Engineer

Skjermleser

programvare som leser en side høyt, eller til punktskrift

Hva det er

En skjermleser er et program som gjør alt på en datamaskin eller en mobil om til syntetisk tale eller til en linje på en leselist for punktskrift: teksten i et dokument, knappene rundt den og tilstanden hver av dem er i. Brukeren styrer grensesnittet fra tastaturet eller med berøringsbevegelser. Ordet betegner en hel familie av programmer, ikke ett produkt: NVDA, som er gratis og har åpen kildekode, og JAWS på Windows; VoiceOver, som er bygd inn i macOS og Apples mobile systemer; og TalkBack på Android. De fleste som bruker en skjermleser, er blinde eller svaksynte, og noen som ser skjermen godt, bruker en fordi det er lettere for dem å lytte enn å lese.

Det de fleste utviklere overser, er hvor ordene kommer fra. Skjermleseren ser ikke på pikslene. Nettleseren gjør HTML-koden om til en egen struktur, tilgjengelighetstreet, der hvert element er redusert til en rolle (knapp, lenke, overskrift, liste), et navn (ordene som identifiserer det) og en tilstand (avkrysset, utvidet, deaktivert). Operativsystemets grensesnitt for tilgjengelighet gir treet videre til skjermleseren, som leser det opp. En skjermknapp der det synlige innholdet er en tegning av et papirfly og koden ikke har noen tekst, er i treet en knapp uten navn, og det er akkurat det lytteren får høre.

Brukerne lytter heller ikke til en side fra topp til bunn. Når NVDA leser en nettside i browse mode, blir enkeltbokstaver til navigasjonstaster: `H` hopper til neste overskrift, `D` til neste landemerke, `K` til neste lenke, `F` til neste skjemafelt, og sifrene 1 til 9 til en overskrift på det nivået. `NVDA+F7` åpner en liste over alle lenker, overskrifter eller landemerker på siden. VoiceOver har rotoren, som åpnes med `VO-U` og gir de samme listene. En seende leser skumleser ved å se etter fet skrift og luft på siden. Den som bruker skjermleser, skumleser ved å hoppe mellom strukturelementene, og en side uten den strukturen gir ingenting å hoppe mellom.

Bruk begrepet i en beslutning

Bruk «Hva er en skjermleser?» 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

Spørsmålet et team faktisk står overfor, er om grensesnittet det leverer, virker for en som bare får det slik koden beskriver det. En skjermleser er den mest bokstavelige mottakeren en side har. Layout, farge, plassering, et ikon alle kjenner igjen, et felt som blir rødt: hvert signal som bare finnes i det visuelle uttrykket, går tapt, og bare det koden sier, når fram til den som lytter.

Derfor er lytting den ene testen av oppmerkingen som ingen mengde titting på siden kan erstatte. En designgjennomgang, en sammenligning av skjermbilder og en nøye gjennomklikking skjer alle i den visuelle kanalen, og det er nettopp der feilene en skjermleser avslører, per definisjon er usynlige. Automatiske regelmotorer hjelper, og de fanger ekte problemer i store mengder: et bilde uten alternativ tekst, en skjemakontroll uten etikett, et attributt med en umulig verdi. Det de sjekker, er at et navn finnes, ikke at det er det riktige navnet. Å høre en betalingsknapp lest opp som «knapp», eller en meny som «hamburger», krever et menneske og noen få minutter.

Det mennesket og de minuttene koster mindre enn de fleste team tror. NVDA er gratis å installere, og VoiceOver ligger allerede på hver Mac og iPhone på kontoret, så den første testen krever verken innkjøp eller spesialist. Den krever at noen lukker øynene, eller slår av skjermen, og prøver å fullføre sidens hovedoppgave med tastaturet og ørene. Rettingen av det de finner, er som regel enda billigere når den kommer tidlig: et innebygd `button` - eller `a` -element tar med seg rolle og tastaturoppførsel, mens en klikkbar `div` må få alt dette bygd og vedlikeholdt for hånd.

Avveiningen er at skjermlesere er forskjellige. Hver av dem tolker tilgjengelighetstreet på sin måte, hver nettleser bygger treet på sin måte, og en side som er testet i én kombinasjon, kan leses annerledes i en annen. To vanlige

kombinasjoner av skjermleser og nettleser dekker mer enn én kombinasjon testet til bunns.

Slik ser det ut i praksis

I én app kunne feilen rett og slett høres. Grensesnittet tegnet kontrollene sine med systemsymboler, og et symbol uten tilgjengelighetsetikett leses opp med sitt eget interne navn, så sendeknappen presenterte seg som «arrow up circle, button». Alle de sytten kontrollene med bare ikon fikk navn etter hva de gjør: hver etikett navngir handlingen i stedet for formen, og der betydningen endrer seg med tilstanden, følger etiketten med. En linter krever nå en tilgjengelighetsetikett innen åtte linjer fra hver kontroll med bare ikon. Det er en bevisst grov avstand, valgt ved måling, og den gjør at en attende kontroll ikke kan legges til uten etikett. Siden er tydelig på grensen: regelen beviser at det finnes en etikett i nærheten, og om etiketten er riktig, krever fortsatt at noen lytter på den.

Dokumenter reiser det samme spørsmålet i et annet format. En PDF som ser riktig ut på skjermen, sier ingenting om hvorvidt en skjermleser kan lese den. Derfor ble ni genererte dokumenter, CV-en i variantene sine, referanselisten, porteføljen og søknadsbrevet på tvers av språk, kjørt gjennom en uavhengig validator for PDF/UA-profilen, profilen for universelt utformet PDF, og de besto på det dokumenterte valideringstrinnet alle 106 konfigurerte regler i tilgjengelighetsprofilen. For å bestå trengtes tagget struktur, deklarert dokumentspråk, alternativ tekst på hver grafikk som ikke er dekorasjon, en reell tittel i metadataene og en eksplisitt leserekkefølge i stedet for den layouten tilfeldigvis gir. Tallene kommer fra én validator, og siden sier det rett ut.

Nettstedet behandler sine egne automatiske revisjoner på samme måte. Siden om teknologien bak nettstedet kaller disse revisjonene et minimum og ikke et bevis. Den ber alle som bruker skjermleser der og hører noe bli lest feil, om å si fra, fordi det er den ene rapporten kvalitetsporten ikke har noen måte å generere på.

Hvor det går galt

Det en skjermleser avslører, er som regel usynlig på skjermen, og derfor overlever det gjennomgangen. Feilene har noen få gjenkjennelige former.

Kontrollen som leser opp tegningen sin. En ikonknapp uten tekst får navnet sitt fra det koden tilfeldigvis gir: et filnavn, et symbolnavn eller ingenting, og da står bare «knapp» igjen. Gi hver kontroll som bare består av et ikon, et navn som beskriver handlingen, og skjul selve tegningen fra treet:

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

En side som leses som én lang strøm. Overskrifter som er stylede `div`-elementer, og en layout uten landemerker ser fine ut, men gir `H` og `D` ingenting å lande på. Den som lytter, må da bla seg gjennom hver linje for å finne det den kom for. Bruk ekte `h1` til `h6` i riktig rekkefølge, og legg sidens områder i `header`, `nav`, `main` og `footer`.

En endring som møtes med stillhet. En feilmelding dukker opp under et felt, tallet i handlekurven går opp, et varsel glir inn og ut igjen. Seende brukere ser det, mens skjermleseren tier, fordi fokuset er et annet sted. Flytt fokus til meldingen, eller legg dynamiske oppdateringer i et område merket med `aria-live`, så skjermleseren leser dem opp.

Skjult innhold som leses, og synlig innhold som forstummer. Tekst som er skjøvet ut av skjermen med CSS, ligger fortsatt i tilgjengelighetstreet og blir lest opp. `aria-hidden="true"` på et synlig element fjerner det for lytteren mens alle andre fortsatt ser det. Begge deler er nyttige med vilje og forvirrende ved et uhell, så sjekk hva treet inneholder, ikke bare hva skjermen viser.

Grønne kontroller og en side som fortsatt leses dårlig. En etikett kan finnes og være feil, en alternativ tekst kan beskrive filen og ikke poenget, og en leserekkefølge kan være gyldig og likevel absurd. Legg en kort lyttetest inn i sjekklisten før lansering for hver side som har fått endret strukturen sin.

Les mer

- Tools and Techniques — W3Cs oversikt over hvordan folk med funksjonsnedsettelser bruker nettet, med skjermlesere blant de andre hjelpemidlene.
- NVDA User Guide — browse mode, navigasjon med enkeltbokstaver og elementlisten, fra dem som lager en gratis skjermleser.

- Use the VoiceOver rotor on Mac — Apples veiledning i å hoppe mellom overskrifter, lenker og landemerker.
- Core Accessibility API Mappings 1.2 — hvordan roller, navn og tilstander går fra nettleseren til plattformgrensesnittet en skjermleser leser.

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 en skjermleser?» 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/screen-reader/