Engineer

Stylesheet

🖌 filen, der holder en sides CSS

Hvad det er

Et stylesheet, på dansk også kaldt et stilark, i specifikationerne skrevet style sheet og i daglig tale ofte bare CSS-filen, er en samling CSS-regler, som en browser henter, fortolker og anvender på et dokument som én enhed.

Der er fire veje ind på en side. Et `link` -element med `rel="stylesheet"` peger på en separat fil. Et `style` -element bærer reglerne inde i selve HTML'en. En `@import` -regel øverst i ét stylesheet trækker et andet ind. Og et script kan bygge et med `new CSSStyleSheet()` og hænge det på et dokument eller en shadow root via `adoptedStyleSheets` . De to første ser sådan ud i sidens `head` :

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

Forfatterens stylesheets er ikke de eneste i spil. Hver browser har sit eget standard-stylesheet med, og derfor bliver en overskrift uden styling stadig stor og fed, og en læser kan tilføje et brugerstylesheet med egne indstillinger; kaskaden vejer de tre oprindelser mod hinanden. `style` -attributten på et enkelt element er noget andet: deklarationer uden selektor, som hører til elementet og til intet stylesheet.

Når et stylesheet er indlæst, repræsenterer browseren det som et `CSSStyleSheet` -objekt i CSS Object Model, den specifikation, begrebets definition stammer fra. Objektet holder styr på det element, der ejer stylesheetet, de

medier, det gælder for, dets titel, om det er slået fra, og dets liste af regler, og `document.styleSheets` viser sidens stylesheets i rækkefølge. Ét flag findes af hensyn til privatlivet: et stylesheet hentet fra en anden origin uden CORS er ikke origin-clean, og et script, der beder om dets regler, får en sikkerhedsfejl i stedet.

`media` -attributten begrænser et stylesheet til et miljø. Et stylesheet med `media="print"` hentes stadig, men browseren venter kun på stylesheets, hvis medie matcher, så skærmen venter ikke på et printstylesheet.

Brug begrebet i en beslutning

Brug »Hvad er et stylesheet?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Brugerflader, der indlæses hurtigt, virker på en telefon og opfører sig ens i alle de browsere, jeres kunder nu bruger.

Se vores ydelser Kontakt os →

Undersøg antagelserne bag ordet

Brug forklaringen og det tilknyttede arbejde til at formulere spørgsmål til jeres projekt. Vi kan undersøge antagelserne sammen, før et værktøj vælges. Har styret full-stack-udviklingen af GIS-kort — full stack-kortløsning, fra database til browser

Har styret full-stack-udviklingen af GIS-kort Kontakt os →

Hvorfor det betyder noget

Et stylesheet bestemmer, hvad browseren skal hente, før den viser noget som helst. HTML-standarden regner et aktiveret stylesheet-link for renderblokerende, hvis parseren finder det, før body-elementet findes, og dets mediebetingelse passer til omgivelserne. Browseren venter derfor normalt med sin første optegning, til filen er ankommet og fortolket, frem for først at tegne siden uden styling og derefter tegne den om for øjnene af læseren. Filens størrelse, dens vært og de rundture, der skal til for at opdage den, ligger alle mellem et klik og det første, læseren ser.

Antallet af filer er den første beslutning. Ét bundt, som alle sider deler, hentes én gang og kommer derefter fra cachen; et stylesheet pr. side sender mindre fra start og mere, efterhånden som læseren bevæger sig rundt. Regler indlejret i et `style` -element koster ingen ekstra forespørgsel, men de rejser med igen på hver side, fordi de ikke kan caches adskilt fra det dokument, der bærer dem. Hver `@import` tilføjer et trin i opdagelsen: browseren lærer den importerede adresse, mens den læser det importerende stylesheet, og hentning af en import, der ikke er cachet, kan tilføje endnu en netværksrundtur.

Caching er den anden. Et stylesheet, der ligger i cachen i et år, gør hver senere side hurtigere, men en ændring skal så nå læsere, der stadig har den gamle kopi. Det sædvanlige svar er et filnavn med fingeraftryk, hvor en hash af indholdet indgår i navnet, så en ændret fil får en ny adresse, og en gammel kan blive i cachen, så længe nogen ønsker det, mens den HTML, der peger på den, holdes kortlivet.

Tillid er den tredje. Et stylesheet bestemmer, hvad der er synligt og hvor; det kan skjule en advarsel, give en knap nyt udseende eller hente billeder fra enhver adresse, dets regler nævner. At hente et fra en tredjepart giver tredjeparten den magt over siden, sammen med en vært, siden ikke kan vises uden. Subresource Integrity lader siden angive en kryptografisk hash for filen, og browseren afviser enhver fil, hvis bytes afviger.

Sådan ser det ud i praksis

Omlægningen af sitets stylesheets i juli 2026 viser, hvor meget ét indlejret aktiv kan vippe balancen. Med 87 KB var bundtet, for et dokumentsite uden framework, det meste af en sides vægt brugt, før noget indhold ankom, og to tredjedele af det største stylesheet viste sig at være én base64-kodet font: omkring 56 KB tekst, der kodede omkring 42 KB font. Den var indlejret for en gevinst ved første optegning, som en forudindlæst ekstern fil alligevel får, og hver side betalte for den, uanset om vægten var nødvendig. At tage den ud bragte den fil fra 84 KB til 28 og hele bundtet fra 87 KB til 31 KB, og fontvægtene forudindlæses i stedet. Base64 skriver hver tre bytes som fire tegn, så en binær fil indlejret i et stylesheet vejer altid omkring en tredjedel mere end den samme fil serveret for sig selv.

Papir er også et medie, et stylesheet skal betjene. Sitets casestudier er det, nogen printer og tager med til et møde. I implementeringen fra august 2026 fik ét stylesheet på 637 linjer med én sideregel og én printblok dem til at printe på ukendt papir i begge farvetemaer. Sidereglen sætter en margen og bevidst ingen papirstørrelse, og en tekstbredde på omkring tooghalvfjerds tegn holder et liggende ark fra at lade en linje løbe til omkring hundrede og ti. Fire typer motoradfærd er skrevet ned som fælder, og hver af dem har forårsaget en reel defekt: viewport-enheder, der resolverer til sideboksen i én motor og til vinduet i en anden, hvilket skar en tredjedel af hver linje af; fastplacerede elementer, der males på hvert ark; en palet, der falder tilbage på hvid tekst til det mørke tema og printede fire afsnitsetiketter hvidt på hvidt; og en flex- eller gitterboks, som paginerede medier ikke kan dele, hvilket gav et blankt første ark. Det tilhørende tjek i seks dele dækkede seks papir- og orienteringskombinationer, begge

farveskemaindstillinger og det virkelige sideantal, og det var det tjek, der opdagede, at en efterladt kommentarafslutning havde fået CSS-linteren til at sluge en hel regel og bestå rent to gange.

Sitets sikkerhedsside nævner Subresource Integrity på stylesheetet: bundtet får fingeraftryk og hash, når sitet bygges, så læserens browser kan verificere, at det kom frem uændret. Samme side beder læseren tælle værterne i browserens netværksfane og giver svaret: én, sitets egen.

Hvor det går galt

Et stylesheet, browseren hører om sent, bliver også taget i brug sent. Et link nederst i `body` eller et stylesheet indsat af et script kan ankomme efter første optegning, så rå tekst hopper ind i et layout, eller et færdigt layout flytter sig. En kæde af `@import` -regler i et renderblokerende stylesheet kan i stedet forlænge ventetiden på første optegning. En importkæde ser uskyldig ud i kilden:

```
/* site.css: browseren finder base.css først, når denne fil er ankommet */
@import url("base.css");
@import url("theme.css");
```

Læg stylesheet-links i `head` , og flad importerne ud i bundtet ved build.

En udrulning kan parre ny markup med gamle regler. Læsere, hvis cache rummer det forrige stylesheet, får HTML, der forventer klasser, den gamle fil aldrig har hørt om, og deres layout går i stykker, til cachen udløber, mens alle, der tester med frisk cache, ser et site, der virker. Årsagen er et stylesheet-navn, der aldrig ændres, kombineret med lang levetid i cachen, og et navn med fingeraftryk fjerner den.

Et stylesheet serveret med forkert indholdstype bliver ignoreret. HTML-standarden anvender kun et linket stylesheet, når det ankommer som `text/css` , med én undtagelse for en fil fra samme origin på en side i quirks mode. En server, der mærker `.css` -filer som `text/plain` , eller som svarer på en manglende fil med en HTML-side og en successtatus, efterlader siden uden styling, mens netværksfanen viser en forespørgsel, der tilsyneladende lykkedes. Kontrollér headeren på den udrullede fil.

En integritets-hash bliver forældet, når filen ændres efter hashingen: en håndrettelse på serveren, et CDN, der minificerer undervejs, eller et build, der hasher før sin sidste omskrivning. Browseren behandler uoverensstemmelsen som en netværksfejl og viser siden uden stylesheetet. Fra en anden origin kræver integritet også CORS: en `crossorigin` -attribut på linket og en matchende header fra serveren. Beregn hashen i det trin, der skriver den endelige fil.

Et framework-stylesheet fra et offentligt CDN blev engang forsvaret med, at læserne allerede havde det i cachen fra andre sites. Fetch-standarden partitionerer HTTP-cachen efter topniveau-site, så hvert sites kopi hentes og gemmes for sig, og tilbage står en ekstra vært på den kritiske sti under en andens kontrol. At servere filen fra sitets egen vært koster de samme bytes uden den ekstra forbindelse.

Læs mere

- CSS Object Model (CSSOM) — hvad et stylesheet er inde i browseren, og den grænseflade, scripts bruger til at læse og ændre det.
- HTML Standard: Links — linktypen `stylesheet` , reglen om indholdstype, og hvornår et stylesheet blokerer rendering.
- Subresource Integrity — `integrity` -attributten, og hvad browseren gør med en fil, der dumper kontrollen.
- Fetch Standard: HTTP cache partitions — hvordan cachen nøgles efter topniveau-site.
- CSS Paged Media Module Level 3 — `@page` -reglen og de sidemargener, et printstylesheet arbejder med.
- Media Queries Level 5 — print, skærm, farveskemaindstillinger og de andre betingelser, en `media` -attribut kan teste.

Ordliste

Vælg ud fra jeres faktiske vilkår

Beskriv arbejdsgangen, de eksisterende grænseflader og de brugere, der har brug for ændringen. Vi kan definere en testbar leverance og en overdragelse, der gør teamet i stand til at vedligeholde løsningen.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er et stylesheet?« gælder i jeres organisation. Et konkret eksempel fra jeres nuværende system er nok til at begynde samtalen.

Se vores ydelser

welcome@engineer.company +45 50 10 61 84 København, Danmark

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://engineer.company/da/glossary/stylesheet/