Engineer

CSS

ページの見た目を指定する言語

どのようなものか

CSSはCascading Style Sheetsの略で、CSS3という呼び方も今なおよく使われます。ブラウザーが文書へ適用し、各要素の描き方を決める規則の言語です。色、大きさ、余白、位置、書体、動きを定め、文書の表示先がスマートフォンでも、大きなモニターでも、紙でも働きます。

スタイルシートは、規則の一覧です。各規則は、要素を選ぶセレクターと、それぞれがプロパティーと値からなる宣言を組み合わせます。

```
.note {
    color: #3a3a3a;
    margin-block: 1.5rem;
}
```

名前の最初の単語が、日々の作業を左右します。同じ要素の同じプロパティーを、複数の規則が指定するのは日常的で、どれを優先するかは、カスケードが決めます。主な判断順序は、出所と重要度、つまりブラウザーの既定値、読者自身の設定、サイトのスタイルシート、宣言に `!important` が付いているかです。その後に、カスケードレイヤー、詳細度、最後に出現順が続きます。同じ重みなら、後の規則が前の規則に優先します。詳細度は、セレクターごとに三つの数値で表します。まずIDの数、次にクラス・属性・疑似クラスの数、最後に要素名の数です。全称セレクターの `*` は、どの数にも加算しません。継承は、カスケードと並んで働きます。色やフォントのような値は子要素へ伝わりますが、マージンや境界線などは、設定した要素にとどまります。

CSSには、一つのバージョン番号はありません。CSS 2.1以降、作業部会はモジュール方式に移りました。Level 3は、モジュールごとに2.1を拡張します。それ以降は、モジュールごとに独立したレベルを持つため、SelectorsはLevel 4の草案

である一方、以前に相当するものがないモジュールは、Level 1から始まります。「CSS3」は、このモジュール方式の時代を指す通称です。現在のCSSを定める年次スナップショットも、CSS Level 4は存在しないと明言しています。

実務上、最も大きな意味を持つのは、エラーの扱いです。ブラウザーは、知らないプロパティーや、そのプロパティーでは無効な値に出会うと、その宣言だけを無視し、規則の残りを適用しなければなりません。新しいプロパティーや値を加えても、古いブラウザーを壊さないための規則です。一部のブラウザーだけが対応する機能を、サイトで使えるのも、そのためです。代償は、沈黙です。ページは、少しだけ間違った状態で描画されます。残るのは、誰かが探しに行かなければ見つからない、開発者ツール内の取り消し線付きの一行だけです。

CSSと比較されるものの大半は、結局、CSSの書き方です。SassとLessは、CSSへコンパイルするプリプロセッサーです。Tailwindなどのユーティリティーフレームワークは、マークアップのクラス名からCSSを生成します。CSS-in-JSライブラリーは、JavaScriptコンポーネントからCSSを出力し、`style` 属性は、一つの要素へ宣言を直接置きます。どれも、CSSの作成方法を変えますが、最後には、同じカスケードで順序を決める宣言になります。

概念を判断に生かす

「CSSとは？」を実際の判断に生かしましょう。その概念を、開発または購入するシステムに結び付けて考えるお手伝いをします。 読み込みが速く、スマートフォンでも使え、お客様がどのブラウザーを使っていても同じように動く画面を提供します。

サービスを見る お問い合わせ →

用語の背後にある前提を検証する

解説とリンク先の実績を参考に、プロジェクトについての質問を整理してください。ツールを選ぶ前に、前提を一緒に検討できます。 フルスタックのGIS地図開発を指揮 — DBからブラウザーまでのフルスタック地図開発

フルスタックのGIS地図開発を指揮 お問い合わせ →

なぜ重要なのか

CSSを使うチームは、規則が効かなくても通知されない状態を、どの段階まで許すか決める必要があります。言語は、適用できないものを捨て、より強い規則を、何も言わずに優先します。そのため、ある程度使ってきたスタイルシートには、何もしない宣言が必ずあります。チームが決められるのは、誰かがそれに気付くか、いつ気付くかです。

アクセシビリティーは、多くの人が思う以上にCSSに依存します。コントラスト、見えるフォーカスリング、動きを減らす設定への対応、使える強制配色モード、拡大しても読める文字は、どれも少数の宣言で成り立ちます。そして、別の

場所にある、たまたま詳細度の高い規則によって、どれも無効になり得ます。その機能を必要としなかった人には、ページは問題なく見えます。

保守コストも、CSSの管理方法に左右されます。誰にも振る舞いを予測できないスタイルシートは、増える一方です。規則を消すと、誰も検査していないページが壊れるかもしれないためです。行き着くのは、多くの場合、以前の上書きをさらに上書きする指定が大半を占め、変更のたびに、画面がよく見えるまで値を試すファイルです。

CSSのこの性質が最も直接的に影響するのは、検証です。受け入れられるものだけを受け入れ、残りを無視するブラウザーを、そのまま合否の検査にはできません。ソースを読むリンターは、誤記や無効な値を、配信前に見つけます。ただし、その後に圧縮処理や、ほかのビルド工程がファイルをどう変えるかは見えません。また、二つの有効な規則のどちらが、特定のページで優先されるかも分かりません。配信するファイルへの検査と、描画したページの見た目の比較で、リンターに見えない部分を確かめます。

早い段階から、詳細度をそろえ、名前付きのレイヤーを使い、カスタムプロパティーにデザイントークンを保持してカスケードを整理すれば、変更ごとに必要なのは、少しの規律です。その規律なしに育ったスタイルシートを整理し直すには、通常、書き直しが必要になります。

実際の活用例

このサイトのクレジットページは、ビルド時に連結し、内容のハッシュ値で識別する、一つのCSSバンドルを説明しています。手描き風の矢印や余白の注記も、CSSだけで描き、品質ゲートのStylelintが、ソースに規則を適用します。同じページは、W3C CSS Validatorへリンクし、何を報告するかも、あらかじめ伝えています。2026年8月時点では、その検証プロファイルは、`system-ui`、`@property`、`font-size`内の`clamp()`、スクロールに連動するアニメーションに対応し切れていませんでした。そのため、/terms/に対して176件を報告し、175件は、ブラウザーが既に提供し、サイトが使っている機能についてのものでした。残る一件は、本当の問題でした。静止時の`rotate: 0deg`は、圧縮処理で単位を失いました。`rotate: 0`は角度ではないため、ブラウザーが宣言全体を捨て、指定した回転のないアニメーションが配信されました。値の仕様が、角度に単位なしのゼロを認めるのは、少数の従来の箇所だけです。Stylelintは、ソースを読み、圧縮処理はその後で書き換えるため、問題を見ませんでした。現在は、当社独自の検査が、実際に配信したものを読んでいます。

アクセシビリティーを改善したある作業では、問題のすべてが詳細度にありました。サイトは、動きを減らす設定へ、一つの全称セレクターの規則で、すべてのトランジションを止めることで対応していました。それでも、その設定をエミュレートすると、11個のセレクターで、まだ動きがありました。全称セレクターの点数はゼロなので、クラスを持つどの規則にも負けます。対策は、セレクターを強くすることではなく、構造でした。現在、動くものは、スタイルシートの末尾近くにある一つのパートへまとめ、ほかの場所でtransformをトランジションさせると、リンターが失敗します。色のフェードは動きではないため、許可します。動きは、キーフレームアニメーションではなく、トランジションとして書く必要があります。トランジションだけが、きれいに取り消せるためです。記録した教訓は短く、詳細度で負ける規則は、気付かれずに失敗するというものです。

スタイルシートで最初にずれるのは、色です。このサイトのパレットは、28色で確定しました。使う面に対するコントラスト比を、各色について記録し、11のブランド値を、トークンとして一度だけ宣言しました。知覚的な距離が0.02未満の二色は、同じ色の二つの表記と数えます。リンターも、初回の実行ですぐに、ほぼ重複する二つの表記を見つけました。透過は、混色関数ではなく、相対色構文で書きます。その理由は、ページ全体を要約するものです。検査は、混色を通して元の色を見抜けないのです。

問題が起きるところ

宣言が消えても、ページは描画されます

一文字の誤りで宣言が無効になり、ブラウザーは、止まらずに読み飛ばします。

```
.card {
    padding: 16; /* 長さには単位が必要：無視される */
    colour: #222; /* プロパティー名ではない：無視される */
    margin-block: 1rem; /* 適用される */
}
```

カードは、マージンがある一方、パディングも色もない状態で描画され、ブラウザーは何も警告しません。ビルド時にソースへリンターをかけ、エラーなら失敗させます。リンターの後に、圧縮処理、プリプロセッサー、バンドル処理がCSSを書き換える場合は、出力も検査します。ブラウザーが読むのは、そのファイルだからです。

より強い規則に負けます

コンポーネントの規則が、別の場所から来た、より詳細度の高いセレクターや、同じ詳細度で後から読み込む規則に負けます。よくある反応は、セレクターを長くするか、`!important`を使うことです。今日は勝てても、次の変更のコストを上げます。IDよりクラスを使い、入れ子を少なくして、詳細度をそろえます。リセット、基本、コンポーネント、ユーティリティーといったグループ全体は、カスケードレイヤーで、詳細度では覆せない明示的な順序に並べます。`!important`の宣言では、レイヤーの順序が逆になり、最も早いレイヤーが優先します。使う前に、知っておくべき点です。

スタイルシートが、増える一方です

古い規則を使っていないと証明するより、新しく書く方が速いため、作業のたびに規則を追加します。既にトークンで定義する値も、その隣へリテラルとして書き直し、似た色が増え、どれを意図したか分からなくなります。共有する値はカスタムプロパティーに保持し、ページ群が実際に使う規則を測り、削除も作業に含めます。ビルドにはサイズの上限を設け、増加を見えるようにします。

一つの大きさでしか、レイアウトが保てません

短い英語の文章を使い、一つの幅だけで検査したデザインは、長い翻訳、拡大した文字、幅の狭いスマートフォンで崩れます。固定した高さは内容を切り取り、固定した幅は横スクロールを強います。アクセシビリティー指針は、200%へ拡大できる文字と、320 CSSピクセル幅で再配置されるコンテンツを求めます。ボックスの大きさは内容から決め、`min()`、`max()`、`clamp()`で流動的な値を使います。完成とする前に、長い文字列とズームで検査します。

参考資料

- CSS Snapshot 2026 — 現在のCSSを構成するモジュールと、バージョンではなくレベルを持つ理由を説明します。
- CSSのカスケードと継承、Level 5 — 完全な並べ替えの順序、カスケードレイヤー、継承を、順に説明します。
- Selectors Level 4 — すべてのセレクターと、その詳細度の計算方法を説明します。
- 構文と基本データ型 — 不明なプロパティーと不正な値を無視する、CSS 2.1の規則を、例とともに示します。
- CSS Values and Units Module Level 4 — 長さ、角度、`clamp()`、単位なしのゼロが今も許可される箇所を説明します。

用語集

実際の制約に合わせて選ぶ

業務の流れ、既存のインターフェース、変更を必要とする利用者をお知らせください。検証可能な納品内容と、お客様のチームが保守を続けられる引き継ぎ内容を定められます。

料金を見る お問い合わせ →

用語を実務に結び付ける

「CSSとは？」がお客様の組織にどう当てはまるかをご相談ください。現在のシステムの具体例が一つあれば、話を始められます。

サービスを見る

welcome@engineer.company +45 50 10 61 84 デンマーク、コペンハーゲン

このページを持ち歩く

連絡先カード

心を込めて丁寧に制作： D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

著作権 © 2025 – 現在 · Engineer ApS · すべての権利を留保します。

印刷元： https://engineer.company/ja/glossary/css/