Engineer

Skärmläsare

programvara som läser en sida högt eller i punktskrift

Vad det är

En skärmläsare presenterar allt på en dator eller telefon, från texten i ett dokument till knapparna runt den och vilket tillstånd var och en befinner sig i, som syntetiskt tal eller som en rad på en punktskriftsdisplay, och låter användaren styra gränssnittet från tangentbordet eller med pekgester. Ordet, på engelska screen reader, betecknar en hel familj av program snarare än en enda produkt: NVDA, som är gratis och har öppen källkod, och JAWS på Windows; VoiceOver, inbyggt i macOS och i de mobila systemen från Apple; TalkBack på Android. Användarna är oftast blinda eller har nedsatt syn, och en del som kan se skärmen använder ändå en skärmläsare, eftersom det är lättare för dem att lyssna än att läsa.

Det de flesta utvecklare missar är var orden kommer ifrån. En skärmläsare tittar inte på pixlarna. Webbläsaren gör om HTML-koden till en andra struktur, tillgänglighetsträdet, där varje element har reducerats till en roll (knapp, länk, rubrik, lista), ett namn (orden som identifierar det) och ett tillstånd (ikryssad, expanderad, inaktiverad). Operativsystemets tillgänglighetsgränssnitt lämnar trädet vidare till skärmläsaren, som läser upp det. En knapp vars synliga innehåll är en teckning av ett pappersflygplan och vars kod saknar text är i det trädet en knapp utan namn, och skärmläsaren säger precis det.

Användarna lyssnar inte heller igenom en sida uppifrån och ned. Bläddringsläget i NVDA gör enskilda bokstäver till navigeringstangenter: `H` hoppar till nästa rubrik, `D` till nästa landmärke, `K` till nästa länk, `F` till nästa formulärfält och siffrorna 1 till 9 till en rubrik på motsvarande nivå. `NVDA+F7` öppnar en lista över alla länkar, rubriker eller landmärken på sidan. VoiceOver har rotorn, som öppnas med `VO-U` och erbjuder samma listor. En seende läsare skummar genom att låta blicken fastna på fetstil och luft; den som använder skärmläsare skummar genom att hoppa mellan strukturella element, och en sida utan den strukturen har ingenting att hoppa mellan.

Använd begreppet i ett beslut

Använd ”Vad är en skärmläsare?” 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

Det beslut ett team faktiskt står inför är om gränssnittet det levererar fungerar för någon som bara tar emot det så som koden beskriver det. En skärmläsare är den mest bokstavliga läsare en sida har. Layout, färg, placering, en ikon som alla känner igen, ett fält som blir rött: varje signal som bara finns i den visuella designen går förlorad, och bara det koden säger når lyssnaren.

Därför är den det enda test av koden som inget tittande på sidan kan ersätta. En designgranskning, en jämförelse av skärmdumpar och en noggrann genomklickning sker alla i den visuella kanalen, och just där är de defekter som en skärmläsare avslöjar per definition osynliga. Automatiska regelmotorer hjälper, och de hittar verkliga problem i mängd: en bild utan alternativtext, en formulärkontroll utan etikett, ett attribut med ett omöjligt värde. Det de kontrollerar är att ett namn finns, inte att det är rätt namn. Att höra en kassaknapp presenteras som ”button” eller en meny som ”hamburger” kräver en människa och några minuter.

Den människan och de minuterna kostar mindre än de flesta team tror. NVDA kostar ingenting att installera, och VoiceOver finns redan på varje Mac och iPhone på kontoret, så det första testet kräver varken inköp eller specialist. Det kräver någon som blundar, eller stänger av skärmen, och försöker slutföra sidans huvuduppgift med tangentbordet och öronen. Det som då hittas blir oftast ännu billigare att rätta om det sker tidigt: ett inbyggt `button` - eller `a` -element har sin roll och sitt tangentbordsbeteende med sig, medan en klickbar `div` behöver allt det byggt och underhållet för hand.

Avvägningen är att skärmläsare skiljer sig åt. Var och en tolkar tillgänglighetsträdet på sitt eget sätt, varje webbläsare bygger trädet på sitt eget sätt, och en sida som har testats i en kombination kan låta annorlunda

i en annan. Att testa i två vanliga kombinationer täcker mer än att testa en enda uttömmande.

Så ser det ut i praktiken

I en applikation gick defekten att höra direkt. Gränssnittet ritade sina kontroller med systemsymboler, och en symbol utan tillgänglighetsetikett läses upp med sitt eget interna namn, så skicka-knappen presenterade sig som ”arrow up circle, button”. Sjutton kontroller med enbart ikon fick namn efter vad de gör: varje etikett namnger handlingen snarare än formen och följer kontrollens tillstånd där betydelsen skiftar med det. Ett lintverktyg kräver nu en tillgänglighetsetikett inom åtta rader från varje kontroll med enbart ikon, ett medvetet grovt avstånd som valdes genom mätning, så en artonde kan inte läggas till utan etikett. Sidan är rak om gränsen: regeln bevisar att det finns en etikett i närheten, och huruvida etiketten är korrekt kräver fortfarande att någon lyssnar på den.

Dokument väcker samma fråga i ett annat format. Att en PDF ser rätt ut på skärmen säger ingenting om huruvida en skärmläsare kan läsa den, så nio genererade dokument, CV:t i sina varianter, referensbladet, portföljen och det personliga brevet på flera språk, kördes genom ett oberoende valideringsverktyg för tillgänglighetsprofilen PDF/UA vid det dokumenterade valideringsskedet och klarade alla 106 konfigurerade regler i tillgänglighetsprofilen. För det krävdes taggad struktur, ett deklarerat dokumentspråk, alternativtext på varje grafik som inte är dekoration, en riktig titel i metadata och en explicit läsordning i stället för den som layouten råkar ge. Båda talen kommer från ett och samma valideringsverktyg, och sidan säger det öppet.

Den här webbplatsen behandlar sina egna automatiska granskningar på samma sätt. Dess kolofon kallar granskningarna golvet och inte beviset, och ber den som använder en skärmläsare här och hör något läsas upp fel att säga till, eftersom det är den enda rapport som kvalitetsgrinden inte har något sätt att ta fram.

Var det går fel

Det mesta som går snett för en skärmläsare är osynligt på skärmen, och därför klarar det sig igenom granskningen. Det brukar ta några få former.

Kontrollen läser upp sin grafik. En ikonknapp utan text får sitt namn från det koden råkar erbjuda: ett filnamn, ett symbolnamn eller ingenting, och då återstår bara ”button”. Ge varje kontroll med enbart ikon ett namn som beskriver

handlingen, och dölj själva teckningen för trädet:

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

Sidan läses som en enda lång ström. Rubriker som är stylade `div`-element och en layout utan landmärken ser bra ut men lämnar `H` och `D` utan något att landa på, så lyssnaren måste pila sig igenom rad för rad för att hitta den del som besöket gällde. Använd riktiga `h1` till `h6` i ordning, och lägg sidans regioner i `header`, `nav`, `main` och `footer`.

Något ändras och lyssnaren hör tystnad. Ett felmeddelande dyker upp under ett fält, antalet i varukorgen ökar, en notis glider in och ut. Seende användare ser det; skärmläsaren förblir tyst, eftersom dess fokus ligger någon annanstans. Flytta fokus till meddelandet, eller lägg dynamiska uppdateringar i en region märkt med `aria-live`, så att skärmläsaren läser upp dem.

Dold text läses upp, eller synlig text tystas. Text som har knuffats utanför skärmen med CSS finns kvar i tillgänglighetsträdet och läses upp; `aria-hidden="true"` på ett synligt element tar bort det för lyssnaren medan alla andra fortfarande ser det. Båda är användbara när de görs med avsikt och förvirrande när de sker av misstag, så kontrollera vad trädet innehåller, inte bara vad skärmen visar.

Varje kontroll går igenom och sidan låter ändå fel. En etikett kan finnas och vara fel, en alternativtext kan beskriva filen och inte poängen, och en läsordning kan vara giltig och ändå absurd. Lägg in en kort lyssningsrunda i checklistan inför release för varje sida vars struktur har ändrats.

Läs mer

- Tools and Techniques — en översikt från W3C över hur personer med funktionsnedsättning använder webben, med skärmläsare bland de andra hjälpmedlen.
- NVDA User Guide — bläddringsläget, navigering med enskilda bokstäver och elementlistan, från dem som utvecklar en gratis skärmläsare.
- Use the VoiceOver rotor on Mac — guiden från Apple till att hoppa mellan rubriker, länkar och landmärken.

- Core Accessibility API Mappings 1.2 — hur roller, namn och tillstånd förs från webbläsaren till det plattformsgränssnitt som skärmläsaren läser.

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 en skärmläsare?” 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/screen-reader/