Engineer

Таблиця стилів

🖌 файл, у якому лежить CSS сторінки

Що це таке

Таблиця стилів (англійською stylesheet, у специфікаціях style sheet, у розмові часто просто CSS-файл) — це набір правил CSS, який браузер завантажує, розбирає й застосовує до документа як одне ціле.

Сторінка може отримати таблицю стилів чотирма шляхами. Елемент `link` з `rel="stylesheet"` вказує на окремий файл. Елемент `style` містить правила просто всередині HTML. Правило `@import` на початку однієї таблиці стилів підтягує іншу. А скрипт може створити таблицю через `new CSSStyleSheet()` і приєднати її до документа чи shadow root через `adoptedStyleSheets`. Перші два способи в `head` сторінки виглядають так:

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

Таблиці стилів автора сайту — не єдині учасники. Кожен браузер має власну типову таблицю стилів, тому заголовок без жодного оформлення все одно виходить великим і жирним, а читач може додати власну таблицю стилів користувача; каскад зважує ці три походження одне проти одного. Атрибут `style` на окремому елементі — інша річ: оголошення без селектора, які належать цьому елементу, а не таблиці.

Після завантаження браузер подає таблицю стилів як об’єкт `CSSStyleSheet` у CSS Object Model — специфікації, з якої походить визначення цього терміна. Об’єкт зберігає елемент-власника, медіа, до яких таблиця застосовується, її заголовок, ознаку вимкнення та список правил, а `document.styleSheets` перелічує таблиці сторінки по порядку. Один прапорець існує заради приватності: таблиця, отримана з іншого походження (origin) без CORS, не має позначки origin-clean, і скрипт, який просить її правила, отримує натомість помилку безпеки.

Атрибут `media` обмежує таблицю стилів певним середовищем. Таблицю з `media="print"` браузер усе одно завантажує, але чекає лише на ті таблиці, чиї медіа збігаються, тож екран не чекає на таблицю для друку.

Застосуйте поняття до рішення

Застосуйте «Що таке таблиця стилів?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Інтерфейси, що швидко завантажуються, працюють на телефоні й поводяться однаково в усіх браузерах ваших клієнтів.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Керувала full-stack-розробкою GIS-карт — full-stack-картографія — від бази даних до браузера

Керувала full-stack-розробкою GIS-карт Зв'язатися з нами →

Чому це важливо

Таблиця стилів визначає, що браузер мусить завантажити, перш ніж щось показати. Стандарт HTML вважає увімкнене посилання на таблицю стилів, знайдене парсером до появи елемента body, таким, що блокує рендеринг, коли його медіаумови відповідають середовищу, тож перше промальовування зазвичай чекає, доки файл прибуде й буде розібраний, замість того щоб намалювати сторінку без стилів і перемалювати її на очах у читача. Розмір файла, його хост і кількість мережевих обмінів, потрібних, щоб його виявити, — усе це стоїть між кліком і першим, що бачить читач.

Перше рішення — скільки файлів. Один пакет, спільний для всіх сторінок, завантажується раз і далі береться з кешу; окрема таблиця для кожної сторінки надсилає менше на початку й більше, коли читач переходить сайтом. Правила, вбудовані в елемент `style`, не потребують окремого запиту, але їдуть знову з кожною сторінкою, бо їх неможливо кешувати окремо від документа, що їх несе. Кожен `@import` додає крок виявлення:

браузер дізнається імпортовану адресу, коли читає таблицю стилів, що її імпортує, а завантаження імпорту, якого немає в кеші, може додати ще один мережевий обмін.

Друге — кешування. Таблиця стилів, що лежить у кеші рік, пришвидшує кожну наступну сторінку, але зміна мусить дійти до читачів, які досі тримають стару копію. Звична відповідь — ім'я файла з відбитком, тобто з хешем вмісту в самому імені, тож змінений файл має нову адресу, а старий може лишатися в кеші скільки завгодно, тоді як HTML, що на нього вказує, тримають короткоживучим.

Третє — довіра. Таблиця стилів вирішує, що видно і де; вона може сховати попередження, змінити вигляд кнопки або запросити зображення з будь-якої адреси, яку називають її правила. Завантаження таблиці від третьої сторони віддає цю владу над сторінкою третій стороні разом із хостом, без якого сторінка не відобразиться. Subresource Integrity дає сторінці змогу вказати криптографічний хеш файла, і браузер відкидає будь-який файл, чиї байти відрізняються.

Як це виглядає на практиці

Переробка таблиць стилів сайту в липні 2026 року показує, наскільки один вбудований ресурс може переважити все інше. За розміру 87 КБ пакет був, для документного сайту без фреймворку, більшою частиною ваги сторінки, витраченою ще до прибуття будь-якого вмісту, і дві третини найбільшої таблиці стилів виявилися одним шрифтом, закодованим у base64: приблизно 56 КБ тексту, що кодували приблизно 42 КБ шрифту. Його вбудували заради виграшу в першому промальовуванні, який попередньо завантажений зовнішній файл дає й так, і кожна сторінка платила за нього незалежно від того, чи була потрібна саме ця насиченість шрифту. Вилучення шрифту зменшило розмір цього файла з 84 КБ до 28, а весь пакет — з 87 КБ до 31 КБ, і насиченості шрифту натомість попередньо завантажуються. Base64 записує кожні три байти чотирма символами, тож двійковий файл, вбудований у таблицю стилів, завжди важить приблизно на третину більше за той самий файл, відданий окремо.

Папір теж є носієм, якому таблиця стилів мусить служити. Кейси сайту — це те, що хтось роздруковує й несе на зустріч. У реалізації від серпня 2026 року одна таблиця стилів на 637 рядків з єдиним правилом сторінки й одним блоком друку дала їм змогу друкуватися на невідомому папері в обох

колірних темах. Правило сторінки задає поле й навмисно не задає розміру паперу, а довжина рядка приблизно в сімдесят два символи не дає рядку на альбомному аркуші розтягнутися приблизно до ста десяти символів. Чотири види поведінки рушіїв записано як пастки, і кожен спричинив справжній дефект: одиниці області перегляду, які один рушій обчислює відносно блока сторінки, а інший — відносно вікна, через що обрізалася третина кожного рядка; елементи з фіксованим положенням, які малюються на кожному аркуші; палітра, що для темної теми усталено бере білий текст і надрукувала чотири підзаголовки розділів білим по білому; і контейнер flex або grid, який не розбивається між сторінками, через що перший аркуш виявився порожнім. Відповідна перевірка з шести частин охоплювала шість комбінацій паперу й орієнтації, обидва налаштування колірної схеми та справжню кількість сторінок, і саме вона помітила, що випадковий термінатор коментаря змусив лінтер CSS проковтнути ціле правило і двічі пройти чисто.

Сторінка безпеки сайту називає Subresource Integrity для таблиці стилів: під час збирання пакет отримує відбиток і хеш, тож браузер читача може перевірити, що він надійшов без змін. Та сама сторінка пропонує читачеві порахувати хости в панелі мережі браузера й дає відповідь: він один, і це хост самого сайту.

Де виникають проблеми

Таблиця стилів, про яку браузер дізнається пізно, відкладає застосування стилів. Посилання в кінці `body` або таблиця стилів, вставлена скриптом, може надійти після першого промальовування, тож сирий текст перескакує в макет або готовий макет зсувається. Ланцюжок правил `@import` у таблиці, що блокує рендеринг, натомість може подовжити очікування першого промальовування. Ланцюжок імпортів у джерелі виглядає невинно:

```
/* site.css: браузер знайде base.css лише після прибуття цього файла */
@import url("base.css");
@import url("theme.css");
```

Ставте посилання на таблиці стилів у `head` і вбудовуйте імпорти в пакет під час збирання.

Розгортання може звести нову розмітку зі старими правилами. Читачі, чий кеш тримає попередню таблицю стилів, отримують HTML, що чекає на класи, про які старий файл ніколи не чув, і макет у них ламається, доки не спливе термін кешу, тоді як кожен, хто тестує зі свіжим кешем, бачить робочий сайт. Причина — ім'я таблиці стилів, яке ніколи не змінюється, разом із довгим терміном кешування, і ім'я з відбитком її усуває.

Таблицю стилів, віддану з неправильним типом вмісту, браузер ігнорує. Стандарт HTML застосовує підключену таблицю стилів лише тоді, коли вона приходить як `text/css`, з одним винятком для файла з того самого походження на сторінці в режимі quirks. Сервер, який позначає файли `.css` як `text/plain` або відповідає на відсутній файл HTML-сторінкою з успішним статусом, лишає сторінку без стилів, хоча панель мережі показує запит, що начебто вдався. Перевіряйте заголовок розгорнутого файла.

Хеш цілісності застаріває, коли файл змінюється після хешування: ручна правка на сервері, CDN, що мініфікує файл дорогою, або збирання, яке хешує файл перед останнім переписуванням. Браузер вважає розбіжність мережевою помилкою й показує сторінку без таблиці стилів. З іншого походження цілісність потребує ще й CORS: атрибута `crossorigin` на посиланні й відповідного заголовка від сервера. Обчислюйте хеш на тому кроці, що записує остаточний файл.

Таблицю стилів фреймворку з публічного CDN колись виправдовували тим, що читачі вже мають її в кеші з інших сайтів. Стандарт Fetch розділяє HTTP-кеш за сайтом верхнього рівня, тож кожен сайт завантажує й зберігає свою копію окремо, а лишається додатковий хост на критичному шляху під чужим контролем. Віддавати файл із власного хоста сайту коштує тих самих байтів без зайвого з'єднання.

Що почитати далі

- CSS Object Model (CSSOM) — що таке таблиця стилів усередині браузера та інтерфейс, яким скрипти її читають і змінюють.
- HTML Standard: Links — тип посилання `stylesheet`, правило щодо типу вмісту й те, коли таблиця стилів блокує рендеринг.
- Subresource Integrity — атрибут `integrity` і те, що браузер робить із файлом, який не пройшов перевірку.
- Fetch Standard: HTTP cache partitions — як ключ кешу будується від сайту верхнього рівня.

- CSS Paged Media Module Level 3 — правило `@page` і поля сторінки, з якими працює таблиця стилів для друку.
- Media Queries Level 5 — друк, екран, вподоби колірної схеми та інші умови, які може перевіряти атрибут `media` .

Словник

Обирайте за фактичними умовами

Опишіть робочий процес, наявні інтерфейси та користувачів, яким потрібна зміна. Ми можемо визначити перевірюваний результат і спосіб передачі, який дасть вашій команді змогу підтримувати його.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке таблиця стилів?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

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

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/stylesheet/