Engineer

スタイルシート

🖌 ページのCSSをまとめたファイル

どのようなものか

スタイルシートは、仕様ではstyle sheetと二語で書かれ、単にCSSファイルとも呼ばれます。ブラウザーが一つの単位として取得、解析し、文書へ適用するCSSルールの集まりです。

ページがスタイルシートを受け取る方法は四つあります。`link`要素に`rel="stylesheet"`を指定すると、別ファイルを指します。`style`要素はHTML自体にルールを置きます。シートの先頭にある`@import`ルールは、別のシートを読み込みます。スクリプトで`new CSSStyleSheet()`を使って作り、`adoptedStyleSheets`を通して文書やシャドウルートに関連付ける方法もあります。最初の二つをページのheadに書くと、次のようになります。

```
<link rel="stylesheet" href="/css/site.css" />
<link rel="stylesheet" href="/css/print.css" media="print" />
<style>
    .banner {
        border-block-end: 2px solid;
    }
</style>
```

作者のシートだけが使われるわけではありません。各ブラウザーは既定のスタイルシートを備えているため、独自のスタイルを付けない見出しも大きな太字で表示されます。読者も自分のユーザースタイルシートを追加でき、カスケードが三つの由来を比較します。一つの要素の`style`属性は別物です。セレクターを持たない宣言であり、シートではなく、その要素だけに属します。

読み込まれたスタイルシートは、CSS Object Model内の`CSSStyleSheet`オブジェクトになります。この用語の定義は、その仕様に由来します。オブジェクトには、シートの所有要素、適用するメディア、タイトル、無効かどうか、ルールの

一覧が記録され、`document.styleSheets` はページのシートを順番に列挙します。プライバシーのためのフラグもあります。CORSなしで別オリジンから取得したシートはorigin-cleanではなく、スクリプトがルールを取得しようとすると、代わりにセキュリティエラーになります。

`media` 属性はシートを適用する環境を限定します。`media="print"` のシートもダウンロードされますが、ブラウザーが待つのはメディア条件が一致するシートだけなので、印刷用シートが画面表示を止めることはありません。

概念を判断に生かす

「スタイルシートとは？」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

スタイルシートは、ブラウザーが何かを表示する前に何を取得する必要があるかを決めます。HTML標準では、body要素が存在する前にパーサーが見つけた有効なスタイルシートへのリンクは、メディア条件が一致すると描画をブロックします。通常、最初の描画はファイルの到着と解析を待ちます。スタイルのないページを描いてから、読者の目の前で描き直すのを避けるためです。サイズ、配信ホスト、発見に必要な通信の往復回数が、クリックから最初の表示までの間に入ります。

最初の判断は、ファイルの数です。全ページで共通のバンドルなら一度取得し、その後はキャッシュから使えます。ページごとのシートなら、最初の転送量は少なく、読み進めるにつれて増えます。`style` 要素内に直接書いたルールは追加リクエストを生みませんが、文書と別にキャッシュできないため、ページごとに再び転送されます。各 `@import` は発見の段階を一つ増やします。ブラウザーは、読み込み元のシートを読む途中で初めて取り込み先のアドレスを知るため、キャッシュにないインポートの取得では、ネットワークの往復がもう一回増えることがあります。

二つ目はキャッシュです。一年キャッシュするシートは、その後のページを速くしますが、変更時には古いコピーを持つ読者にも更新を届ける必要があります。一般的な答えは、内容のハッシュをファイル名に含めるフィンガープリントで

す。ファイルが変われば新しいアドレスになり、古いものは好きなだけキャッシュしたままにできます。一方、それを参照するHTMLのキャッシュ期間は短くします。

三つ目は信頼です。シートは何をどこに表示するかを決め、警告を隠し、ボタンの見た目を変え、ルール内で指定したどのアドレスからでも画像を要求できます。第三者から読み込むと、その権限を第三者に渡すとともに、ページの描画に不可欠な別ホストを追加します。Subresource Integrityを使えば、ページ側でファイルの暗号学的ハッシュを指定でき、ブラウザーはバイト列が違うファイルを拒否します。

実際の活用例

2026年7月のスタイルシート改修は、直接埋め込んだ一つのリソースが、どれほど負担を左右するかを示しています。フレームワークを使わない文書サイトなのに、87 KBのバンドルが、コンテンツの到着前に使うページ重量の大部分を占めていました。最大のスタイルシートの三分の二は、base64で符号化した一つのフォントで、約42 KBのフォントを約56 KBのテキストで表していました。初回描画を速めるために埋め込んだものでしたが、外部ファイルのプリロードでも同じ利点が得られ、そのウェイトが必要かどうかにかかわらず、すべてのページが負担していました。取り出すと、そのファイルは84 KBから28に、バンドル全体は87 KBから31 KBに減りました。代わりに各フォントウェイトをプリロードします。Base64は三バイトを四文字で表すため、バイナリーファイルをシート内に埋め込むと、単独で配信する場合より常に約三分の一重くなります。

紙も、スタイルシートが対応すべきメディアです。このサイトの事例記事は、誰かが印刷して打ち合わせへ持っていくものです。2026年8月の実装では、637行のスタイルシート一つにページルール一つと印刷ブロック一つを設け、用紙サイズが事前に分からなくても、両方のカラーテーマで印刷できるようにしました。ページルールは余白を定め、用紙サイズは意図的に指定しません。行長を約七十二文字にすると、横向きの紙で一行が約百十文字まで延びるのを防げます。描画エンジンの四つの挙動を落とし穴として記録しており、どれも実際に不具合を起こしました。あるエンジンはビューポート単位をページボックス基準で、別のエンジンはウィンドウ基準で測り、各行の三分の一を切り落としました。固定配置の要素は各用紙に描かれました。ダークテーマ用に白文字を既定とするパレットでは、四つの節ラベルが白地に白で印刷されました。ページをまたいで分割できないflexやgridのボックスは、白紙の一枚目を生みました。付随する六項目のチェックでは、用紙と向きの六通りの組み合わせ、両方の配色設定、実際のページ数を確認しました。余分なコメント終端によってCSSリンターがルール全体を読み飛ばし、二度も問題なしと判定したことを見つけたのも、そのチェックでした。

セキュリティのページでは、スタイルシートにSubresource Integrityを使うと説明しています。バンドルのフィンガープリントとハッシュをビルド時に計算するため、読者のブラウザーは、改変されずに届いたかを検証できます。同じページは、ブラウザーのネットワークパネルでホストの数を数えるよう勧め、その答えは、このサイト自身のホスト一つです。

問題が起きるところ

ブラウザーが存在を知るのが遅いシートは、適用も遅れます。body要素の末尾に置いたリンクやスクリプトが挿入するシートは、最初の描画後に届くことがあり、素のテキストが突然レイアウトされたり、完成したレイアウトがずれたりします。一方、描画をブロックするシートの `@import` 連鎖は、最初の描画までの待ち時間を延ばすことがあります。ソースでは、インポートの連鎖も無害に見えます。

```
/* site.css: the browser finds base.css only after this file arrives */
@import url("base.css");
@import url("theme.css");
```

スタイルシートのリンクはheadに置き、インポート先のシートの内容は、ビルド時にバンドルへまとめます。

デプロイによって、新しいマークアップと古いルールが組み合わさることがあります。キャッシュに前のシートが残る読者は、その古いファイルにないクラスを前提とするHTMLを受け取り、キャッシュが切れるまでレイアウトが崩れます。一方、空のキャッシュでテストする人には、正しく動くサイトに見えます。変わらないシート名と長いキャッシュ期間の組み合わせが原因であり、フィンガープリント付きの名前にすれば取り除けます。

誤ったコンテンツタイプで配信されたシートは無視されます。HTML標準で、リンクしたシートを適用するのは、 `text/css` として届く場合だけです。ただし、quirksモードのページで同一オリジンのファイルを使う場合には例外があります。サーバーが `.css` ファイルを `text/plain` としたり、存在しないファイルにHTMLページと成功ステータスを返したりすると、ネットワークパネルでは成功したように見えるのに、ページにはスタイルが付きません。デプロイ後のファイルのヘッダーを確認します。

ハッシュ計算後にファイルが変わると、整合性ハッシュは古くなります。サーバー上での手修正、転送中にコードを縮小するCDN、最後の書き換え前にハッシュを計算するビルドなどが原因です。ブラウザーは不一致をネットワークエラーとして扱い、そのシートなしで描画します。別オリジンの整合性検証にはCORSも必要で、リンクの `crossorigin` 属性と、対応するサーバーヘッダーを使います。最終ファイルを書き出す工程でハッシュを計算します。

公開CDNからフレームワークのシートを読み込む理由として、以前は、ほかのサイトから読者がすでにキャッシュしていることが挙げられていました。Fetch標準はHTTPキャッシュをトップレベルサイトごとに分割するため、各サイト用のコピーを別々に取得、保存します。残るのは、他者が管理する二つ目のホストがクリティカルパスに入ることです。サイト自身のホストから配信すれば、転送バイト数は同じで、余分な接続は不要です。

参考資料

- CSS Object Model (CSSOM) — ブラウザー内部のスタイルシートと、スクリプトがそれを読み、変更するインターフェースを説明しています。
- HTML標準: リンク — `stylesheet` リンク型、コンテンツタイプの規則、描画をブロックする条件を説明しています。
- Subresource Integrity — `integrity` 属性と、検査に通らないファイルに対するブラウザーの処理を定めています。
- Fetch標準: HTTPキャッシュの分割 — トップレベルサイトに基づくキャッシュキーの仕組みを説明しています。
- CSS Paged Media Module Level 3 — 印刷用スタイルシートで扱う `@page` ルールとページの余白を説明しています。
- Media Queries Level 5 — 印刷、画面、配色設定など、 `media` 属性で判定できる条件を説明しています。

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

「スタイルシートとは？」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/stylesheet/