Engineer

Skjermknapp

en kontroll en person trykker på for å få noe til å skje

Hva det er

En skjermknapp, i dagligtale bare en knapp og i eldre tekster en trykknapp, er den delen av et brukergrensesnitt som handler når den aktiveres: den sender en melding, lagrer et utkast, åpner en dialog eller slår av lyden i en samtale. På nettet er den som regel HTML-elementet `<button>`, og i universell utforming er det alt som har rollen `button`.

WAI-ARIA 1.2 definerer rollen som «an input that allows for user-triggered actions when clicked or pressed», altså en inndata som lar brukeren utløse handlinger ved klikk eller trykk, og legger til at knapper stort sett brukes til avgrensede handlinger. En vanlig kommandoknapp husker ingen tilstand. En av/på-knapp, merket med `aria-pressed`, blir stående på eller av til den trykkes igjen, og en menyknapp, merket med `aria-haspopup`, åpner en liste med flere valg.

Det viktigste skillet går mellom en knapp og en lenke. En lenke tar leseren et annet sted. En knapp endrer noe der leseren allerede er. ARIA Authoring Practices Guide (APG) kaller de to funksjonene klart forskjellige. Når en lenke er tegnet som en knapp eller omvendt, anbefaler veiledningen å endre det visuelle slik at det passer funksjonen, framfor å låne det andre elementets rolle.

HTML Living Standard beskriver `<button>` som en knapp som får etiketten sin fra innholdet. Attributtet `type` har tre nøkkelord. `submit` sender skjemaet knappen hører til, `reset` setter feltene i skjemaet tilbake til startverdiene, og `button` gjør ingenting på egen hånd og overlater handlingen til et skript. Når `type` mangler eller er feilstavet, står knappen i tilstanden Auto, og en knapp i den tilstanden inne i et skjema, uten noen kommando knyttet til seg, er en sendeknapp. Interaktivt innhold får ikke ligge inne i den: lenker, felt og andre knapper hører hjemme utenfor.

Standarden lar også en knapp virke på et annet element uten skript. `commandfor` peker ut målet med ID-en, og `command` sier hva som skal skje med det, f.eks. at en popover vises eller at en dialog åpnes som modal. Hvor mye av dette en bestemt nettleser har tatt i bruk, står i den nettleserens egne versjonsmerknader.

Bruk begrepet i en beslutning

Bruk «Hva er en skjermknapp?» 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

Knappen er stedet der lesing blir til handling. Kjøp, sending, lagring og sletting går alle gjennom en. En knapp som noen ikke finner, ikke når eller ikke forstår, stenger dem ute fra den delen av produktet som betaler for det.

Det ekte elementet kommer med oppførselen på plass. En `<button>` tar tastaturfokus, utfører handlingen sin på mellomrom og på Enter, forteller hjelpemiddelteknologi at den er en knapp og hva den heter, og slutter å reagere når den har `disabled`. Understanding-dokumentet til suksesskriterium 4.1.2 i WCAG 2.2, Name, Role, Value, påpeker at vanlige HTML-kontroller allerede oppfyller kriteriet når de brukes slik spesifikasjonen sier. En `<div>` med en klikkhåndterer har ingenting av dette, og hver bit som mangler, stenger noen ute.

Navnet er et løfte. En skjermleser leser opp en knapp med det tilgjengelige navnet og rollen, så navnet må si hva knappen gjør, og ikke hvordan den ser ut. APG har en regel for av/på-knapper som er lett å bryte: etiketten står fast mens tilstanden endrer seg. Eksempelet er en knapp med navnet «Mute», som leses opp som «Mute toggle button pressed» når den er på. Foretrekker et design en etikett som veksler mellom «Mute» og «Unmute», trengs ikke `aria-pressed`, fordi etiketten selv melder tilstanden.

Størrelse er et krav, ikke en smakssak. Suksesskriterium 2.5.8 i WCAG 2.2, Target Size (Minimum), på nivå AA, krever at et pekermål er minst 24 ganger 24 CSS-piksler. Unntakene gjelder små mål med nok luft rundt seg, en kontroll i full

størrelse et annet sted på siden som gjør samme jobb, lenker inne i en setning, kontroller der nettleseren bestemmer størrelsen, og størrelser som er avgjørende for funksjonen. Skjelvinger, en buss i bevegelse og en tommel på en telefon gjør alle små mål vanskeligere å treffe.

Fokus må lande et fornuftig sted. Etter et trykk venter APG at fokus flytter seg inn i en dialog som knappen åpnet, og at det som regel går tilbake til knappen når dialogen lukkes. Når trykket lar leseren bli i samme sammenheng, slik en Bruk-knapp gjør, blir fokus som regel stående på knappen. En tastaturbruker som får fokus kastet til toppen av siden etter hver lagring, begynner på siden på nytt hver gang.

Slik ser det ut i praksis

Det tydeligste eksempelet begynner med én opplesning. I en SwiftUI-app bygget med systemsymboler leste skjermleseren opp sendeknappen som «arrow up circle, button», fordi et symbol uten etikett leses opp med sitt interne navn. Arbeidet med å gi alle sytten rene ikonkontroller et navn for skjermlesere ga hver av dem en etikett som navngir handlingen i stedet for formen, og der betydningen avhenger av tilstand, følger etiketten tilstanden. Den mest varige delen er sjekken: en linter som ser etter konstruksjonen som lager en ren ikonkontroll, og krever en tilgjengelighetsetikett innenfor åtte linjer fra den. Åtte linjer ble valgt ved måling og er bevisst grovt. Regelen beviser at det finnes en etikett i nærheten, ikke at den er riktig, og noen må fortsatt lytte.

Et annet eksempel handler om hva en knapp ikke kan gjøre. I laget for betaling og tilgangsrettigheter, bygget på Stripe og på kjøp i appen hos Apple og Google, avgjør et eget databaseskjema for tilgang med 11 tabeller om en konto får bruke bedriftskatalogen, søket og dataeksporten. Serveren stiller det spørsmålet igjen for hver forespørsel, uansett hvilke knapper grensesnittet viste, og siden sier hvorfor i én linje: en skjult knapp er en høflighet og ikke en kontroll.

Knapper betyr noe også for lesere som ikke er mennesker. Siden om å være klar for KI-agenter peker på at mange agenter som styrer en nettleser, leser en side gjennom tilgjengelighetstreet, slik en skjermleser gjør. Ekte knapper og skjema med etiketter er en del av det som lar en agent finne skjemaet og fylle det ut.

Nettstedets egne maler inneholder ikke ett eneste `<button>` -element. Den fylte oppfordringen «Ta kontakt» er en `<a>` formet som en knapp, fordi den fører til kontaktsiden, og regelen i stilarket setter `min-height` til 24 piksler med en

kommentar som viser til suksesskriterium 2.5.8. Hovedmenyen er et `<details>`-element der `<summary>` åpner og lukker den, og kommentaren i malen sier at nettleseren eier åpen-tilstanden og knappesemantikken, uten skript.

Hvor det går galt

- **En klikkbar boks tar elementets plass.** En `<div>` eller `<span>` med en klikkhåndterer svarer på musen og lite annet: den tar ikke fokus, overser mellomrom og Enter og leses opp som vanlig tekst. Legger du til `role="button"`, rettes bare opplesningen. Fokus og taster må fortsatt skrives for hånd.
- **En lenke gjør jobben til en knapp, eller omvendt.** En lenke som sletter en post, kan åpnes i en ny fane eller lagres som bokmerke, og da gjentas slettingen eller går tapt av syne. En knapp som bare navigerer, lar seg ikke åpne i en ny fane og har ingen adresse å kopiere.
- **En knapp uten type inne i et skjema sender skjemaet.** Siden en manglende `type` setter knappen i tilstanden Auto, sender en «Legg til en rad til»- eller «Vis passord»-knapp inne i et skjema hele skjemaet i det øyeblikket den trykkes. Skriv `type="button"` på hver knapp som ikke skal sende, så forsvinner overraskelsen.
- **Ikonet er hele etiketten.** En søppelbøtte, en blyant eller en pil betyr noe for den som tegnet den. Har knappen bare ikonet, leser skjermleseren opp et filnavn, et symbolnavn eller bare «button». Et synlig ord er best. Der det mangler plass til et, fyller et tilgjengelig navn som sier handlingen, hullet.
- **En skjult eller deaktivert knapp behandles som sikkerhet.** Å fjerne knappen fra siden stopper bare ærlige brukere. Forespørselen bak den kan fortsatt sendes, og serveren må selv avgjøre om akkurat denne kontoen får gjøre akkurat dette.

Les mer

- WAI-ARIA 1.2: button role — definisjonen av rollen, variantene for av/på og meny, og tilstandene og egenskapene den støtter.
- HTML Living Standard: the button element — innholdsmodellen, typene submit, reset og button, tilstanden Auto og attributtene `command` og `commandfor`.
- ARIA Authoring Practices Guide: Button Pattern — knapper mot lenker, oppførsel på tastaturet, hvor fokus går etter et trykk, og regelen for etiketter på av/på-knapper.
- Understanding SC 2.5.8: Target Size (Minimum) — minstemålet på 24 ganger 24 piksler, avstandstesten og de andre unntakene.

- Understanding SC 4.1.2: Name, Role, Value — hvorfor hver kontroll trenger et navn og en rolle som programvare kan lese, og når vanlige HTML-kontroller allerede oppfyller kriteriet.

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 skjermknapp?» 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/button/