Engineer

CSS

språket som anger hur en sida ser ut

Vad det är

CSS är det regelspråk som en webbläsare tillämpar på ett dokument för att avgöra hur varje element ritas ut: färg, storlek, avstånd, placering, typsnitt och rörelse, oavsett om dokumentet hamnar i en mobil, på en bred skärm eller på ett papper. Förkortningen står för Cascading Style Sheets, många säger fortfarande CSS3, och på svenska kallas filerna stilmallar.

En stilmall är en lista med regler. Varje regel parar ihop en selektor, som pekar ut element, med deklarationer, var och en bestående av en egenskap och ett värde:

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

Det är det första ordet i namnet som präglar det dagliga arbetet. Flera regler gör ofta anspråk på samma egenskap hos samma element, och kaskaden avgör vilken som vinner. Dess huvudsteg är i tur och ordning ursprung och vikt (webbläsarens standardstilar, läsarens egna inställningar, webbplatsens stilmall och om en deklaration är märkt `!important`), sedan kaskadlager, sedan specificitet och sist ordningen i källan, där en senare regel slår en tidigare med samma tyngd. Specificiteten räknas per selektor som tre tal: id, sedan klasser, attribut och pseudoklasser, sedan elementnamn. Den universella selektorn `*` räknas inte alls. Arv verkar vid sidan av kaskaden: vissa egenskaper, som färg och typsnitt, förs vidare till ett elements barn, medan andra, som marginaler och ramar, stannar där de sattes.

CSS har inget enskilt versionsnummer. Efter CSS 2.1 gick arbetsgruppen över till moduler: nivå 3 bygger vidare på 2.1 en modul i taget, och därifrån får varje modul sina egna nivåer, så Selectors skrivs på nivå 4 medan moduler utan äldre motsvarighet börjar på nivå 1. ”CSS3” är det populära namnet på den modulära eran, och den årliga ögonblicksbild som definierar dagens CSS säger rakt ut att det inte finns någon CSS Level 4.

Den regel som väger tyngst i praktiken är hur språket behandlar fel. En webbläsare som stöter på en egenskap den inte känner till, eller ett värde som är ogiltigt för sin egenskap, ska ignorera just den deklarationen och tillämpa resten av regeln. Regeln finns för att nya egenskaper och ny syntax ska kunna läggas till utan att äldre webbläsare går sönder, och det är den som låter en webbplats använda en funktion som bara vissa webbläsare stöder. Priset är tystnad. Sidan renderas, lite fel, och det enda spåret är en överstruken rad i utvecklarverktygen som någon måste leta upp.

Det mesta som jämförs med CSS visar sig vara ett sätt att skriva det. Sass och Less är preprocessorer som kompileras till CSS. Ramverk med hjälpklasser, som Tailwind, genererar det ur klassnamn i märkningen. Bibliotek för CSS-in-JS skapar det ur komponenter i JavaScript, och ett `style` -attribut lägger deklarationer direkt på ett enda element. Vart och ett ändrar hur CSS skrivs; vart och ett slutar som deklarationer som kaskaden sorterar på exakt samma sätt.

Använd begreppet i ett beslut

Använd ”Vad är CSS?” i ett konkret beslut. Vi kan knyta begreppet till systemet ni bygger eller köper. Gränssnitt som laddar snabbt, fungerar i mobilen och beter sig likadant i alla webbläsare dina kunder råkar använda.

Se våra tjänster Hör av er →

Granska antagandena bakom ordet

Använd förklaringen och det länkade arbetet för att formulera frågor till projektet. Vi kan undersöka antagandena tillsammans innan ett verktyg väljs. Styrde full stack-utvecklingen av GIS-kartor — kartutveckling i full stack, från databas till webbläsare

Styrde full stack-utvecklingen av GIS-kartor Hör av er →

Varför det spelar roll

Beslutet som CSS tvingar på ett team är var en regel får misslyckas i tysthet. Språket släpper det som det inte kan tillämpa och låter en starkare regel vinna utan kommentar, så varje stilmall med några år på nacken innehåller deklarationer som inte gör någonting. Det ett team styr över är om någon får veta det, och när.

Tillgängligheten hänger mer på det än de flesta tror. Kontrast, en synlig fokusmarkering, att en begäran om minskad rörelse respekteras, ett användbart läge för tvingade färger, text som tål att förstoras: vart och ett är en handfull deklarationer, och vart och ett kan stängas av av en regel någon annanstans som råkar vara mer specifik. Sidan ser bra ut för alla som inte behövde funktionen.

Underhållskostnaden hänger också på det. En stilmall vars beteende ingen kan förutsäga bara växer, eftersom en borttagen regel kan förstöra en sida som ingen kontrollerade. Det vanliga slutläget är en fil där det mesta av vikten är överskrivningar av tidigare överskrivningar, och varje ändring görs genom att pröva sig fram tills skärmen ser rätt ut.

Kontrollen hänger allra mest på det. En webbläsare som tar emot det den kan och ignorerar resten kan inte fungera som testet. En linter som läser källan fångar stavfel och ogiltiga deklarationer innan de levereras. Den kan inte se vad en minifierare eller något annat byggsteg gör med filen efteråt, och den kan inte se vilken av två giltiga regler som vinner på en viss sida. Kontroller av den levererade filen och visuell jämförelse av renderade sidor täcker det som lintern inte når.

Att ordna kaskaden tidigt, med platt specificitet, namngivna lager och designtokens i anpassade egenskaper (custom properties), kostar lite disciplin per ändring. Att ordna om en stilmall som har vuxit utan den disciplinen kostar oftast en omskrivning.

Så ser det ut i praktiken

Webbplatsens egen sida med erkännanden beskriver ett enda CSS-paket, sammanfogat och försett med ett fingeravtryck när webbplatsen byggs, med de handritade pilarna och marginalanteckningarna ritade enbart i CSS, och med Stylelint som håller källkoden till sina regler i kvalitetsgrinden. Samma sida länkar till W3C CSS Validator och förvarnar om vad den kommer att säga. I augusti 2026 stannar dess profil före `system-ui`, `@property`, `clamp()` i en `font-size` och scrolldrivna animationer, så mot /terms/ rapporterade den 176 anmärkningar, varav 175 gällde funktioner som webbläsarna har och webbplatsen använder. Den sista var verklig. Ett `rotate: 0deg` i viloläge förlorade sin enhet i minifieraren, `rotate: 0` är ingen vinkel, och webbläsaren kastade hela deklarationen, så en animation levererades utan den rotation den skrevs med. Specifikationen CSS Values and Units tillåter en naken nolla som vinkel bara på

några få äldre ställen. Stylelint såg aldrig problemet, eftersom den läser källan och minifieraren skriver om den senare; en egen kontroll läser nu det som faktiskt levererades.

Specificitet var hela förklaringen i ett arbete med tillgänglighet. Webbplatsen respekterade inställningen för minskad rörelse med en enda universell regel som stängde av alla övergångar, och under en emulerad inställning animerades fortfarande 11 selektorer. En universell selektor ger noll i specificitet, så varje regel som bar en klass slog den. Åtgärden var strukturell snarare än en starkare selektor: allt som rör sig ligger nu i en del av stilmallen som kommer sent, och en linter underkänner en transform med övergång var som helst annars. Färgtoningar godkänns, eftersom de inte är rörelse, och rörelse måste skrivas som en övergång snarare än som en keyframe-animation, eftersom det är övergången som går att avbryta rent. Lärdomen som antecknades är kort: en regel som förlorar på specificitet fallerar i tysthet.

Färg är det som glider först i en stilmall, och webbplatsens palett slöts vid 28 färger, var och en dokumenterad med det kontrastförhållande den uppmäts till på den yta där den förekommer, och 11 varumärkesfärger deklarerades en gång som tokens. Två färger som ligger närmare varandra än ett perceptuellt avstånd på 0,02 räknas som en färg med två stavningar, och lintern hittade två nästan identiska stavningar så snart den kördes. Transparens skrivs med relativ färgsyntax i stället för med blandningsfunktionen, av ett skäl som sammanfattar hela sidan: kontrollen kan inte se igenom en blandning.

Var det går fel

En deklaration försvinner och sidan visas ändå

Ett enda fel tecken gör en deklaration ogiltig, och webbläsaren hoppar över den utan att stanna:

```
.card {
    padding: 16; /* en längd kräver en enhet: ignoreras */
    colour: #222; /* inte ett egenskapsnamn: ignoreras */
    margin-block: 1rem; /* tillämpas */
}
```

Kortet renderas med sin marginal men utan sin utfyllnad och sin färg, och webbläsaren säger ingenting om det. Linta källan i bygget och låt dess fel stoppa bygget. Där en minifierare, preprocessor eller bundler skriver om CSS-koden efter att lintern har körts bör resultatet också kontrolleras, eftersom det är den filen som webbläsaren läser.

En regel förlorar mot en starkare

En komponents regel möter en mer specifik selektor från något annat håll, eller en lika stark som laddas senare, och förlorar. Den vanliga reflexen är en längre selektor eller ett `!important`, som vinner i dag och höjer priset för nästa ändring. Håll specificiteten platt, med klasser i stället för id och lite nästling, och använd kaskadlager för att ställa hela grupper, som reset, bas, komponenter och hjälpklasser, i en uttrycklig ordning som specificiteten inte kan välta. För deklarationer med `!important` går lagerordningen åt andra hållet, så att det tidigaste lagret vinner, och det är värt att veta innan man griper efter det.

Stilmallen fortsätter bara att växa

Nytt arbete lägger till regler, eftersom det tar längre tid att bevisa att en gammal regel är oanvänd än att skriva en till. Färger och mått skrivs in på nytt som literaler bredvid de tokens som definierar dem, och nästan identiska färger förökar sig tills ingen kan säga vilken som avses. Håll det som delas i anpassade egenskaper, mät vilka regler en uppsättning sidor faktiskt använder och ta bort som en del av arbetet, med en storleksbudget i bygget som gör tillväxten synlig.

Layouten håller bara i en storlek

En design som har kontrollerats vid en enda bredd med kort engelsk text går sönder med längre översättningar, förstorad text och smala telefoner. Fasta höjder klipper av sitt innehåll, och fasta bredder tvingar fram rullning i sidled. Riktlinjerna för tillgänglighet begär text som kan förstoras till 200 % och innehåll som flödar om vid en bredd på 320 CSS-pixlar. Låt rutor få sin storlek av innehållet, använd `min()`, `max()` och `clamp()` för flytande mått och testa med långa strängar och zoom innan en layout kallas klar.

Läs mer

- CSS Snapshot 2026 — vilka moduler som utgör CSS i dag, och varför språket har nivåer i stället för versioner.
- CSS Cascading and Inheritance Level 5 — hela sorteringsordningen, kaskadlager och arv, steg för steg.
- Selectors Level 4 — varje selektor, och hur dess specificitet räknas ut.

- Syntax and basic data types — reglerna i CSS 2.1 för när en okänd egenskap eller ett otillåtet värde ska ignoreras, med exempel.
- CSS Values and Units Module Level 4 — längder, vinklar, `clamp()` och var en naken nolla fortfarande är tillåten.

Ordlista

Välj utifrån era faktiska förutsättningar

Beskriv arbetsflödet, befintliga gränssnitt och användarna som behöver förändringen. Vi kan definiera en testbar leverans och en överlämning som gör att teamet kan underhålla lösningen.

Se vad det kostar Hör av er →

Gör begreppen praktiska

Fråga hur ”Vad är CSS?” gäller i er organisation. Ett konkret exempel från ert nuvarande system räcker för att börja samtalet.

Se våra tjänster

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

Ta med dig den här sidan

Visitkort

Omsorgsfullt skapad med kärlek av D.

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

Upphovsrätt © 2025 – nu · Engineer ApS · Alla rättigheter förbehållna.

Utskrivet från https://engineer.company/sv/glossary/css/