Engineer

HTML

sproget, en sides struktur er skrevet i

Hvad det er

HTML, en forkortelse af HyperText Markup Language og stadig ofte kaldt HTML5, er det rene tekstformat, enhver webside kommer frem i: indhold pakket ind i tags, der fortæller, hvad hver del er, så browseren kan skelne en overskrift fra et afsnit, et link fra et listepunkt og et formularfelt fra dets etiket.

Browseren læser teksten oppefra og ned og bygger et træ af elementer ud fra den, Document Object Model eller DOM. Det er dette træ, stylesheets formgiver, scripts ændrer, og skærmlæsere læser højt. Arbejdsdelingen på nettet er gammel og holder stadig: HTML siger, hvad noget er, CSS siger, hvordan det ser ud, og JavaScript siger, hvad det gør, når nogen rører ved det. Et element som `<nav>`, `<button>` eller `<h2>` bærer en betydning, som alle tre kan handle på. Et ægte `<button>`-element får tastaturfokus, en rolle, som hjælpemidler annoncerer, og aktivering med Enter eller mellemrumstasten, uden en eneste ekstra kodelinje.

Sproget begyndte som en måde at beskrive videnskabelige dokumenter semantisk på, og det voksede til laget under næsten alle programmer med en webgrænseflade. De første fem år, 1990–1995, gik på CERN og derefter i IETF; HTML 3.2 blev gjort færdig i W3C i 1997, og HTML4 fulgte kort efter. I årevis passede W3C og WHATWG hver deres tekst, og teksterne gled fra hinanden, indtil de to i 2019 indgik en aftale om at arbejde på én: HTML Living Standard, som WHATWG vedligeholder, og som alle større browsermotorer implementerer. Den har intet versionsnummer. På spørgsmålet »er det her HTML5?« svarer standarden kort ja og kalder ordet et modeord for den moderne webplatform frem for navnet på en udgivelse.

Hvor står det i forhold til det, det oftest sammenlignes med? Markdown er en genvejsnotation, som et byggeværktøj omsætter til HTML. PDF låser layoutet til en trykt side, hvor HTML flyder om til den skærm, det lander på. XHTML er det samme ordforråd skrevet under XML's strengere regler. JSX-, Jinja- og Hugo-skabeloner er måder at producere HTML på, og det, der når browseren fra dem alle, er stadig HTML. Et team, der vejer et håndbygget site op mod en hjemmesidebygger, vælger, hvem der skriver markuppen, aldrig om der er nogen.

Brug begrebet i en beslutning

Brug »Hvad er HTML?« 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

Få teams vælger HTML med vilje. Det kommer som det, deres framework, sidebygger eller indholdssystem tilfældigvis spytter ud. Det valg, de i virkeligheden træffer, er, om det output er en leverance, nogen efterser, eller et biprodukt, ingen læser, og konsekvenserne lander tre steder.

Først: hvem der kan bruge siden. Hjælpemidler læser træet, ikke designet. En skærmlæserbruger bevæger sig gennem en side via overskrifter og landmærker; en tastaturbruger tabber fra det ene link eller kontrolelement til det næste. En stylet `<div>` med en klikhandler ligner en knap for en seende musebruger og er almindelig tekst for de to andre. Tilgængelighedskrav som WCAG er i høj grad skrevet op mod det, markuppen udstiller, så at opfylde dem efter lanceringen betyder som regel at omskrive skabeloner frem for at justere farver.

Dernæst: hvor længe indholdet holder. Browsere åbner stadig sider skrevet i 1990'erne, fordi bagudkompatibilitet er et af de principper, standarden vedligeholdes efter. Frameworks skiftes ud med få års mellemrum. Indhold i ren, gyldig HTML kan flyttes til et andet system, arkiveres, konverteres til andre formater eller få nyt design uden at blive skrevet om. Indhold begravet i en sidebyggers indlejrede wrappere skal som regel kopieres ud i hånden.

Til sidst: hastighed og rækkevidde. En side, serveren sender færdig, kan tegnes, mens bytes ankommer. En applikation, der sender en næsten indholdsløs skal og bygger dokumentet i JavaScript, lader enhver læser vente på scriptet, og alt, der læser siden uden at køre det, fra en linkforhåndsvisning til mange crawlere, ser kun skallen. For et produkt bag et login, hvor interaktionen er pointen, kan den byttehandel være det værd. For markedsføringssider, dokumentation og alt, der skal findes via søgning, er færdig HTML det billigere udgangspunkt.

Prisen for alle tre er den samme og lille: én på holdet, der læser HTML lige så omhyggeligt som kode. Semantisk markup koster næsten intet, når den skrives først, og rigtig meget, når den eftermonteres.

Sådan ser det ud i praksis

Dette site er det mest ligetil eksempel. Dets kolofon viser, hvad der når en besøgendes browser: HTML, ét stylesheet, lokale webfonte, billeder, men ingen JavaScript til afvikling i browseren. Hver side skrives færdig som en fil på disken, før nogen beder om den, i stedet for at blive samlet fra en skabelon ved hver forespørgsel. Sitets kvalitetsport validerer HTML'en med vnu og tæller hver meddelelse som en fejl, og samme side linker til W3C's markup-validator, som kører den samme motor mod den levende standard, så en læser selv kan efterprøve påstanden.

Tidligere kundearbejde kom til HTML fra den anden side. På tværs af tredive websites bygget ud fra færdige visuelle designs blev Photoshop-filerne omsat til HTML i hånden. Standarden havde to sider: trofast mod de mellemrum, den typografi og de detaljer, designeren havde tænkt sig, og fleksibel nok til, at et site blev ved med at virke, første gang nogen redigerede indholdet. At ramme mockuppen er den ene halvdel af opgaven; markup, der overlever næste redigering, er den anden.

At holde HTML som den kanoniske form giver også plads til at udgive andre formater ved siden af. Da sitet blev spejlet til Gemini og Gopher, lavede det samme build 777 Gemini-dokumenter og 777 Gopher-menuer ud fra samme kilde i den dokumenterede implementeringsfase. En separat sammenligning af builds målte nul ændrede HTML-bytes, da de formater blev tilføjet; resultatet beskriver ikke senere redaktionelle eller visuelle ændringer.

Hvor det går galt

Layoutet forskubber sig på måder, intet stylesheet forklarer. En side, der begynder uden `<!DOCTYPE html>`, eller med en gammel og fejlformet en af slagsen, sætter browseren i quirks mode, en gengivelsestilstand, der efterligner gamle browsere. Procenthøjder, tabelceller og linjehøjder følger så gammel adfærd, der afviger fra CSS-specifikationerne, og resultatet adskiller sig fra samme side i standardtilstand. Rettelsen er én linje øverst i hver skabelon.

Fejlbehæftet markup bliver vist, og træet bliver et andet. HTML-parseren kommer sig over enhver fejl på en fastlagt måde, ens i alle browsere, så fejl sjældent ses på skærmen. De dukker op senere, når en selektor eller et script leder efter et element et andet sted, end parseren anbragte det. Et blokelement inde i et afsnit er det klassiske tilfælde:

```
<p>Første linje
  <div>En blok placeret inde i den</div>
</p>
```

Parseren lukker afsnittet før `<div>`, og det overskydende `</p>` bliver til et andet afsnit uden indhold. Kør en validator i buildet, og lad dens fejl stoppe den; browserens tolerance er en genopretningsmekanisme, ikke en gennemgang.

Alle kontrolelementer bliver til en `div`. Klikbare `div`- og `span`-elementer, overskrifter valgt efter størrelse, sænket skrift brugt til at gøre tekst mindre: hver af dem ser rigtig ud og fortæller hjælpemidlerne noget forkert. Tastaturbrugere passerer kontrolelementer, der aldrig får fokus, og skærmlæserbrugere mister de overskrifter og landmærker, de navigerer efter. En side uden et `<main>`-element eller en `<h1>` lader dem læse fra toppen hver gang. Brug det element, der navngiver opgaven, som `button`, `a`, `h2`, `main` eller `nav`, og lad CSS ændre udseendet.

Siden står tom, indtil scriptet har kørt. Et klientrenderet site sender ikke meget mere end `<div id="app"></div>` og bygger dokumentet i browseren. På en langsom telefon venter læseren; fejler scriptet, eller blokerer et filter det, bliver siden ved med at stå tom. Generér dokumentet på serveren eller under buildet, og læg interaktion oven på HTML, der allerede giver mening i sig selv.

Læs mere

- HTML Living Standard — den specifikation, browserne implementerer, vedligeholdt af WHATWG og opdateret løbende.
- Introduction — historien fra CERN til aftalen i 2019 og standardens eget svar på HTML5-spørgsmålet.
- Parsing HTML documents — den fejlhåndtering, alle browsere deler, skrevet ud regel for regel.
- The HTML syntax — doctype, elementer og attributter, og hvad der gør et dokument gyldigt.
- The Nu Html Checker — motoren bag vnu og W3C's markup-validator, med brug fra kommandolinjen.
- DOM Standard — det træ, parseren bygger, og de grænseflader, scripts bruger til at ændre det.

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 HTML?« 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/html/