Engineer

Skærmlæser

software der læser en side op eller til braille

Hvad det er

En skærmlæser er et program, der gengiver alt på en computer eller telefon, fra teksten i et dokument til knapperne omkring den og den tilstand, hver af dem står i, som syntetisk tale eller som en linje på et punktdisplay, og som lader brugeren betjene grænsefladen fra tastaturet eller med berøringsbevægelser. Ordet dækker en hel familie af skærmlæsere og ikke ét produkt: NVDA, der er gratis og open source, og JAWS på Windows; VoiceOver, der er bygget ind i macOS og Apples mobile systemer; TalkBack på Android. Brugerne er for det meste blinde eller svagsynede, og nogle, der sagtens kan se skærmen, bruger en alligevel, fordi det er lettere for dem at lytte end at læse.

Det, de fleste udviklere overser, er, hvor ordene kommer fra. Skærmlæseren kigger ikke på pixels. Browseren omsætter HTML'en til en anden struktur, tilgængelighedstræet, hvor hvert element er skåret ned til en rolle (knap, link, overskrift, liste), et navn (de ord, der identificerer det) og en tilstand (markeret, udfoldet, deaktiveret). Styresystemets tilgængelighedsgrænseflade giver det træ videre til skærmlæseren, som læser det op. En knap, hvis synlige indhold er en tegning af et papirfly, og hvis kode ikke indeholder tekst, er i det træ en knap uden navn, og det er præcis, hvad brugeren får at høre.

Brugerne lytter heller ikke til en side fra top til bund. Med NVDA's standardtaster bliver enkelte bogstaver i gennemsynstilstand til navigationstaster: `H` springer til næste overskrift, `D` til næste landmærke, `K` til næste link, `F` til næste formularfelt, og tallene 1 til 9 til en overskrift på det niveau. `NVDA+F7` åbner en liste over alle links, overskrifter eller landmærker på siden. VoiceOver har rotoren, som åbnes med `VO-U` og giver de samme lister. En seende læser skimmer ved at lade blikket falde på fed skrift og luft; en skærmlæserbruger skimmer ved at springe mellem strukturelle elementer, og en side uden den struktur giver ingen steder at springe hen.

Brug begrebet i en beslutning

Brug »Hvad er en skærmlæser?« 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

Den beslutning, et team reelt står over for, er, om den grænseflade, det sender ud, virker for en person, der kun modtager den, sådan som markup'en beskriver den. Skærmlæseren er den mest bogstavelige modtager, en side har. Layout, farve, placering, et ikon som alle genkender, et felt der bliver rødt: hvert signal, der kun bor i det visuelle design, går tabt, og kun det, koden siger, når frem.

Derfor er den den ene test af markup'en, som ingen mængde kiggen på siden kan erstatte. Et designreview, en sammenligning af skærmbilleder og en omhyggelig gennemklikning foregår alle i den visuelle kanal, og det er netop dér, de fejl, skærmlæseren afslører, per definition er usynlige. Automatiske regelmotorer hjælper, og de fanger rigtige problemer i bunker: et billede uden alternativ tekst, et formularfelt uden etiket, en attributværdi, der ikke giver mening. Det, de tjekker, er, at et navn findes, ikke at det er det rigtige. At høre en betalingsknap blive annonceret som »knap« eller en menu som »hamburger« kræver et menneske og et par minutter.

Det menneske og de minutter koster mindre, end de fleste teams regner med. NVDA er gratis at installere, og VoiceOver ligger allerede på hver Mac og iPhone på kontoret, så den første test kræver hverken indkøb eller specialist. Den kræver en, der lukker øjnene eller slukker skærmen og prøver at gennemføre sidens hovedopgave med tastaturet og ørerne. Rettelsen af det, de finder, er som regel billigst, når den sker tidligt: et ægte `button` - eller `a` -element har sin rolle og sin tastaturadfærd med fra start, mens en klikbar `div` skal have det hele bygget og vedligeholdt i hånden.

Afvejningen er, at skærmlæsere er forskellige. Hver af dem tolker tilgængelighedstræet på sin egen måde, hver browser bygger træet på sin egen måde, og en side, der er testet i én kombination, kan lyde anderledes i en anden. To almindelige kombinationer dækker mere end én, der er testet til bunds.

Sådan ser det ud i praksis

I én app kunne fejlen høres direkte. Grænsefladen tegnede sine kontroller med systemsymboler, og et symbol uden tilgængelighedsetiket læses op med sit eget interne navn, så send-knappen blev annonceret som »arrow up circle, button«. Sytten kontroller med kun et ikon fik navn efter det, de gør, med en etiket, der navngiver handlingen frem for formen, og som følger kontrollens tilstand, hvor betydningen skifter med den. En linter kræver nu en tilgængelighedsetiket inden for otte linjer fra enhver kontrol med kun et ikon, en bevidst grov afstand, der er valgt ved måling, så en attende ikke kan tilføjes uden. Siden er ærlig om grænsen: reglen beviser, at en etiket findes i nærheden, og om den er korrekt, kræver stadig, at nogen lytter efter.

Dokumenter rejser det samme spørgsmål i et andet format. En PDF, der ser rigtig ud på skærmen, siger intet om, hvorvidt en skærmlæser kan læse den, så ni genererede dokumenter, CV'et i sine varianter, referencearket, porteføljen og ansøgningen på tværs af sprog, blev kørt gennem en uafhængig validator for PDF/UA-tilgængelighedsprofilen på det dokumenterede valideringstrin og bestod alle 106 konfigurerede regler i tilgængelighedsprofilen. For at bestå krævede det tagget struktur, erklæret dokumentsprog, alternativ tekst på hver grafik, der ikke er pynt, en rigtig titel i metadataene og en eksplicit læserækkefølge i stedet for den, layoutet tilfældigvis giver. Begge tal kommer fra én validator, og det står der også.

Sitet her behandler sine egne automatiske tjek på samme måde. Kolofonen kalder dem gulvet og ikke beviset og beder alle, der bruger en skærmlæser her og hører noget blive læst forkert op, om at sige til, fordi det er den ene rapport, tjekket ikke selv kan frembringe.

Hvor det går galt

De fleste skærmlæserfejl er usynlige på skærmen, og derfor overlever de et review. De tager som regel en af få former.

Kontrollen annoncerer sin tegning. En ikonknap uden tekst får navn fra det, koden nu tilbyder: et filnavn, et symbolnavn eller ingenting, så der kun står »knap« tilbage. Giv hver kontrol med kun et ikon et navn, der beskriver handlingen, og skjul selve tegningen fra træet:

```
<button type="button" aria-label="Send besked">
    <svg aria-hidden="true" focusable="false"><!-- ikon --></svg>
</button>
```

Siden læses som én lang strøm. Overskrifter, der i virkeligheden er stylede `div`-elementer, og et layout uden landmærker ser fine ud og efterlader `H` og `D` uden noget at lande på, så lytteren må pile sig gennem hver linje for at finde det, de kom efter. Brug rigtige `h1` til `h6` i rækkefølge, og læg sidens områder i `header`, `nav`, `main` og `footer`.

Noget ændrer sig, og lytteren hører stilhed. En fejlmeddelelse dukker op under et felt, et tal i kurven stiger, en kort besked glider ind og ud. Seende brugere ser det; skærmlæseren siger intet, fordi dens fokus er et andet sted. Flyt fokus til meddelelsen, eller læg dynamiske opdateringer i et område markeret med `aria-live`, så skærmlæseren annoncerer dem.

Skjult indhold læses op, eller synligt indhold forstummer. Tekst, der er skubbet uden for skærmen med CSS, ligger stadig i tilgængelighedstræet og bliver læst op; `aria-hidden="true"` på et synligt element fjerner det for lytteren, mens alle andre stadig ser det. Begge dele er nyttige med vilje og forvirrende ved et uheld, så tjek, hvad træet indeholder, og ikke kun hvad skærmen viser.

Alle tjek består, og siden lyder stadig forkert. En etiket kan findes og være forkert, en alternativ tekst kan beskrive filen i stedet for pointen, og en læserækkefølge kan være gyldig og absurd. Sæt en kort lyttegennemgang på udgivelseslisten for hver side, der har ændret struktur.

Læs mere

- Tools and Techniques — W3C's overblik over, hvordan mennesker med handicap bruger nettet, skærmlæsere blandt andre hjælpemidler.
- NVDA User Guide — gennemsynstilstand, navigation med enkelte bogstaver og elementlisten, fra udviklerne af en gratis skærmlæser.
- Use the VoiceOver rotor on Mac — Apples vejledning i at springe efter overskrift, link og landmærke.
- Core Accessibility API Mappings 1.2 — hvordan roller, navne og tilstande går fra browseren til den platformsgrænseflade, en skærmlæser læser.

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 en skærmlæser?« 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/screen-reader/