Engineer

Stilark

🖌 filen som holder en sides CSS

Hva det er

Et stilark, skrevet «style sheet» i spesifikasjonene og ofte bare kalt CSS-filen, er én samling CSS-regler som en nettleser henter, parser og bruker på et dokument som en samlet enhet.

En side kan få stilark på fire måter. Et `link` -element med `rel="stylesheet"` peker på en egen fil. Et `style` -element bærer reglene inne i selve HTML-en. En `@import` -regel øverst i ett stilark trekker inn et annet. Og et skript kan lage et med `new CSSStyleSheet()` og feste det til et dokument eller en skyggerot gjennom `adoptedStyleSheets` . De to første ser slik ut i hodet på en side:

```
<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 egne stilark er langt fra de eneste som er i spill. Hver nettleser kommer med et standardstilark, og det er grunnen til at en overskrift uten stil likevel blir stor og fet. En leser kan i tillegg legge til et eget brukerstilark, og kaskaden veier de tre opphavene mot hverandre. Attributtet `style` på ett enkelt element er noe annet: deklarasjoner uten selektor, som hører til det elementet og ikke til noe stilark.

Når det er lastet, blir et stilark et `CSSStyleSheet` -objekt i CSS Object Model, spesifikasjonen definisjonen av dette ordet er hentet fra. Objektet holder rede på elementet som eier stilarket, mediene det gjelder for, tittelen, om det er slått

av, og listen over regler, og `document.styleSheets` lister stilarkene på en side i rekkefølge. Ett flagg finnes av hensyn til personvernet: et stilark hentet fra et annet opphav uten CORS regnes ikke som «origin-clean», og et skript som ber om reglene i det, får en sikkerhetsfeil i stedet.

Attributtet `media` begrenser et stilark til ett bestemt miljø. Et stilark merket `media="print"` lastes fortsatt ned, men nettleseren venter bare på stilark der mediet stemmer, så et stilark for utskrift forsinker ikke visningen på skjermen.

Bruk begrepet i en beslutning

Bruk «Hva er et stilark?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Grensesnitt som laster raskt, fungerer på mobilen og oppfører seg likt i alle nettlesere kundene dine måtte bruke.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Styrte fullstack-utviklingen av GIS-kart — fullstack-kartutvikling, fra database til nettleser

Styrte fullstack-utviklingen av GIS-kart Ta kontakt →

Hvorfor det er viktig

Et stilark avgjør hva en nettleser må hente før den viser noe som helst. HTML-standarden regner en aktivert stilarklenke som parseren finner før body-elementet finnes, som blokkerende for gjengivelsen når mediebetingelsen passer til visningsmiljøet. Den første opptegningen venter derfor normalt til filen er kommet fram og er parset, i stedet for å tegne siden uten stil og så tegne den på nytt foran leseren. Størrelsen på filen, verten den ligger på og rundturene som trengs for å oppdage den, ligger alle mellom et klikk og det første leseren ser.

Antall filer er den første beslutningen. Én pakke som hver side deler, hentes én gang og leveres deretter fra cachen, mens et stilark per side sender mindre i starten og mer etter hvert som leseren beveger seg rundt. Regler lagt rett inn i et `style` -element koster ingen ekstra forespørsel, men de reiser på nytt med hver side, fordi de ikke kan bufres for seg, atskilt fra dokumentet som bærer dem. Hver `@import` legger til et trinn for å finne ressursen: nettleseren får vite den importerte adressen mens den leser stilarket som importerer, og henting av en import som ikke er bufret, kan legge til enda en nettverksrundtur.

Hurtigbufring er den andre. Et stilark som får ligge i bufferen i et år, gjør hver senere side raskere, men en endring må da nå lesere som sitter med den gamle kopien. Det vanlige svaret er et filnavn med fingeravtrykk, der en hash av innholdet står inne i navnet. En endret fil får da en ny adresse, og en gammel kan bli liggende i bufferen så lenge noen vil, mens HTML-en som peker på den, holdes kortlevd.

Tillit er den tredje. Et stilark avgjør hva som er synlig, og hvor. Det kan skjule en advarsel, gi en knapp et helt nytt utseende eller hente bilder fra en hvilken som helst adresse reglene navngir. Å laste et stilark fra en tredjepart gir den makten til tredjeparten, og gir siden samtidig en vert den ikke kan vises uten. Subresource Integrity lar siden oppgi en kryptografisk hash for filen, og nettleseren avviser hver fil der bytene avviker.

Slik ser det ut i praksis

Omleggingen av nettstedets stilark i juli 2026 viser hvor langt én innebygd ressurs kan vippe regnestykket. Med sine 87 KB var pakken, for et dokumentnettsted uten rammeverk, mesteparten av vekten på en side, brukt før noe innhold var kommet fram. To tredjedeler av det største stilarket viste seg å være én font kodet i base64: omtrent 56 KB tekst som kodet omtrent 42 KB font. Den var bygd inn for en gevinst ved første opptegning som en forhåndslastet ekstern fil får uansett, og hver side betalte for den, enten vekten trengtes eller ei. Å ta den ut fikk den filen ned fra 84 KB til 28 og hele pakken ned fra 87 KB til 31 KB, med fontvektene forhåndslastet i stedet. Base64 skriver hver gruppe på tre byte som fire tegn, så en binærfil bygd inn i et stilark veier alltid omtrent en tredjedel mer enn den samme filen levert for seg selv.

Papir er også et medium et stilark må betjene. Casestudiene på nettstedet er det noen skriver ut og tar med inn i et møte. I implementeringen fra august 2026 sørget ett stilark på 637 linjer med én sideregel og én utskriftsblokk for at de ble skrevet ut riktig på ukjent papir i begge fargetemaene. Sideregelen setter en marg og bevisst ingen papirstørrelse, og en linjelengde på rundt syttito tegn hindrer at et ark i liggende format gir linjer på rundt hundre og ti. Fire typer oppførsel i nettlesermotorene er skrevet ned som feller, og hver av dem har ført til en reell feil: visningsområdeenheter, som én motor regner ut fra sideboksen og en annen fra vinduet, noe som kuttet bort en tredjedel av hver linje; elementer med fast posisjon som males på hvert eneste ark; en palett med hvit tekst som standard for det mørke temaet, som skrev ut fire seksjonsetiketter hvitt på hvitt; og en flex- eller grid-boks som sidedelte medier ikke kan dele, som ga et tomt første ark. Den tilhørende sjekken i seks deler dekket seks

kombinasjoner av papir og retning, begge innstillingene for fargevalg og det faktiske sidetallet. Det var den sjekken som merket det da en forvillet kommentaravslutning fikk CSS-linteren til å svelge en hel regel og passere feilfritt to ganger.

Slik arbeider vi sikkert, nettstedets side om sikkerhet, lister Subresource Integrity på stilarket: pakken får fingeravtrykk og hash når nettstedet bygges, så nettleseren til leseren kan bekrefte at den kom fram uendret. Den samme siden ber leseren telle vertene i nettverkspanelet i nettleseren og gir svaret selv: én, nettstedets egen.

Hvor det går galt

Et stilark nettleseren får vite om sent, blir tatt i bruk sent. En lenke nederst i `body` eller et stilark som et skript setter inn, kan komme fram etter den første opptegningen, så rå tekst hopper inn i et oppsett, eller et ferdig oppsett forskyver seg. En kjede av `@import` -regler i et stilark som blokkerer gjengivelsen, kan i stedet forlenge ventetiden før første opptegning. En importkjede ser harmløs ut i kilden:

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

Legg stilarklenkene i hodet på siden, og flat ut importene i pakken ved bygging.

En produksjonssetting kan koble ny oppmerking med gamle regler. Lesere som har det forrige stilarket i cachen, får HTML som venter seg klasser den gamle filen aldri har hørt om, og oppsettet deres er ødelagt til bufferen går ut, mens alle som tester med en tom buffer, ser et nettsted som virker. Årsaken er et stilarknavn som aldri endres, kombinert med lang levetid i bufferen, og et navn med fingeravtrykk fjerner den.

Et stilark som leveres med feil innholdstype, blir oversett. HTML-standarden tar et lenket stilark i bruk bare når det kommer som `text/css` , med ett unntak for en fil fra samme opphav på en side i quirks-modus. En server som merker `.css` -filer som `text/plain` , eller som svarer på en fil som mangler, med en HTML-side

og en statuskode som angir at forespørselen lyktes, etterlater siden uten stil, mens nettverkspanelet viser en forespørsel som tilsynelatende gikk bra. Sjekk headeren på filen slik den ligger i produksjon.

En integritetshash blir utdatert når filen endres etter at den ble hashet: en håndredigering på serveren, et CDN som minifiserer underveis, eller et bygg som regner ut hashen før den siste omskrivingen. Nettleseren behandler avviket som en nettverksfeil og viser siden uten stilarket. Fra et annet opphav krever integritet også CORS: et `crossorigin` -attributt på lenken og en tilsvarende header fra serveren. Regn ut hashen i det trinnet som skriver den endelige filen.

Et stilark for et rammeverk på et offentlig CDN ble en gang forsvart med at leserne allerede hadde det i cachen fra andre nettsteder. Fetch-standarden deler HTTP-cachen opp etter toppnivånettsted, så hvert nettsted henter og lagrer sin egen kopi, og det som er igjen, er en ekstra vert på den kritiske veien, styrt av noen andre. Å levere filen fra nettstedets egen vert koster de samme bytene, men sparer den ekstra tilkoblingen.

Les mer

- CSS Object Model (CSSOM) — hva et stilark er inne i nettleseren, og grensesnittet skript bruker til å lese og endre det.
- HTML Standard: Links — lenketypen `stylesheet` , regelen om innholdstype og når et stilark blokkerer gjengivelsen.
- Subresource Integrity — attributtet `integrity` og hva nettleseren gjør med en fil som feiler sjekken.
- Fetch Standard: HTTP cache partitions — hvordan nøkkelen til cachen bygges på toppnivånettstedet.
- CSS Paged Media Module Level 3 — `@page` -regelen og sidemargene et utskriftsstilark arbeider med.
- Media Queries Level 5 — print, screen, innstillingene for fargevalg og de andre betingelsene et `media` -attributt kan teste.

Ordliste

Velg ut fra deres faktiske rammer

Beskriv arbeidsflyten, eksisterende grensesnitt og brukerne som trenger endringen. Vi kan definere en testbar leveranse og en overlevering som gjør teamet i stand til å vedlikeholde løsningen.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er et stilark?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

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

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

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

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/stylesheet/