Engineer

HTML

språket en sides struktur skrives i

Hva det er

HTML, kort for HyperText Markup Language og fortsatt ofte kalt HTML5, er ren tekst med koder rundt innholdet, og det er i dette formatet hver eneste nettside kommer fram. Kodene, eller taggene, sier hva hver bit er, slik at nettleseren kan skille en overskrift fra et avsnitt, en lenke fra et listepunkt og et skjemafelt fra etiketten sin.

En nettleser leser teksten ovenfra og ned og bygger et tre av elementer fra den, Document Object Model, eller DOM. Det treet er det stilark gir utseende, det skript endrer, og det skjermlesere leser opp. Arbeidsdelingen på nettet er gammel og holder fortsatt: HTML sier hva noe er, CSS sier hvordan det ser ut, og JavaScript sier hva det gjør når noen tar på det. Et element som `<nav>`, `<button>` eller `<h2>` bærer en betydning som alle disse leserne kan handle på. En ekte `<button>` får tastaturfokus, en rolle som hjelpemidler leser opp, og aktivering med Enter eller mellomrom, alt uten en eneste ekstra kodelinje.

Språket begynte som en måte å beskrive vitenskapelige dokumenter semantisk på, og vokste til laget under nesten hver applikasjon med et nettgrensesnitt. De første fem årene, 1990–1995, ble tilbrakt ved CERN og deretter i IETF. HTML 3.2 ble ferdigstilt i W3C i 1997, og HTML4 kom kort etter. I årevis vedlikeholdt W3C og WHATWG tekster som gled fra hverandre, til de i 2019 inngikk en avtale om å arbeide på én: HTML Living Standard, som WHATWG vedlikeholder og alle de store nettlesermotorene implementerer. Den har ikke noe versjonsnummer. Spør du «er dette HTML5?», svarer standarden med et kort ja og kaller begrepet et moteord for den moderne webplattformen, ikke en utgivelse.

Hvor står det ved siden av det det sammenlignes med? Markdown er en kortform som et byggeverktøy gjør om til HTML. PDF låser oppsettet til en trykt side, mens HTML flyter om til den skjermen den havner på. XHTML er det samme

vokabularet skrevet etter de strengere reglene i XML. Maler i JSX, Jinja og Hugo er måter å produsere HTML på, og det som når nettleseren fra alle sammen, er fortsatt HTML. Et team som veier et håndbygd nettsted mot en nettstedsbygger, velger hvem som skriver koden, aldri om det finnes noen kode.

Bruk begrepet i en beslutning

Bruk «Hva er HTML?» 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

Få team velger HTML med vilje. Den kommer som det rammeverket, sidebyggeren eller publiseringssystemet tilfeldigvis spytter ut. Valget de egentlig tar, er om dette resultatet er et leveranseobjekt noen kontrollerer, eller et biprodukt ingen leser, og konsekvensene lander tre steder.

Først kommer hvem som kan bruke siden. Hjelpemidler leser treet, ikke designet. En som bruker skjermleser, beveger seg gjennom en side etter overskriftene og landemerkene; en tastaturbruker tabber fra én lenke eller kontroll til den neste. En stylet `<div>` med en klikkhåndterer ser ut som en knapp for en seende musebruker og er ren tekst for begge de andre. Krav til universell utforming, som WCAG 2.2, er i stor grad skrevet mot det markupen eksponerer, så å oppfylle dem etter lansering betyr gjerne å skrive om maler, ikke å justere farger.

Deretter kommer hvor lenge innholdet varer. Nettlesere åpner fortsatt sider skrevet på 1990-tallet, fordi bakoverkompatibilitet er et av prinsippene standarden vedlikeholdes etter. Rammeverk skiftes ut med noen års mellomrom. Innhold som ligger i ren, gyldig HTML, kan flyttes til et annet system, arkiveres, konverteres til andre formater eller få et nytt design uten å skrives om. Innhold som er begravd i en sidebyggers nøstede innpakninger, må som regel kopieres ut for hånd.

Til sist kommer fart og rekkevidde. En side serveren sender komplett, kan tegnes etter hvert som bytene kommer. En applikasjon som sender et nesten tomt skall og bygger dokumentet i JavaScript, får hver leser til å vente på skriptet, og alt

som leser siden uten å kjøre det, fra en lenkeforhåndsvisning til mange søkeroboter, ser bare skallet. For et produkt bak en innlogging, der samhandlingen er hele poenget, kan den avveiningen lønne seg. For markedsføringssider, dokumentasjon og alt som skal finnes gjennom søk, er komplett HTML det billigste utgangspunktet.

Prisen for alle tre er den samme og liten: én på teamet som leser HTML like nøye som de leser kode. Semantisk markup koster nesten ingenting når den skrives først, og svært mye når den ettermonteres.

Slik ser det ut i praksis

Dette nettstedet er det enkleste eksemplet å vise til. Krediteringssiden lister hva som når en besøkendes nettleser: HTML, ett stilark, lokale webfonter, bilder, uten kjørbar JavaScript. Hver side skrives ut i sin helhet som en fil på disk før noen ber om den, i stedet for å settes sammen fra en mal ved hver forespørsel. Kvalitetsporten på nettstedet validerer HTML-en med vnu og behandler hver melding som en feil, og den samme siden lenker til W3C Markup Validator, som kjører den samme motoren mot den levende standarden, slik at en leser kan kontrollere påstanden i stedet for å stole på den.

Tidligere kundearbeid nærmet seg HTML fra den andre siden. På tretti nettsteder bygd fra ferdige visuelle design ble Photoshop-filene gjort om til HTML for hånd. Standarden hadde to sider: tro mot avstandene, typografien og detaljene designeren hadde tenkt seg, og fleksibel nok til at nettstedet fortsatt virket første gang noen redigerte innholdet. Å treffe skissen er halve jobben; markup som overlever neste redigering, er den andre halvdelen.

Å beholde HTML som den kanoniske formen gir også plass til å publisere andre formater ved siden av. Da nettstedet ble speilet til Gemini og Gopher, produserte det samme bygget 777 Gemini-dokumenter og 777 Gopher-menyer fra samme kilde på det dokumenterte implementeringstidspunktet. En separat sammenligning av bygg målte null endrede HTML-byte da disse formatene ble lagt til; dette resultatet beskriver ikke senere redaksjonelle eller visuelle endringer.

Hvor det går galt

Oppsettet forskyver seg på måter intet stilark forklarer. En side som begynner uten `<!DOCTYPE html>`, eller med en gammel og feilskrevet doctype, setter nettleseren i quirks mode, en gjengivelsesmodus som etterligner eldre nettlesere. Prosenthøyder, tabellceller og linjehøyder følger da gammel oppførsel som avviker fra CSS-spesifikasjonene, og resultatet blir et annet enn den samme siden i standardmodus. Løsningen er én linje øverst i hver mal.

Ødelagt markup blir vist, og treet blir et annet. HTML-parseren kommer seg etter hver eneste feil på en definert måte, likt i alle nettlesere, så feil synes sjelden på skjermen. De dukker opp senere, når en selektor eller et skript leter etter et element et annet sted enn der parseren la det. Et blokkelement inne i et avsnitt er skoleeksemplet:

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

Parseren lukker avsnittet før `<div>`, og gjør så den løse `</p>` om til et nytt, tomt avsnitt. Kjør en validator i bygget og la bygget feile på meldingene den gir; nettleserens toleranse er en mekanisme for å komme seg etter feil, og den gjennomgår ingenting.

Hver kontroll blir til en `div`. Klikkbare `div`- og `span`-elementer, overskrifter valgt etter størrelse, senket skrift brukt for å gjøre tekst mindre: hver av dem ser riktig ut og forteller hjelpemidlene noe feil. Tastaturbrukere passerer kontroller som aldri får fokus, og skjermleserbrukere mister overskriftene og landemerkene de navigerer etter. En side uten et `<main>`-element eller en `<h1>` lar dem lese fra toppen hver gang. Bruk elementet som navngir jobben, som `button`, `a`, `h2`, `main` eller `nav`, og la CSS endre utseendet.

Siden står tom til skriptet har kjørt. Et nettsted som gjengis hos klienten, sender lite mer enn `<div id="app"></div>` og bygger dokumentet i nettleseren. På en treg telefon venter leseren; når skriptet feiler eller et filter blokkerer det, blir siden stående tom for godt. Gjengi dokumentet på serveren eller ved byggetid, og legg samhandling oppå HTML som allerede gir mening alene.

Les mer

- HTML Living Standard — spesifikasjonen nettleserne implementerer, vedlikeholdt av WHATWG og oppdatert fortløpende.
- Introduction to the HTML standard — historien fra CERN til avtalen i 2019, og standardens eget svar på spørsmålet om HTML5.
- Parsing HTML documents — feilrettingen alle nettlesere deler, skrevet ut regel for regel.
- The HTML syntax — doctype, elementer og attributter, og hva som gjør et dokument gyldig etter standarden.
- The Nu Html Checker — motoren bak vnu og W3Cs markupvalidator, med bruk fra kommandolinjen.
- DOM Standard — treet parseren bygger, og grensesnittene skript bruker for å endre det.

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 HTML?» 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/html/