: 2px;

/* Jarak antar baris */
line-height: 1.5;

/* Text shadow */
text-shadow: 2px 2px 4px rgba(0,0,0,0.3);
}
```

4.2 Background Properties

```
.background-styling {
```

```
/* Warna background */
background-color: #f0f0f0;

/* Gambar background */
background-image: url('image.jpg');

/* Repeat background */
background-repeat: no-repeat;

/* Position background */
background-position: center;

/* Size background */
background-size: cover;

/* Attachment background */
background-attachment: fixed;

/* Shorthand */
background: #f0f0f0 url('image.jpg') no-repeat center/cover;
}
```

4.3 Border Properties

```
.border-styling {
    /* Border width */
    border-width: 2px;

    /* Border style */
    border-style: solid;

    /* Border color */
    border-color: #333;

    /* Shorthand */
    border: 2px solid #333;
```

```

/* Individual borders */
border-top: 1px solid red;
border-right: 2px dashed blue;
border-bottom: 3px dotted green;
border-left: 4px double orange;

/* Border radius */
border-radius: 10px;

/* Individual corners */
border-top-left-radius: 5px;
border-top-right-radius: 10px;
border-bottom-right-radius: 15px;
border-bottom-left-radius: 20px;
}

```

4.4 Spacing Properties

```

.spacing {
    /* Margin (luar) */
    margin: 20px;
    margin-top: 10px;
    margin-right: 15px;
    margin-bottom: 20px;
    margin-left: 25px;

    /* Margin shorthand */
    margin: 10px 20px 30px 40px; /* top right bottom left */
    margin: 10px 20px; /* top-bottom left-right */
    margin: 10px; /* semua sisi */

    /* Padding (dalam) */
    padding: 15px;
    padding-top: 10px;
    padding-right: 15px;
}

```

```
padding-bottom: 20px;  
padding-left: 25px;  
  
/* Padding shorthand */  
padding: 10px 20px 30px 40px; /* top right bottom left */  
padding: 10px 20px; /* top-bottom left-right */  
padding: 10px; /* semua sisi */  
}
```

4.5 Dimension Properties

```
.dimensions {  
    /* Width */  
    width: 300px;  
    width: 50%;  
    width: 100vw; /* viewport width */  
  
    /* Height */  
    height: 200px;  
    height: 50vh; /* viewport height */  
  
    /* Min/Max width */  
    min-width: 200px;  
    max-width: 1200px;  
  
    /* Min/Max height */  
    min-height: 100px;  
    max-height: 500px;  
}
```

BAB V: BOX MODEL

5.1 Konsep Box Model

Setiap elemen HTML diperlakukan sebagai box yang terdiri dari:

1. **Content**: Konten aktual (teks, gambar, dll)
2. **Padding**: Ruang antara content dan border
3. **Border**: Garis pembatas
4. **Margin**: Ruang di luar border

```
.box {  
    width: 300px;  
    height: 200px;  
    padding: 20px;  
    border: 5px solid black;  
    margin: 10px;  
}  
  
/* Total width = width + padding-left + padding-right + border-left + border-right */  
/* Total width = 300 + 20 + 20 + 5 + 5 = 350px (tidak termasuk margin) */
```

5.2 Box Sizing

```
/* Default (content-box) */  
.default-box {  
    box-sizing: content-box;  
    width: 300px;  
    padding: 20px;  
    border: 5px solid black;  
    /* Total width = 350px */  
}  
  
/* Border-box (lebih praktis) */  
.border-box {  
    box-sizing: border-box;  
    width: 300px;  
    padding: 20px;  
    border: 5px solid black;  
    /* Total width tetap 300px */  
}
```

```
/* Best practice untuk semua elemen */  
* {  
    box-sizing: border-box;  
}
```

BAB VI: DISPLAY DAN POSITIONING

6.1 Display Property

```
/* Block - elemen mengambil lebar penuh */  
.block {  
    display: block;  
}
```

```
/* Inline - elemen sebaris dengan teks */  
.inline {  
    display: inline;  
}
```

```
/* Inline-block - kombinasi inline dan block */  
.inline-block {  
    display: inline-block;  
    width: 200px;  
    height: 100px;  
}
```

```
/* None - menyembunyikan elemen */  
.hidden {  
    display: none;  
}
```

```
/* Flex - flexbox layout */  
.flex-container {  
    display: flex;
```

```
}
```

```
/* Grid - grid layout */
```

```
.grid-container {
```

```
    display: grid;
```

```
}
```

6.2 Position Property

```
/* Static (default) */
```

```
.static {
```

```
    position: static;
```

```
}
```

```
/* Relative - relatif terhadap posisi normalnya */
```

```
.relative {
```

```
    position: relative;
```

```
    top: 10px;
```

```
    left: 20px;
```

```
}
```

```
/* Absolute - relatif terhadap parent dengan position selain static */
```

```
.absolute {
```

```
    position: absolute;
```

```
    top: 0;
```

```
    right: 0;
```

```
}
```

```
/* Fixed - relatif terhadap viewport */
```

```
.fixed {
```

```
    position: fixed;
```

```
    bottom: 20px;
```

```
    right: 20px;
```

```
}
```

```
/* Sticky - kombinasi relative dan fixed */
```

```
.sticky {  
    position: sticky;  
    top: 0;  
}
```

6.3 Z-Index

```
/* Mengatur urutan tumpukan elemen */
```

```
.layer-1 {  
    position: relative;  
    z-index: 1;  
}
```

```
.layer-2 {  
    position: relative;  
    z-index: 2; /* Di atas layer-1 */  
}
```

```
.layer-3 {  
    position: relative;  
    z-index: 3; /* Di atas layer-2 */  
}
```

6.4 Float dan Clear

```
/* Float */
```

```
.float-left {  
    float: left;  
    margin-right: 10px;  
}
```

```
.float-right {  
    float: right;  
    margin-left: 10px;  
}
```

```
/* Clear float */
.clear {
    clear: both;
}

/* Clearfix hack */
.clearfix::after {
    content: "";
    display: table;
    clear: both;
}
```

BAB VII: FLEXBOX

7.1 Flex Container Properties

```
.flex-container {
    display: flex;

    /* Direction */
    flex-direction: row; /* row | row-reverse | column | column-reverse */

    /* Wrap */
    flex-wrap: wrap; /* nowrap | wrap | wrap-reverse */

    /* Justify content (horizontal alignment) */
    justify-content: center; /* flex-start | flex-end | center | space-between |
    space-around | space-evenly */

    /* Align items (vertical alignment) */
    align-items: center; /* flex-start | flex-end | center | stretch | baseline */

    /* Align content (multiple lines) */
    align-content: center;
```

```
/* Gap */  
gap: 20px;  
}
```

7.2 Flex Item Properties

```
.flex-item {  
    /* Flex grow */  
    flex-grow: 1;  
  
    /* Flex shrink */  
    flex-shrink: 1;  
  
    /* Flex basis */  
    flex-basis: 200px;  
  
    /* Shorthand */  
    flex: 1 1 200px; /* grow shrink basis */  
  
    /* Align self */  
    align-self: center;  
  
    /* Order */  
    order: 2;  
}
```

7.3 Contoh Praktis Flexbox

```
/* Navbar dengan flexbox */  
.navbar {  
    display: flex;  
    justify-content: space-between;  
    align-items: center;  
    padding: 20px;  
    background-color: #333;  
}
```

```
.navbar-links {  
    display: flex;  
    gap: 20px;  
}  
  
/* Card layout */  
.card-container {  
    display: flex;  
    flex-wrap: wrap;  
    gap: 20px;  
    justify-content: center;  
}  
  
.card {  
    flex: 1 1 300px;  
    max-width: 350px;  
    padding: 20px;  
    border: 1px solid #ddd;  
}  
  
/* Centering dengan flexbox */  
.center-everything {  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    min-height: 100vh;  
}
```

BAB VIII: GRID LAYOUT

8.1 Grid Container Properties

```
.grid-container {  
    display: grid;
```

```

/* Columns */
grid-template-columns: 200px 200px 200px;
grid-template-columns: 1fr 2fr 1fr; /* fractional units */
grid-template-columns: repeat(3, 1fr); /* repeat */
grid-template-columns: repeat(auto-fit, minmax(250px, 1fr)); /* responsive */

/* Rows */
grid-template-rows: 100px 200px auto;

/* Gap */
gap: 20px;
row-gap: 10px;
column-gap: 20px;

/* Justify items (horizontal) */
justify-items: center; /* start | end | center | stretch */

/* Align items (vertical) */
align-items: center;

/* Justify content */
justify-content: center; /* start | end | center | space-between | space-around |
space-evenly */

/* Align content */
align-content: center;
}

```

8.2 Grid Item Properties

```

.grid-item {
    /* Grid column */
    grid-column: 1 / 3; /* dari kolom 1 sampai sebelum kolom 3 */
    grid-column-start: 1;
    grid-column-end: 3;
}

```

```
grid-column: span 2; /* spanning 2 kolom */  
  
/* Grid row */  
grid-row: 1 / 3;  
grid-row-start: 1;  
grid-row-end: 3;  
grid-row: span 2;  
  
/* Justify self */  
justify-self: center;  
  
/* Align self */  
align-self: center;  
}
```

8.3 Grid Template Areas

```
.grid-layout {  
    display: grid;  
    grid-template-columns: 1fr 3fr 1fr;  
    grid-template-rows: auto 1fr auto;  
    grid-template-areas:  
        "header header header"  
        "sidebar main aside"  
        "footer footer footer";  
    gap: 20px;  
}
```

```
.header { grid-area: header; }  
.sidebar { grid-area: sidebar; }  
.main { grid-area: main; }  
.aside { grid-area: aside; }  
.footer { grid-area: footer; }
```

BAB IX: RESPONSIVE DESIGN

9.1 Media Queries

```
/* Mobile first approach */
.container {
    width: 100%;
    padding: 10px;
}

/* Tablet */
@media (min-width: 768px) {
    .container {
        width: 750px;
        margin: 0 auto;
    }
}

/* Desktop */
@media (min-width: 1024px) {
    .container {
        width: 1000px;
    }
}

/* Large desktop */
@media (min-width: 1440px) {
    .container {
        width: 1200px;
    }
}

/* Orientation */
@media (orientation: landscape) {
    .container {
        flex-direction: row;
    }
}
```

```
}
```



```
@media (orientation: portrait) {  
    .container {  
        flex-direction: column;  
    }  
}
```

9.2 Responsive Units

```
.responsive-units {  
    /* Viewport units */  
    width: 100vw; /* 100% viewport width */  
    height: 100vh; /* 100% viewport height */  
    font-size: 5vmin; /* 5% of smaller viewport dimension */  
  
    /* Relative units */  
    font-size: 1em; /* relatif terhadap parent */  
    font-size: 1.5rem; /* relatif terhadap root element */  
  
    /* Percentage */  
    width: 50%;  
    padding: 5%;  
  
    /* Calc function */  
    width: calc(100% - 40px);  
    height: calc(100vh - 80px);  
}
```

9.3 Responsive Images

```
.responsive-image {  
    max-width: 100%;  
    height: auto;  
    display: block;  
}
```

```
/* Object fit */
.cover-image {
    width: 100%;
    height: 300px;
    object-fit: cover;
}

.contain-image {
    width: 100%;
    height: 300px;
    object-fit: contain;
}
```

BAB X: TRANSFORMS DAN ANIMATIONS

10.1 Transform Properties

```
.transform-element {
    /* Translate (pindah posisi) */
    transform: translate(50px, 100px);
    transform: translateX(50px);
    transform: translateY(100px);

    /* Rotate (putar) */
    transform: rotate(45deg);

    /* Scale (ukuran) */
    transform: scale(1.5);
    transform: scaleX(2);
    transform: scaleY(0.5);

    /* Skew (miring) */
    transform: skew(10deg, 20deg);}
```

```
/* Multiple transforms */
transform: translate(50px, 50px) rotate(45deg) scale(1.5);

/* Transform origin */
transform-origin: center;
transform-origin: top left;
}
```

10.2 Transitions

```
.transition-element {
    background-color: blue;
    transition: background-color 0.3s ease;
}

.transition-element:hover {
    background-color: red;
}

/* Multiple properties */
.multi-transition {
    transition: all 0.3s ease;
    /* atau */
    transition:
        background-color 0.3s ease,
        transform 0.5s cubic-bezier(0.68, -0.55, 0.265, 1.55);
}

/* Timing functions */
.timing {
    transition-timing-function: ease; /* ease | linear | ease-in | ease-out | ease-in-out |
cubic-bezier() */
}
```

10.3 Animations

```
/* Define keyframes */
@keyframes slide-in {
    0% {
        transform: translateX(-100%);
        opacity: 0;
    }
    100% {
        transform: translateX(0);
        opacity: 1;
    }
}

/* Apply animation */
:animated-element {
    animation: slide-in 1s ease-out;
}

/* Animation properties */
.full-animation {
    animation-name: slide-in;
    animation-duration: 1s;
    animation-timing-function: ease-out;
    animation-delay: 0.5s;
    animation-iteration-count: infinite;
    animation-direction: alternate;
    animation-fill-mode: forwards;
    animation-play-state: running;

    /* Shorthand */
    animation: slide-in 1s ease-out 0.5s infinite alternate forwards;
}

/* Complex animation example */
@keyframes bounce {
    0%, 20%, 50%, 80%, 100% {
        transform: translateY(0);
    }
}
```

```
        }
        40% {
            transform: translateY(-30px);
        }
        60% {
            transform: translateY(-15px);
        }
    }

.bouncing {
    animation: bounce 2s infinite;
}
```

BAB XI: ADVANCED CSS

11.1 CSS Variables (Custom Properties)

```
:root {
    /* Define variables */
    --primary-color: #3498db;
    --secondary-color: #2ecc71;
    --font-size-base: 16px;
    --spacing-unit: 8px;
}

.element {
    /* Use variables */
    color: var(--primary-color);
    font-size: var(--font-size-base);
    padding: calc(var(--spacing-unit) * 2);
}

/* Override in specific context */
.dark-theme {
    --primary-color: #1a5490;
```

```
--secondary-color: #1e8449;  
}
```

11.2 Filters

```
.filtered-image {  
    /* Blur */  
    filter: blur(5px);  
  
    /* Brightness */  
    filter: brightness(150%);  
  
    /* Contrast */  
    filter: contrast(200%);  
  
    /* Grayscale */  
    filter: grayscale(100%);  
  
    /* Hue rotate */  
    filter: hue-rotate(90deg);  
  
    /* Invert */  
    filter: invert(100%);  
  
    /* Opacity */  
    filter: opacity(50%);  
  
    /* Saturate */  
    filter: saturate(200%);  
  
    /* Sepia */  
    filter: sepia(100%);  
  
    /* Drop shadow */  
    filter: drop-shadow(5px 5px 10px rgba(0,0,0,0.5));
```

```
/* Multiple filters */  
filter: brightness(110%) contrast(120%) saturate(130%);  
}
```

11.3 Gradients

```
/* Linear gradient */  
.linear-gradient {  
    background: linear-gradient(to right, red, yellow);  
    background: linear-gradient(45deg, red, yellow, green);  
    background: linear-gradient(to bottom, rgba(0,0,0,0), rgba(0,0,0,1));  
}
```

```
/* Radial gradient */  
.radial-gradient {  
    background: radial-gradient(circle, red, yellow, green);  
    background: radial-gradient(ellipse at top left, red, yellow);  
}
```

```
/* Conic gradient */  
.conic-gradient {  
    background: conic-gradient(red, yellow, green, blue, red);  
}
```

```
/* Multiple gradients */  
.multiple-gradients {  
    background:  
        linear-gradient(45deg, transparent 30%, rgba(255,255,255,0.3) 30%,  
        rgba(255,255,255,0.3) 70%, transparent 70%),  
        linear-gradient(-45deg, transparent 30%, rgba(255,255,255,0.3) 30%,  
        rgba(255,255,255,0.3) 70%, transparent 70%);  
}
```

11.4 Shadows

```
.shadows {
```

```

/* Box shadow */
box-shadow: 5px 5px 10px rgba(0,0,0,0.3);
box-shadow: 0 2px 4px rgba(0,0,0,0.1), 0 8px 16px rgba(0,0,0,0.1); /* Multiple shadows */
*/
/* Inner shadow */
box-shadow: inset 0 0 10px rgba(0,0,0,0.5);

/* Text shadow */
text-shadow: 2px 2px 4px rgba(0,0,0,0.5);
text-shadow: 0 0 10px #fff, 0 0 20px #fff; /* Glow effect */
}

```

11.5 Clip Path

```

.clip-path {
    /* Circle */
    clip-path: circle(50%);

    /* Ellipse */
    clip-path: ellipse(50% 30%);

    /* Polygon */
    clip-path: polygon(50% 0%, 100% 50%, 50% 100%, 0% 50%);

    /* Inset */
    clip-path: inset(10px 20px 30px 40px);
}

```

BAB XII: BEST PRACTICES

12.1 Naming Convention

BEM (Block Element Modifier)

```
/* Block */
.card { }

/* Element */
.card__title { }
.card__content { }
.card__button { }

/* Modifier */
.card--featured { }
.card__button--primary { }
```

12.2 CSS Organization

```
/* 1. Reset/Normalize */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

/* 2. Variables */
:root {
  --primary-color: #3498db;
}

/* 3. Base styles */
body {
  font-family: Arial, sans-serif;
  line-height: 1.6;
}

/* 4. Layout */
.container { }
.grid { }
```

```
/* 5. Components */
.button { }
.card { }

/* 6. Utilities */
.text-center { }
.mt-20 { }

/* 7. Media queries */
@media (min-width: 768px) { }
```

12.3 Performance Tips

```
/* Hindari selector yang terlalu spesifik */
/* ✗ Bad */
div.container ul li a.link { }

/* ✓ Good */
.nav-link { }

/* Gunakan transform untuk animasi */
/* ✗ Bad - triggers layout */
.element:hover {
    margin-left: 10px;
}

/* ✓ Good - only triggers composite */
.element:hover {
    transform: translateX(10px);
}

/* Minimize repaints dan reflows */
/* ✓ Gunakan will-change untuk animasi kompleks */
.animated-element {
    will-change: transform;
}
```

12.4 Accessibility

```
/* Focus visible untuk keyboard navigation */
button:focus-visible {
    outline: 2px solid blue;
    outline-offset: 2px;
}

/* Sufficient color contrast */
.text {
    color: #333;
    background-color: #fff; /* Minimum 4.5:1 ratio */
}

/* Hide visually but keep for screen readers */
.
```