Engineer

HTML

språket som en sidas struktur skrivs i

Vad det är

HTML, kort för HyperText Markup Language och fortfarande ofta kallat HTML5, är det textformat som varje webbsida levereras i: innehåll inneslutet i taggar som talar om vad varje del är, så att webbläsaren kan skilja en rubrik från ett stycke, en länk från en listpunkt och ett formulärfält från sin etikett.

En webbläsare läser texten uppifrån och ned och bygger ett träd av element ur den, Document Object Model, eller DOM. Det trädet är vad stilmallar formger, vad skript ändrar och vad skärmläsare läser upp. Arbetsfördelningen på webben är gammal och håller fortfarande: HTML säger vad något är, CSS hur det ser ut och JavaScript vad det gör när någon rör vid det. Ett element som `<nav>`, `<button>` eller `<h2>` bär en betydelse som alla dessa läsare kan agera på. En riktig `<button>` får tangentbordsfokus, en roll som hjälpmedel läser upp och aktiveras med Enter eller mellanslag, allt utan en enda rad extra kod.

Språket började som ett sätt att beskriva vetenskapliga dokument semantiskt och växte till lagret under nästan varje applikation med ett webbgränssnitt. De första fem åren, 1990–1995, tillbringade det vid CERN och sedan inom IETF. HTML 3.2 färdigställdes vid W3C 1997, och HTML4 följde strax efter. Under många år förde W3C och WHATWG var sin text som gled isär, tills de 2019 skrev under ett avtal om att arbeta på en enda: HTML Living Standard, som WHATWG förvaltar och som varje stor webbläsarmotor implementerar. Den har inget versionsnummer. På frågan om något är HTML5 svarar standarden med ett kort ja och kallar begreppet ett modeord för den moderna webbplattformen, inte en utgåva.

Hur förhåller sig HTML till det som det jämförs med? Markdown är en förkortad skrivform som ett byggverktyg gör om till HTML. PDF låser layouten till en tryckt sida, medan HTML flödar om till vilken skärm det än hamnar på. XHTML är samma

vokabulär skriven under XML:s strängare regler. JSX, Jinja och mallar i Hugo är sätt att producera HTML, och det som når webbläsaren från dem alla är fortfarande HTML. Ett team som väger en egenbyggd webbplats mot en sidbyggare väljer vem som skriver koden, aldrig om det finns någon.

Använd begreppet i ett beslut

Använd ”Vad är HTML?” 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

Få team väljer HTML med avsikt. Det kommer som det som ramverket, sidbyggaren eller publiceringssystemet råkar spotta ut. Det val ni egentligen gör är om den utdatan är en leverans som någon granskar eller en biprodukt som ingen läser, och följderna landar på tre ställen.

Vem som kan använda sidan kommer först. Hjälpmedel läser trädet, inte designen. Den som använder skärmläsare rör sig genom en sida via dess rubriker och landmärken; den som använder tangentbord tabbar från en länk eller kontroll till nästa. En stylad `<div>` med en klickhanterare ser ut som en knapp för en seende musanvändare och är vanlig text för de båda andra. Krav på webbtillgänglighet som WCAG 2.2 är till stor del skrivna mot det som koden exponerar, så att uppfylla dem efter lansering brukar betyda att skriva om mallar snarare än att justera färger.

Hur länge innehållet håller kommer därnäst. Webbläsare öppnar fortfarande sidor som skrevs på 1990-talet, eftersom bakåtkompatibilitet är en av de principer som standarden förvaltas efter. Ramverk byts ut med några års mellanrum. Innehåll i ren, giltig HTML kan flyttas till ett annat system, arkiveras, konverteras till andra format eller få en ny design utan att skrivas om. Innehåll som ligger begravt i en sidbyggares nästlade omslag måste oftast kopieras ut för hand.

Hastighet och räckvidd kommer sist. En sida som servern skickar komplett kan ritas upp medan byten anländer. En applikation som skickar ett nästan tomt skal och bygger dokumentet i JavaScript låter varje läsare vänta på skriptet, och allt som läser sidan utan att köra skriptet, från en länkförhandsvisning till många sökrobotar, ser bara skalet. För en produkt bakom en inloggning, där interaktionen är själva poängen, kan den avvägningen vara värd det. För marknadssidor, dokumentation och allt som ska hittas via sökning är komplett HTML det billigare förvalet.

Kostnaden för alla tre är densamma och liten: någon i teamet som läser HTML lika noga som kod. Semantisk kod kostar nästan ingenting när den skrivs från början och väldigt mycket när den ska byggas in i efterhand.

Så ser det ut i praktiken

Den här webbplatsen är det enklaste exemplet som finns till hands. Sidan med medverkande räknar upp vad som når besökarens webbläsare: HTML, en stilmall, lokala webbtypsnitt, bilder, utan körbar JavaScript. Varje sida skrivs ut i sin helhet som en fil på disk innan någon frågar efter den, i stället för att sättas ihop ur en mall vid varje anrop. Webbplatsens kvalitetsgrind validerar den HTML-koden med vnu och behandlar varje diagnostiskt meddelande som ett fel, och samma sida länkar till W3C:s Markup Validator, som kör samma motor mot den levande standarden, så att en läsare kan pröva påståendet i stället för att lita på det.

Tidigare kunduppdrag närmade sig HTML från andra hållet. I trettio webbplatser byggda från färdiga visuella designer omsattes Photoshop-filerna för hand till HTML. Måttstocken hade två sidor: trogen avstånden, typografin och detaljerna som designern avsåg, och flexibel nog att webbplatsen fortsatte att fungera första gången någon redigerade innehållet. Att träffa mockupen är halva jobbet; kod som överlever nästa redigering är den andra halvan.

Att hålla HTML som den kanoniska formen ger också utrymme att publicera andra format bredvid. När webbplatsen speglades till Gemini och Gopher producerade samma bygge 777 Gemini-dokument och 777 Gopher-menyer ur samma källa i det dokumenterade implementeringsskedet. En separat jämförelse av byggen mätte noll ändrade HTML-byte när formaten lades till; resultatet beskriver inte senare redaktionella eller visuella ändringar.

Var det går fel

Layouten förskjuts på sätt som ingen stilmall förklarar. En sida som börjar utan `<!DOCTYPE html>`, eller med en gammal och felformad doctype, försätter webbläsaren i quirks mode, ett renderingsläge som härmar äldre webbläsare. Procentuella höjder, tabellceller och radhöjder följer då gammalt beteende som avviker från CSS-specifikationerna, och resultatet skiljer sig från samma sida i standardläge. Lösningen är en rad överst i varje mall.

Trasig kod renderas, och trädet blir ett annat. HTML-parsern återhämtar sig från varje fel på ett definierat sätt, likadant i alla webbläsare, så misstagen syns sällan på skärmen. De dyker upp senare, när en selektor eller ett skript letar efter ett element någon annanstans än där parsern placerade det. Ett blockelement inuti ett stycke är det klassiska fallet:

```
<p>Opening line
  <div>A block placed inside it</div>
</p>
```

Parsern stänger stycket före `<div>` och gör sedan om det lösa `</p>` till ett andra, tomt stycke. Kör en validator i bygget och låt dess fel stoppa det; webbläsarens tolerans är en mekanism för återhämtning, och den granskar ingenting.

Varje kontroll blir en `div`. Klickbara `div` - och `span` -element, rubriker valda för sin storlek, nedsänkt text som används för att göra texten mindre: var och en ser rätt ut och berättar fel sak för hjälpmedlen. Tangentbordsanvändare passerar kontroller som aldrig får fokus, och den som använder skärmläsare förlorar de rubriker och landmärken som hen navigerar efter. En sida som saknar `<main>` -element eller `<h1>` tvingar dem att läsa från toppen varje gång. Använd elementet som namnger uppgiften, som `button`, `a`, `h2`, `main` eller `nav`, och låt CSS ändra utseendet.

Sidan förblir tom tills skriptet har körts. En klientrenderad webbplats skickar knappt mer än `<div id="app"></div>` och bygger dokumentet i webbläsaren. På en långsam telefon får läsaren vänta, och när skriptet kraschar eller ett filter blockerar det förblir sidan tom för gott. Rendera dokumentet på servern eller vid byggtid, och lägg interaktion ovanpå HTML som redan går att förstå på egen hand.

Läs mer

- HTML Living Standard — specifikationen som webbläsarna implementerar, förvaltad av WHATWG och uppdaterad löpande.
- Introduction to the HTML standard — historien från CERN till avtalet 2019, och standardens eget svar på frågan om HTML5.
- Parsing HTML documents — den felhantering som alla webbläsare delar, nedskriven regel för regel.
- The HTML syntax — doctype, element och attribut, och vad som gör ett dokument korrekt enligt standarden.
- The Nu Html Checker — motorn bakom vnu och W3C:s validator för kod, med dess användning på kommandoraden.
- DOM Standard — trädet som parsern bygger och de gränssnitt som skript använder för att ändra det.

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 HTML?” 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/html/