Funka

Smittestopp iOS

Fakta om rapporten

Fakta om rapporten

Oppdragsgiver: Norsk Helsenett

Utført av: Andreas Jacobsen, Kristian Munter Simonsen

Deler som ikke inngår: PDF-dokument, Tredjepartsløsinger, bank ID

Utstyr og nettlesere som ble benyttet: MacOS Catelina, iPhone 11 Pro, iPhone 12, Accessibiltiy Scanner via xcode, VoiceOver for iOS

Bakgrunn

Funka Nu AB har gjennomført en gransking av nettstedet for å bedømme hvor godt nettstedet oppfyller kravene til universell utforming. Granskingen består av et antall kontrollpunkter som vi har utviklet for å måle alle relevante aspekter knyttet til universell utforming i nettbaserte grensesnitt. For hvert punkt finnes følgende bedømmelser:

- **Ingen feil funnet.** Grensesnittet oppfyller kravene i punktet på en tilfredsstillende måte på de delene av løsningen vi har kontrollert. Det kan bety alt fra gode løsninger til akseptable løsninger.
- **Ingen feil funnet men kan forbedres.** Grensesnittet oppfyller kravene i punktet på en tilfredsstillende måte opp mot gjeldende lovpålagte krav. At punktet er satt som "kan forbedres" betyr at vi ser forbedringspotensialer utover det som er inkludert i lovpålagte krav.
- **Underkjent.** Det er opplagte problemer med universell utforming som må korrigeres.
- **Ikke aktuelt.** Grensesnittet bruker ikke den type løsning som kravet gjelder, eller så er punktet av en annen grunn ikke bedømt innen rammene for granskingen

Funkas metodikk er utviklet i nært samarbeid med den svenske handikapbevegelsen, og alt vi anbefaler er testet ut i virkeligheten. Vår virksomhet bygger på de internasjonale retningslinjene for universell utforming, Web Content Accessibility Guidelines 2.0 (WCAG 2.0). WCAG 2.0 AA er det nivået som i øyeblikket gjelder for det norske og europeiske regelverket. Når Webdirektivet trer i kraft i Norge vil minimumskravene i den europeiske standarden EN 301 549 gjelde, som kommer til å harmoniseres med WCAG 2.1, som ble publisert den 5.juni 2018.

Funkas lange erfaring med universell utforming og testing med brukere med ulike behov og forutsetninger, med og uten hjelpemidler, viser at WCAG 2.0 alene ikke er tilstrekkelig. Vi har derfor utarbeidet våre egne testkriterier for punktene som kompletterer det internasjonale regelverket som både offentlig og privat sektor er pålagt å følge gjennom det norske lovverket.

Funka har på oppdrag fra EU gjort tilstandsrapport innen universell utforming i EU og utvalgte land, deriblant Norge. Videre begynte vi i 2015 å utvikle målemetoder for den kommende EU-lovgivningen om universell utforming av IKT. I Norge har vi hjulpet Difi med veiledningstekster for hvordan jobbe med universell utforming,

samt med utvikling av indikatorsett for konkret statusmåling av et nettsteds grad av universell utforming.

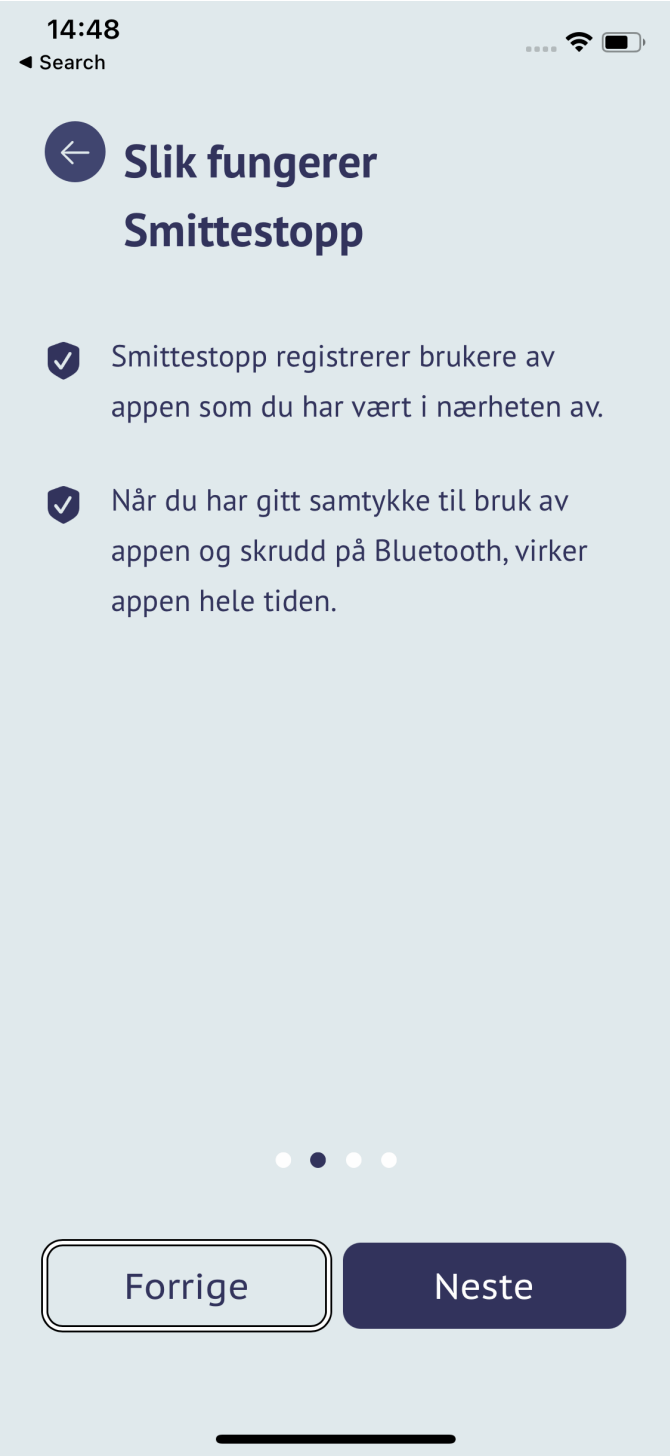
Funka er den eneste private aktøren i verden som har fått i oppdrag å gjøre offisiell oversettelse av WCAG. Denne oversettelsen danner også grunnlaget for den svenske E-delegasjonens Veiledning for webutvikling, som er de offisielle retningslinjene for hvordan man bør arbeide med nettsteder innom offentlig sektor i Sverige.

Funkas spesialister har siden 2008 deltatt i arbeidet med å ta frem EN 301 549 på oppdrag fra EU-kommisjonen. Vi er utnevnte eksperter i Special Task Force for ETSIs regning harmoniserer EN-standarden med WCAG 2.1, og for å kunne oppfylle kravene i Webdirektivet når det gjelder dokument og apper. Dessuten er vi valgt inn som tekniske eksperter i den svenske speilkommitéen hos Swedish Standards Institute, som representerer Sverige i standardiseringsarbeidet. Vi leder ekspertgruppen som gir EUs medlemsland og kommisjonen støtte i overgangsperioden for Webdirektivet, og utformingen av tilsynsmetodikk og tilgjengelighetsdeklarasjon.

Sammendrag av granskingen

Prioriterte områder

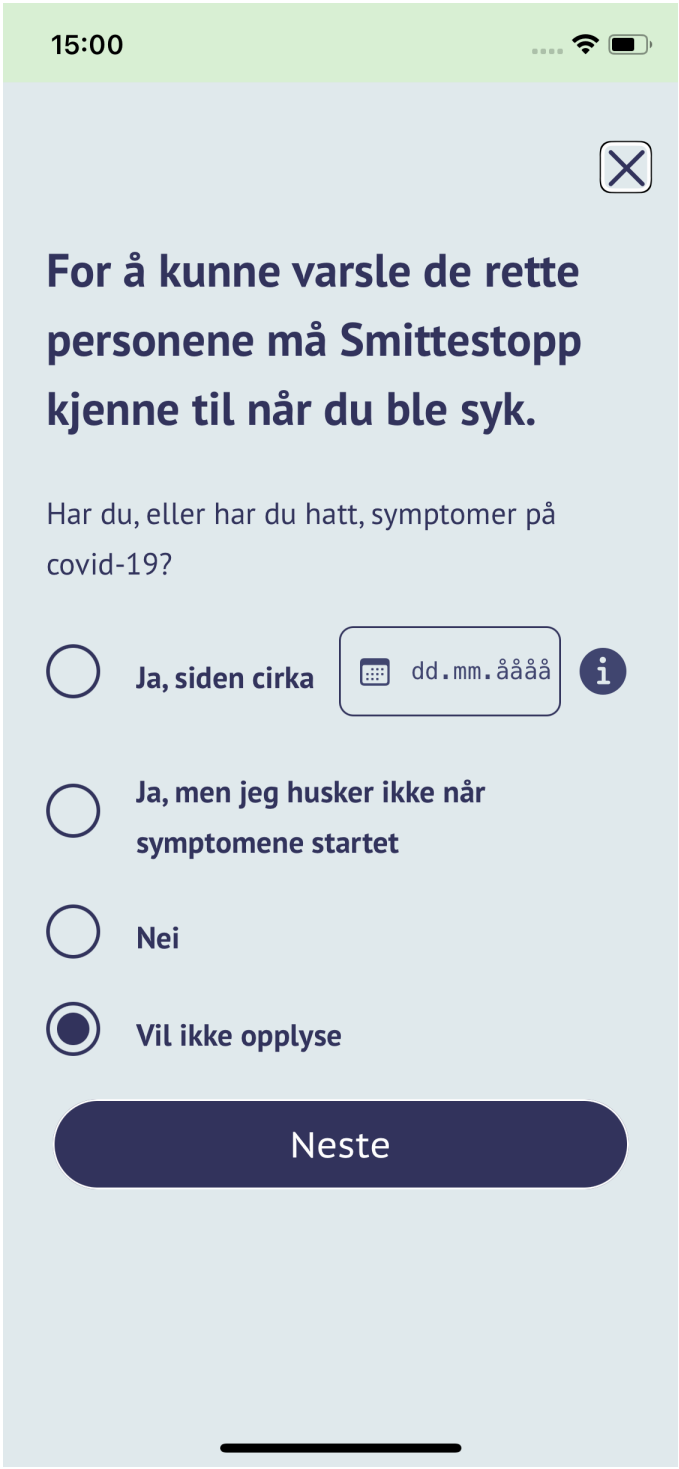
- Appen forteller ikke skjermlesere hvilket språk den er på, som forhindrer skjermlesere å automatisk sette korrekt språk. Dette løses dog relativt raskt av mange skjermleserbrukere ved å manuelt bytte språk
- Introduksjonssiden viser visuelt hvor mange sider det er, men kommuniserer ikke til skjermleser hvor mange sider det er eller hvilken side skjermleserbrukeren er på



• Fokus i instillingersiden settes automatisk til tilbake-knappen istedenfor på overskriften på siden under instillinger

• Fokus settes automatisk på X (lukk side) knappen istedenfor overskriften når brukere har rapportert positiv corona test. Dette kan føre til at brukere lukker sider med svært viktig informasjon siden de ikke forventer at lukke knappen er

det som først får fokus. Vi anbefaler å sette fokus på overskriften på alle sider.

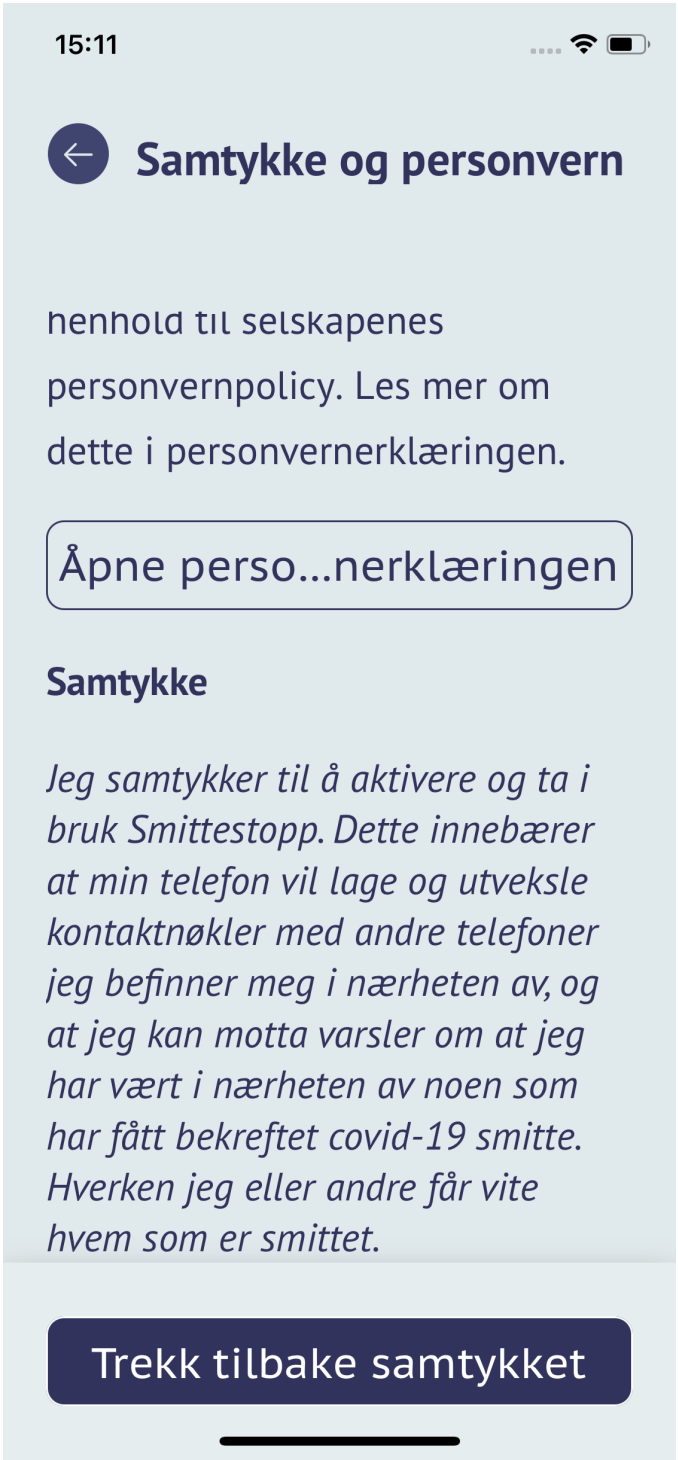


• Det gis ikke innformasjon til skjermlesere om hva radioknappene som setter språk under instillinger faktisk er. hvis en av radioknappene er valgt annonser VoiceOver "Markert" men på de ikke markerte knappene annonseres ingenting av skjermleseren. Det er dog mulig å navigere seg videre og for eksempel få høre "Nynorsk" annonsert av skjermleseren. Her må da brukere forstå at det tomme elementet vedsiden av "Nynorsk" faktisk er en radioknapp de kan klikke på.

• Ingen av overskriftene i appen er kodet som overskrifter

• FHI-logoen vises visuelt i appen men det er ingen måte for skjermlesere å nå logoen eller få informasjon om at den er der, selv om iOS støtter å navigere mellom bilder

• "Åpne personvernerklæring" overflower når man zoomer teksten i iOS

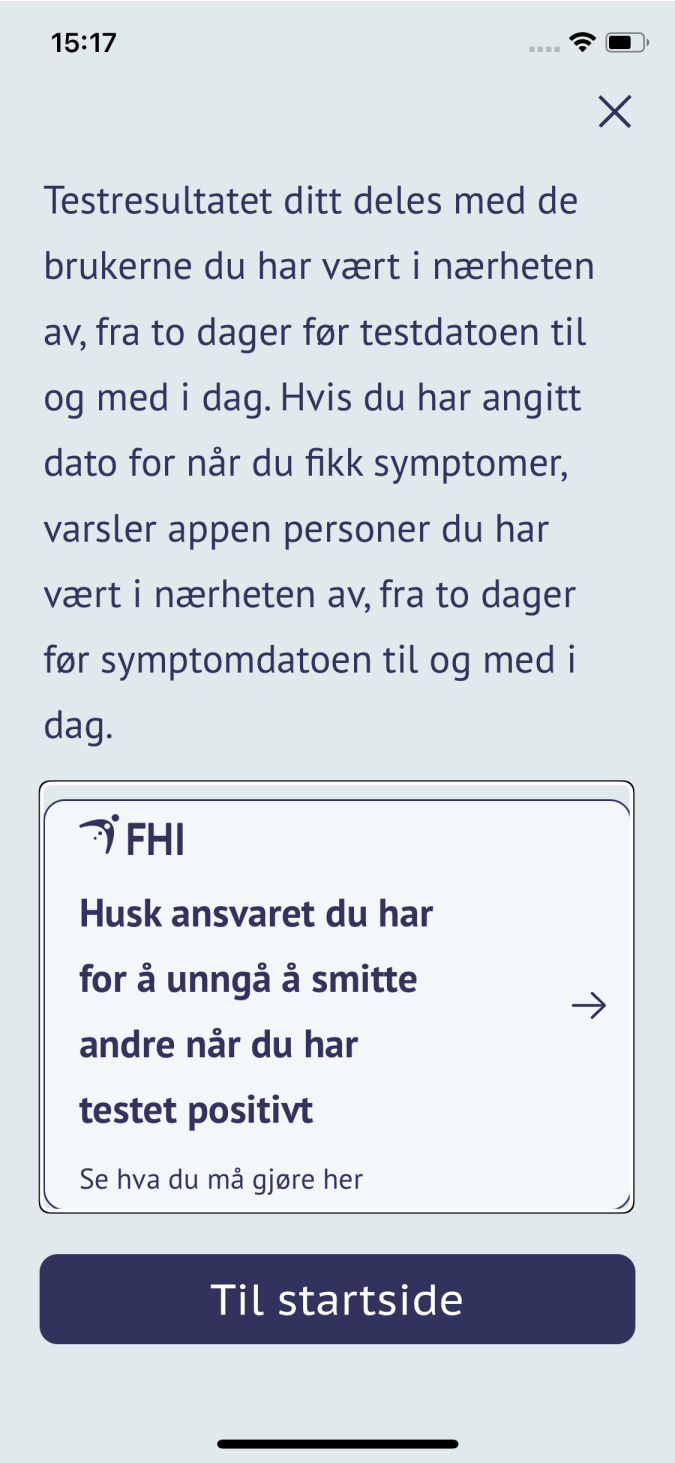


• Apple info boks når du deler informasjon om rammeverket er på engelsk selv om appen er på norsk

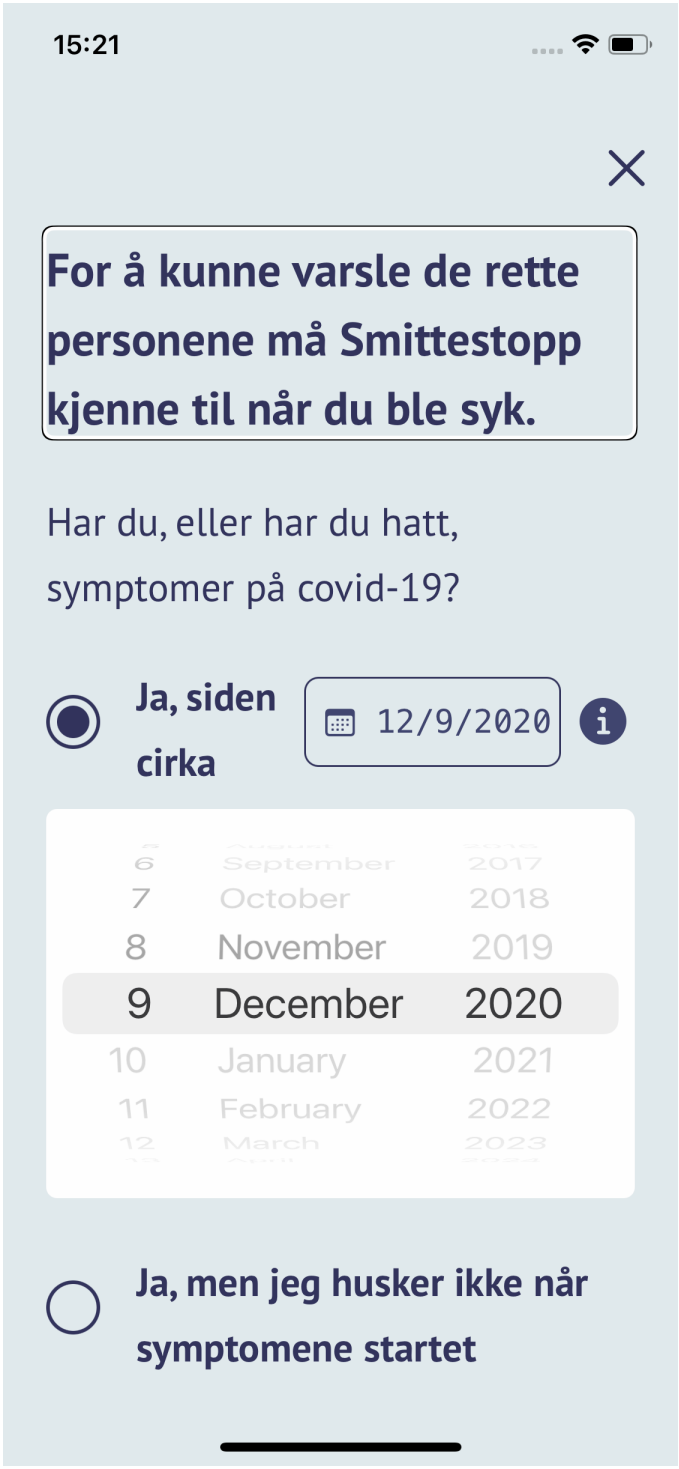
• Ved rapporter positive smitte får seende brukere opp meldingen "Takk for at du bidrar til å stoppe smitten. God bedring", VoiceOver brukere får annonsert "Takk for at du nå har deltat du har covid 19 i smittetopp"

• Etter å rapportert positiv smitte får seende brukere en knapp som sier "Husk

ansvaret du har for å unngå å smitte andre når du har testet positivt Se hva du må gjøre" men skjermleserbrukere får kun annonsert "Se hva du må gjøre"



• På siden man får når man har en positiv test kan seende brukere lese "For å kunne varsle de rette personene må Smittestopp kjenne til når du ble sykt", skjermlesere får annonsert "For å kunne varsle de rette personene må smittestopp kjenne symptomforløpet ditt". Dette betyr omtrent det samme men versjonen som gis til skjermleserbrukere er vanskeligere å forstå, symptomforløp er heller ikke det samme som "når du ble syk"


15:21
For å kunne varsle de rette personene må Smittestopp kjenne til når du ble syk.
Har du, eller har du hatt, symptomer på covid-19?
Ja, siden cirka
12/9/2020
6 September 2017
7 October 2018
8 November 2019
9 December 2020
10 January 2021
11 February 2022
12 March 2023
Ja, men jeg husker ikke når symptomene startet

Kommentarer og anbefalinger

Generelle poeng

GE90: Sider og funksjoner er konsekvent beskrevet i lenker, referanser og overskrifter

Ingen feil funnet men kan forbedres

Bakgrunn

Et viktig aspekt for at brukerne skal oppleve sidene som enkle og tydelige handler om at nettstedet er konsekvent i hvordan ulike funksjoner, lenker og områder beskrives. Det skaper en usikkerhet hos brukerne hvis nettstedet for eksempel bruker "E-tjenester" på noen sider, men kaller samme tjeneste for "Selvbetjening" på andre sider.

Kommentar

Visuelt er det god bruk av overskrifter, men disse er ikke kodet som overskrifter så skjermlesere får ingen informasjon om hva som faktisk er overskrifter.

Koblede retningslinjer

WCAG 2.0 - 3.2.4 (AA)

WCAG 2.1 - 3.2.4 (AA)

HTML og CSS

HC10: Velg teknikker som lar deg bygge universelt utformet

Ingen feil funnet

Bakgrunn

Det finnes en rekke ulike teknikker du kan bruke for å formidle informasjon og tjenester. Noen eksempler er HTML, CSS, Java, Flash, Silverlight og PDF. En del av disse fungerer bedre enn andre med tanke på muligheter for universell utforming. Hvilke teknikker som fungerer bra varierer også over tid. For en del år siden fungerte for eksempel Flash bra, men i dag er støtten for Flash for lite utbredt til at du bør bruke teknikken.

Vi anbefaler at du i første rekke skaper informasjon og funksjoner på en nettside med HTML, CSS, WAI-ARIA og JavaScript. Dette er teknikker som du kan bruke på en universelt utformet måte, samtidig som støtten blant nettlesere og hjelpemidler er godt utbygd.

Bruk ikke Java applets, Flash eller Silverlight. Dette er teknikker som har noe funksjonalitet for universell utforming, men som i praksis skaper problemer for brukerne. I enkelte tilfeller kan det være grunn til å bruke noen av disse teknikkene, men det skal være i en spesiell sammenheng hvor det ikke finnes noe annet alternativ, og der dere har full kontroll på brukerne og deres utstyr.

Koblede retningslinjer

AnnenConformance Criteria 4
WCAG 2.0 - 4.1.1 (A)
WCAG 2.1 - 4.1.1 (A)

HC20: Teknikker brukes på en måte som støtter universell utforming

− Ikke aktuelt

Bakgrunn

Det er ikke tilstrekkelig å velge en teknikk som kan brukes på en universelt utformet måte, du må også bruke teknikken korrekt. Dette handler blant annet om å bruke riktig HTML-element i enhver sammenheng, sørge for at skriptbaserte funksjoner fungerer med hjelpemidler, og at bruken av WAI-ARIA ikke skaper problemer for hjelpemiddelbrukere.

HTML

De fleste HTML-elementene har en semantisk betydning, alt fra overordnede ting som å definere navigasjonsområde og hovedinnholdsområde, til mer spesifikke ting, som lenker og overskrifter. Du skal bruke elementer som gir en verdi til brukeren, mens du bør unngå elementer som ikke tilfører verdi.

Mange av elementene som gir verdi til brukeren er så viktige og/eller skaper så store problemer at de har egne punkt senere i revisjonen. Andre elementer er derimot så innebygd i de fleste publiseringsverktøy og/eller i fingrene til de fleste utviklere, at vi velger ikke å løfte dem frem i egne punkt. Det sistnevnte gjelder for eksempel å kode lenker med <a>-element.

Grunnregelen er enkel: hold det enkelt, bruk elementer til det som de er tiltenkt, og forsøk å holde strukturen så enkel som mulig.

Script

Bruk av script stiller større krav på brukernes hjelpemidler enn hva kun HTML og CSS gjør. Det innebærer at risikoen for problemer øker. Vi anbefaler derfor at alle

komplekse funksjoner testes med de vanligste hjelpemidler for å sikre at ingen unødvendige vanskeligheter oppstår.

Eksempler på problemer som er vanlige, er at brukere med hjelpemidler ikke får fokus på lag som åpnes dynamisk, eller at de ikke kommer til søkeforslagene i søkefunksjonen. Dette er på grunn av at koden for nye lag og ny informasjon ofte legges inn på slutten av sidens struktur og ikke ved det objektet der brukeren har fokus. Et annet vanlig problem er at klikkbare områder skapes med script i stedet for med lenker og knapper. Da oppfatter ikke hjelpemidler dette som ekte lenker eller knapper, og presenterer de derfor ikke som interaktive objekt.

WAI-ARIA

WAI-ARIA er en spesifikasjon av kode som gjør det mulig å gi informasjon om nettstedet til hjelpemiddelbrukere. Med WAI-ARIA kan man for eksempel merke opp dynamisk innhold slik at hjelpemidler har lettere for å håndtere og presentere informasjonen til brukerne.

Vi anbefaler at WAI-ARIA benyttes, men det må gjøres med forsiktighet. Benytt ikke WAI-ARIA for å redde dårlig HTML. Det er for eksempel ingen god ide å benytte role=”button” på et div-element i stedet for å benytte en ekte knapp. Benytt heller ikke WAI-ARIA uten å ha kontroll på hvordan det faktisk fungerer for brukerne. Et vanlig problem er at man legger inn alt av WAI-ARIA man kan tenke seg med følgen at hjelpemidler blir veldig pratsomme uten at det er nødvendig.

Støtten for WAI-ARIA varierer mellom ulike versjoner av hjelpemidler og nettlesere. Det gjør at vi inntil videre anbefaler at alle dynamiske løsninger testes i praksis med ulike hjelpemidler, enheter og versjoner.

Informasjon om WAI-ARIA (W3C’s nettsted)

Test med hjelpemiddel

For å være sikker på at script ikke skaper problemer for hjelpemiddelbrukere behøver dette ofte testes i praksis. Nøyaktig hvilke hjelpemidler som burde testes varierer avhengig av løsningen. Det er heller ikke bare å velge et hjelpemiddel og gå gjennom tjenesten. Det er nødvendig med blant annet kunnskap om brukernes bakgrunnskunnskap og tankesett, men også hvordan brukerne håndterer grensesnittet, hvordan de benytter hjelpemiddelet, hvordan det oppleves om man for eksempel ikke ser og så videre. Som utvikler eller designer er det veldig vanskelig å teste løsningen objektivt, siden man da har forkunnskaper som brukerne mangler.

Det anbefales derfor å la eksperter teste med hjelpemidler.

Koblede retningslinjer

WCAG 2.0 - 4.1.1 (A)

WCAG 2.1 - 4.1.1 (A)

WCAG 2.0 - 4.1.2 (A)

WCAG 2.1 - 4.1.2 (A)

HC30: HTML-koden inneholder ingen alvorlige feil

Ikke aktuelt

Bakgrunn

Det eksisterer flere ulike HTML-standarder, blant annet HTML5 og XHTML 1.0 strict. Hver standard har sitt regeldokument for hvilke element og attributt du kan bruke og hvordan.

Når du koder HTML, må du først ta stilling til hvilken standard koden skal forholde seg til. Vi anbefaler i første rekke å vurdere og bruke HTML5, ettersom det her finnes en del attributter og elementer som hever nivået på universell utforming sammenlignet med eldre standarder. Denne anbefalingen baserer seg dog på at HTML5 brukes forsiktig. Det er problemer med enkelte attributt og element, ved at de ikke har støtte i alle nettlesere og hjelpemidler, eller ikke blir formidlet likelydende av ulike hjelpemidler.

En viktig prinsipp er også å effektivisere koden. Det handler blant annet om å skille ut script og stilmaler til egne filer og å effektivisere strukturen og formateringen.

!DOCTYPE

For at nettlesere, hjelpemidler og andre verktøy skal tolke koden korrekt, må de vite hvilken standard den aktuelle siden er kodet i. Dette må du derfor angi øverst i koden, på samtlige HTML-sider på nettstedet, med en !DOCTYPE-deklarasjon.

!DOCTYPE deklarerer du som følger i HTML5:

```
<!DOCTYPE HTML>
```

Hvis dere velger å følge XHTML i stedet for HTML5, skal deklarasjonen inneholde henvisninger til regeldokument for standarden også. Slik ser en deklarasjon av XHTML 1.0 transitional ut:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0
```

```
Transitional//EN" "http://www.w3.org/TR/xHTML1/DTD/xHTML1-
transitional.dtd">
```

DTD-filen definerer nøyaktig hvilke element, attributt og ulike verdier som er tillatt ifølge standarden.

HTML5

Hvis du velger å følge HTML5, skal du gjøre det med varsomhet ettersom støtten for ulike deler i standarden varierer mellom nettlesere og hjelpemidler. Ta inn nye deler i HTML5 standarden når det øker brukervennligheten og ikke gjør det vanskeligere for brukerne.

Inntil videre bør du unngå å bruke HTML-elementene section og article, med mindre du har svært spesifikke årsaker for det. Grunnen er at disse skaper en struktur som kan påvirke sidenes overskriftstruktur. Ulike hjelpemidler tolker dette på ulike måter, noe som gjør at brukere som er avhengig av hjelpemiddel for å forstå sidenes struktur (for eksempel alvorlig svaksynte brukere) kan få ulike overskriftstrukturer avhengig av hjelpemiddelversjonen.

Tenk også på bakoverkompatibilitet. Dersom en noe eldre nettleser ikke har støtte for HTML5, må det det finnes en fallbackløsning. Les mer om dette på T1a ovenfor.

Følg valgt standard

Det er ikke tilstrekkelig å deklarere med !DOCTYPE for at en nettside skal følge en viss standard. Det er også viktig å faktisk følge den standarden du har angitt i koden. Standarden er utviklet for at alle utviklere av nettsteder, nettlesere og hjelpemidler skal ha en felles kilde å forholde seg til for å sikre at innholdet på nettstedet presenteres på en optimal måte for besøkeren.

Hvis nettstedet ikke følger standarden, risikerer dere at brukeren får problemer med å ta del i innholdet. Blir problemene for store, risikerer dere å utestenge brukere fullstendig.

Det finnes flere ulike valideringsverktøy for å kontrollere nettstedets kode. W3C har en egen HTML-validator som kan nås på adressen:

http://validator.w3.org/

Alle avvik fra valgt standard resulterer ikke i like store tilgjengelighetsproblemer. Enkelte typer avvik er mer alvorlig enn andre. W3C peker i WCAG 2.0 ut fire prinsipper som koden som et minimum må følge:

- Bruk komplette start- og avslutningstagger.
- Ikke angi samme id-verdi på to ulike elementer på samme side.
- Ikke angi samme attributt to ganger i samme element (for eksempel to alt-attributter i samme img-element).
- Elementetene et nøstet i henhold til standarden.

Disse betraktes som alvorlige, og hvis de forekommer oppfyller ikke grensesnittet kravene i WCAG 2.0.

Koblede retningslinjer

WCAG 2.0 - 4.1.1 (A)

WCAG 2.1 - 4.1.1 (A)

HC80: Innholdets leserekkefølge er logisk

Ingen feil funnet

Bakgrunn

Når du styrer layout og presentasjon med hjelp av CSS, kan du posisjonere ulike områder visuelt helt uavhengig av hvor de er plassert i HTML-koden. Hjelpemiddelet leser derimot nettstedets HTML-kode ovenfra og ned, som innebærer at presentasjonsrekkefølgen kommer i den rekkefølgen innholdet er lagt i koden og ikke den visuelle rekkefølgen som CSS-koden bestemmer. Innholdets rekkefølge må derfor være logisk og speile den visuelle rekkefølgen.

Koblede retningslinjer

WCAG 2.0 - 1.3.2 (A)

WCAG 2.1 - 1.3.2 (A)

Grensesnittets fleksibilitet

RD10: Grensesnittet fungerer bra i ulike skjermbredder

Ingen feil funnet

Bakgrunn

For de fleste webgrensesnitt er det rimelig å tenke seg at brukeren av og til kommer til å bruke en liten og smal skjerm, og av og til en større og bredere. Da er det også rimelig at grensesnittet skal fungere bra med ulike skjermbredder.

Oftest er det smart å dele opp konseptet i tre hovedkategorier:

- Store skjermer, som på vanlige desktops og bærbar PC/Mac med normal skjermstørrelse.
- Mellomstore skjermer, for eksempel de største nettbrettene og bærbar PC/Mac med små skjermer.
- Små skjermer, som mobiltelefoner og små nettbrett.

Det kan være nødvendig med en finere inndeling avhengig av grensesnitt, men det er viktig at det fungerer godt i hele spekteret fra små mobiler til stor desktops, og at det fungerer både i stående og liggende visning.

Viktige prinsipper

Det finnes ulike prinsipper og metoder som kan brukes for å bygge responsive løsninger. Om linjelengder og flater tilpasser seg etter bredden på vindu er det bra, men det viktigeste er at grensesnittet er mulig å bruke i alt fra smale skjermer til svært brede skjermer, og at presentasjonsrekkefølgen er logisk uansett skjermbredde. Det omfatter å hindre at tekstlinjer blir for korte slik at de ikke klarer å vise hele ord, eller for lange slik at det blir krevende å finne neste linje. Tekstlinjene skal ikke vise mer enn maksimalt 70 tegn per linje inklusive mellomrom.

Så fort menyen kollapser til en menyknapp øker problemene for brukeren. Derfor er det viktig å prøve å opprettholde en meny der nivå 1 er direkte synlig i størst mulig grad. Det betyr for eksempel at menynivå 1 skal være synlig i mellomstørrelsen så langt det er mulig.

I utgangspunktet bør det vanligste grensesnittet fungere uavhengig av skjermbredde, men i spesifikke situasjoner kan det være ønskelig å tilby et alternativt, forenklet grensesnitt for små skjermer. Da er det viktig at brukerne ikke tvinges til en viss versjon, men at de kan velge det ordinære grensesnittet, selv på

små skjermer. Dette løses ved å tilby en lenke til det ordinære grensesnittet fra mobilversjonen. Det kan også være bra å tilby en lenke til mobilversjonen fra det ordinære grensesnittet, til tross for at brukeren sitter ved en stasjonær datamaskin. For eksempel setter en del brukere med konsentrasjonsproblemer pris på en forenklet visning i mobilversjonen.

Koblede retningslinjer

WCAG 2.0 - 1.4.8 (AAA)

WCAG 2.1 - 1.4.8 (AAA)

WCAG 2.1 - 1.4.10 (AA)

RD30: Grensesnittet skal kunne vises og brukes uten at brukeren må rulle i mer enn én retning

Ingen feil funnet

Bakgrunn

Det er umulig å forutse hvordan en bruker velger å vise eller å bruke et grensesnitt. Det innebærer at grensesnittet må være fleksibelt og tilpasse seg den skjermbredden som brukeren velger å vise det i. Grensesnittet skal være responsivt og tilpasse seg brukerens skjermbredde uten at innhold eller funksjon skal forsvinne. Om innhold eller funksjoner ikke er synlig på brukerens skjerm er skrolling tillat enten vertikalt eller horisontalt, men ikke i begge retninger samtidig. Grensesnittet må fungere med denne forutsetningen fra minst 320pixler bredde, eller 256 pixler høyde. I blant kan unntak være nødvendige for spesielt innhold, for eksempel tabeller, diagrammer eller illustrasjoner.

Koblede retningslinjer

WCAG 2.1 - 1.4.10 (AA)

RD40: Nettstedet er fullt brukbart og lesbart ved forstørring opp til og med 200 prosent

Ingen feil funnet

Bakgrunn

En del nettlesere, for eksempel Internet Explorer, gir brukeren mulighet til å tilpasse tekststørrelsen dersom tekstene er matet inn med relative mål (for eksempel em og %). Denne innstillingen har i dag havnet i skyggen av den mer utspredde zoom-funksjonen som forstørrer alt, uavhengig av målenheter på teksten.

Det er ikke lenger et krav å bygge nettstedet med relative mål, men det er viktig at teksten ikke blir vanskelig å lese og funksjonene vanskelig å bruke dersom brukeren har endret innstillingene for tekststørrelse i nettleseren. Om nettstedet bygges med relative størrelser må det altså gjøres på en bra måte som ikke forvrenger visningen av sidene i nettleseren. En enkel måte å teste dette på er å bruke funksjonen Tekststørrelse i Internet Explorer. Ta frem menyen (er den ikke synlig så trykk på Alt-tangenten) og velg ”Vis” > ”Tekststørrelse”. Prøv med både ”Størst” og ”Minst”.

Det må også være mulig for brukeren å zoome nettstedet opp til 200 % uten at flater legger seg over hverandre og gjør teksten tunglest eller funksjonene vanskelig å bruke. Dette gjelder ikke bare store skjermer med HD-oppløsning, men også på mindre skjermer og i mobiltelefoner.

Tenk også på at responsivt bygde grensesnitt skal tilpasse seg når brukeren zoomer. Det skal altså være mulig, selv med zoom aktivert, å komme til alle funksjoner og all informasjon i et responsivt grensesnitt. Det er selvfølgelig en grense der en alt for liten skjerm med alt for mye zoom ikke fungerer, men når brukeren når dit skal denne fortsatt kunne zoome videre, selv om det da kan innebære at brukeren må skrolle sidelengs for å kunne lese all tekst.

Du sikrer at brukeren kan zoome, også i mobile enheter, med følgende kode i <head>:

```
<meta name="viewport" content="user-scalable=yes">
```

Attributtet content kan inneholde mange flere delattributt og tilhørende verdier for å styre presentasjonen på ulike skjermer. Bruken av delattributtet maximum-scale kan hindre brukeren mulighet til å zoome selv om du har angitt user-scalable=yes.

Vi anbefaler derfor at du ikke bruker maximum-scale. Øvrige delattributt har vi frem til nå ikke sett noen problemer med.

Den nøyaktige løsningen for hvordan forstørring og endring av vindusstørrelse skal håndteres må skreddersys fra gang til gang, men her er noen korte anbefalinger for vanlige webgrensesnitt:

- Nettstedet skal automatisk tilpasse seg etter vindusstørrelsen, men det er viktig å legge inn mekanismer som hindrer at linjelengdene overstiger 70 tegn inklusive mellomslag, eller at de blir så korte at ikke hele ord kan vises.
- Om brukeren endrer tekstforstørringsfunksjonen i nettleseren skal ikke dette forårsake at innhold ikke kan leses eller funksjoner ikke kan brukes.
- Forstørring via nettleser med zoomfunksjon skal minst fungere opp til 200 %.
- Ved forminskning av vinduer eller tekststørrelse skal teksten fortsatt være mulig å lese og funksjonene mulig å bruke.

Kommentar

Testet med både zoom og tekst-zoom i iOS

Koblede retningslinjer

WCAG 2.0 - 1.4.4 (AA)
WCAG 2.0 - 1.4.8 (AAA)
WCAG 2.1 - 1.4.4 (AA)
WCAG 2.1 - 1.4.8 (AAA)

RD60: Brukere kan få tilgang til all informasjon og bruke alle funksjoner uavhengig av skjermretning

× Underkjent

Bakgrunn

Når nytt innhold presenteres for en bruker skal ikke innholdet være låst til liggende eller stående visning. Brukeren skal kunne velge hvordan innholdet vises. Iblant er det ikke mulig uten at målet med funksjonen mistes. Film og tabeller skal være responsive og kunne vises uansett orientering, dersom det ikke finnes spesielle grunner for det. Film i fullskjerm krever ofte at enheten har en liggende retning, men skal også kunne spilles av i stående retning.

I ulike situasjoner kan brukeren trenge å bruke mobiltelefoner eller nettbrett i en spesiell retning, for eksempel om enheten sitter fast i en holder på en rullestol.

Grensesnittets retning skal da tilpasses etter brukerens forutsetninger. En konsekvens blir at opplevelsen ikke alltid blir optimal, men det skal fungere. Et eksempel på dette er filmvisning i stående visning.

Denne retningslinjen gjelder ikke forandringer av innhold eller funksjoner på grunn av skjermens størrelse, men gjelder kun skjermretning.

Kommentar

Appen er låst til vertikal modus, brukere som benytter seg av sin mobiltelefon horisontalt vil nok fortsatt klare å benytte appen. Men det gir ett mye høyere nivå av komfor for disse brukerene å også støtte horsintal modus i appen.

Anbefalinger

Støtt både horisontal og vertikal modus i appen.

Koblede retningslinjer

WCAG 2.1 - 1.3.4 (AA)

Navigasjon og linker

NL20: Grensesnittet kan styres ved hjelp av tastatur på både mobil og stor skjerm

× Underkjent

Bakgrunn

Det finnes en mengde ulike måter å styre en datamaskin eller telefon på. Det finnes ulike løsninger for å bevege musepekeren, klikke, samt å fylle inn tekst. Ved å gjøre grensesnittet tilgjengelig for styring med mus, tastatur og berøringsskjerm, har brukeren mulighet til å selv velge styreløsning. For en del brukere finnes det ingen mulige alternativ. Noen brukere har ikke mulighet til å benytte tastatur, og må navigere med mus. Andre brukere kan ikke bruke noen form for mus-løsning, og må i stedet benytte tastaturet.

Et grunnprinsipp i WCAG er at det skal være mulig å styre grensesnittet kun med tastatur, uansett om det er på en stor eller liten skjerm. For at det skal kunne anses å følge retningslinjene, må det fungere.

Unntak for dette punktet kan forekomme om det for eksempel gjelder en funksjon som ikke på noen rimelig måte kan gjøres mulig å benytte uavhengig av styreløsning.

Vanlige problem

Unngå å legge inn funksjoner som krever at brukeren klikker eller holder musepekeren over ulike områder, som for eksempel menyer som krever at brukeren peker med musen. Unngå også lister som oppdaterer siden eller flytter fokus så fort verdien i listen endres. Slike løsninger skaper store problemer, særlig for personer som må navigere med tastaturet, men også for personer som har nedsatt presisjonsevne eller som sitter i urolige miljøer.

Vær også forsiktig med å endre interaksjonsmønster som brukeren er vant med. For eksempel er brukere som navigerer med tastaturet vant med å benytte tab-tangenten for å bevege seg mellom lenker og skjemaer. Om det da plutselig forventes av brukeren at det navigeres med piltastene i stedet, kommer mange brukere til å oppleve problemer. Det er også vanlig at det kun trengs ett klikk med musen. Om det plutselig kreves dobbeltklikk i en nettside vil mange brukere oppleve problemer.

Merk også at kombinasjonen av responsivt design og zoom ofte resulterer i en mobilvisning selv for brukere med desktop. Det er derfor viktig at det fungerer uavhengig av input-enhet selv i mobilvisning.

Kommentar

Se NL30

Koblede retningslinjer

WCAG 2.0 - 2.1.1 (A)
WCAG 2.0 - 2.1.2 (A)
WCAG 2.0 - 2.1.3 (AAA)
WCAG 2.0 - 3.2.1 (A)
WCAG 2.0 - 3.2.2 (A)
WCAG 2.1 - 2.1.1 (A)
WCAG 2.1 - 2.1.2 (A)
WCAG 2.1 - 2.1.3 (AAA)
WCAG 2.1 - 3.2.1 (A)
WCAG 2.1 - 3.2.2 (A)

NL30: Rekkefølgen ved tastaturnavigering er logisk

× Underkjent

Bakgrunn

Når brukere navigerer med tastaturet, kan ikke brukeren styre rekkefølgen på samme måte som brukere som benytter mus eller pekeskjerm. I stedet er det nettstedet som styrer navigasjonsrekkefølgen. Dette gjelder ofte kun rekkefølgen ved bruk av tab-tasten. Den vanligste metoden for å hoppe mellom objekter med tastaturet er å trykke gjentatte ganger på tab-tasten. Da flytter fokus mellom nettsidens ulike lenker og skjemaobjekt. Rekkefølgen ved bruk av tab er som oftest den samme rekkefølgen innholdet ligger i koden. Hver gang brukeren trykker på tab-tasten hopper fokuset fra en lenke eller et skjemaobjekt til neste.

Attributtet tabindex kan benyttes for å styre rekkefølgen. Det kan i helt spesielle tilfeller finnes årsaker til å benytte dette attributtet, men alle objekt som brukeren skal komme til ved bruk av tab-tasten burde ligge i en logisk rekkefølge i sidens struktur, og da behøves ikke tabindex for å styre rekkefølgen.

Tabindex fungerer som følger:

- **Tabindex = 0** legger til inaktive objekt, som for eksempel en div, i rekkefølgen på det stedet de naturlig ligger i sidens struktur. Vår anbefaling er å benytte HTML-element som har innebygd interaktivitet, slik som lenker og skjemaobjekt, men det kan finnes situasjoner der det er nødvendig å la brukeren tabbe til andre typer objekt, og da er tabindex=0 metoden for å oppnå dette.
- **Tabindex > 1** prioriterer objekter og endrer navigasjonsrekkefølgen slik at den ikke følger sidens struktur. Rekkefølgen blir fra laveste nummer og oppover. Etter at brukeren kommer til objektet med høyeste tabindex-verdi så fortsetter rekkefølgen fra dette objektet i koden. Det gjør at brukeren ikke kommer seg til objekt som ligger tidligere i sidens struktur om de ikke har tabindex erklært.
- **Tabindex < 0** gjør at objektet ikke går å tabbe seg til, men at det kan få fokus gjennom script. Kan for eksempel benyttes for å sette fokus på en feilmelding med script slik at det formidles til hjelpemiddelsbrukere. Normalt sett benyttes verdien -1 for dette.

I en del funksjoner navigerer brukeren med andre taster også, for eksempel piltastene for å flytte fokus mellom ulike radioknapper. Det er viktig at ikke bare

tabrekkefølgen er logisk, men også at rekkefølgen generelt med tastaturnavigasjon har en logisk rekkefølge.

Kommentar

Det er ikke mulig å tastaturnavigere seg til "Meny" elementet (hamburgermenyen) på iOS 14. Det er heller ikke mulig å tab navigere seg til noen av elementene på forsiden, men det er mulig å nå de interagerbare elemente utenom hamburgermenyen med piltastnavigering.

Koblede retningslinjer

WCAG 2.0 - 2.4.3 (A)

WCAG 2.1 - 2.4.3 (A)

NL40: Fokus er visuelt tydelig når brukeren forflytter seg med tastatur

Ingen feil funnet

Bakgrunn

Når brukeren forflytter seg med tastatur på et nettsted eller i en funksjon, må det være visuelt tydelig hvor fokus er. Når man for eksempel benytter Tab-tasten for å ta seg gjennom menyen kan det menyalternativet som akkurat nå har fokus fremheves med en ramme eller ved at fargerne endres. I en del funksjoner kan man forflytte seg med piltastene og da gjelder samme anbefaling: Det objektet som har fokus skal være visuelt tydelig fremhevet.

Selv skjulte lenker og snarveier skal løftes frem og vises visuelt når de får fokus ved tastaturnavigering.

Den visuelle fremhevingen skal være tydelig. Det holder altså ikke med den innebygde fremhevingen i nettleseren. Det kan i stedet være en noe tykkere ramme, en tykkere understreking, invertering av fager eller en endring av bakgrunn.

Koblede retningslinjer

WCAG 2.0 - 2.4.7 (AA)

WCAG 2.1 - 2.4.7 (AA)

NL50: Det finnes snarveier for å muliggjøre raskere navigering med tastaturet

Ikke aktuelt

Bakgrunn

For brukere som navigerer med tastaturet kan det være krevende å måtte tabbe seg gjennom hele menystrukturen på alle sider. Har man funnet den siden man søkte, er det innholdet man er interessert i.

For å tilrettelegge for disse brukerne, skal det finnes snarveier på sidene. Den vanligste snarveien er forbi navigasjonen til sidens innhold, men i ulike sammenheng kan det også være gunstig med lenker tilbake til sidens topp eller til ulike menygrupper på siden.

Denne typen snarveier skapes ved å legge inn lenker som leder til element lenger ned på siden. Disse lenkene skal i utgangspunktet være visuelt skjult, men vises når de får fokus med tastaturet.

Det vanligste er å lenke til innholdets hovedoverskrift. Denne kan utformes slik:

```
<h1 id="content" tabindex="-1">Innholdets
hovedoverskrift</h1>
```

Tabindex="-1" benyttes for å sikre riktig oppførsel i alle nettlesere.

Snarveilenker skapes som vanlige lenker, med tegnet # fulgt av ankerets id. For eksempel:

```
<a href="#content" accesskey="s">
  Hopp til hovedinnholdet
 </a>
```

Koblede retningslinjer

WCAG 2.0 - 2.4.1 (A)

WCAG 2.1 - 2.4.1 (A)

NL70: Hurtigkommandoer aktiveres ikke bare med alfanumeriske taster

Ikke aktuelt

Bakgrunn

Mange brukere styrer grensesnittet helt eller delvis med tastaturnavigasjon og hurtigtaster. For en del brukere med motoriske funksjonsnedsettelser eller sterk synsnedsettelse er tastatur den primære måten å styre grensesnittet på. Selv stemmestyring innebærer i mange situasjoner at talekommandoen omvandles til tastaturkommandoer. For eksempel kan en person som styrer grensesnittet med stemmen si ”tab” for å hoppe til neste lenke, noe som innebærer at hjelpemiddelet sender tastaturkommandoen ”tab” til grensesnittet.

Om denne type hurtigtast brukes, så må noe av følgende være oppfylt:

- Det finnes en mekanisme for å stenge av hurtigtasten.
- Hurtigtasten kan endres, slik at de styres av en annen tast.
- Hurtigtasten er bare mulig å bruke når objektet som styrer kommandoen er i fokus.

HTML-attributtet accesskey omfattes ikke av denne retningslinjen ettersom den typen av hurtigtaster aktiveres sammen med Ctrl-, Alt- eller Command-tasten.

Koblede retningslinjer

WCAG 2.1 - 2.1.4 (A)

NL90: Påbegynt klikkhendelse skal kunne avbrytes eller angres

Ingen feil funnet

Bakgrunn

Et klikk består av to deler, delvis en ned-hendelse, og delvis en opp-hendelse. Ned-hendelsen skjer når brukeren trykker ned musepekeren sin eller fingeren sin på skjermen. Opp-hendelsen er når knappen slippes eller fingeren løftes fra skjermen.

Når brukeren navigerer i et grensesnitt gjennom å klikke seg frem med musepekeren eller med en berøringsskjerm, inntreffer iblant utilsiktete hendelser, hvor brukeren helt enkelt trykker på feil sted. Gjennom å unngå funksjonalitet som skjer når brukeren klikker ned minsker de problemene. La i stedet funksjonaliteten aktiveres når brukeren slipper museknappen.

I visse tilfeller er det ikke mulig, for eksempel for en piano-app, der brukeren

spiller på pianoet gjennom å klikke ned tastene.

Om en hendelse må aktiveres gjennom ned-hendelse så skal det om mulig finnes en sjanse for brukeren å angre hendelsen. Et eksempel på dette er mange skjermtastaturer for mobiler. Gjennom å la fingeren hvile på en bokstav dukker et antall mindre vanlige bokstaver opp, som baserer seg på den bokstaven brukeren har hvilt fingeren over (for eksempel kan du skrive á â om du hviler med fingeren over A-tasten). Om brukeren slipper skjermen uten å velge noe skrives heller ikke noe tegn.

Koblede retningslinjer

WCAG 2.1 - 2.5.2 (A)

NL100: Innhold og funksjoner skal kunne brukes med enkle gester

Ingen feil funnet

Bakgrunn

Brukere med nedsatt motorikk i hender og fingre har vansker med å kontrollere den presisjonen som ofte kreves for å styre et grensesnitt. Det finnes også inntastingsteknikker der det ikke går å gjøre komplekse gester, som øyestyring. For eksempel er det veldig vanskelig eller umulig å gjøre gester som krever mer enn ett berøringspunkt med øyestyring, og å gi en bevegelsesbane med bryterstyring.

Derfor er det viktig å skape grensesnitt som bygger på så enkle gester som mulig, utover de gester som kreves for å styre operativsystem og nettlesere. Det skal finnes alternative måter som gjør det mulig for disse brukerne å enkelt styre grensesnittet uten komplekse gester. Brukere med hodemus eller øyestyring blir ikke hjulpet av hurtigtaster via et skjermtastatur. Gester skal i stedet kompletteres med knapper som brukeren kan trykke på.

Eksempler på et type grensesnitt som bygger på gester er serverbaserte kart der brukeren gjennom å knipe sammen eller fra hverandre med to fingre zoomer inn eller ut i et kart. En annen gest er å flytte det geografiske området som finnes i kartet med to fingre vertikalt eller horisontalt. Alternativet for brukere med nedsatt motorikk er at zoomfunksjonen kan styres med knapper: pluss for å zoome inn og minus for å zoome ut. Knapper for høyre, venstre, opp og ned lar brukeren vippe og panorere i kartet for å endre det geografiske området.

Koblede retningslinjer

WCAG 2.1 - 2.5.1 (A)

NL110: Synlige beskrivelser av klikkbare objekter beskrives på lignende måte gjennom oppmerking i koden

Ikke aktuelt

Bakgrunn

Det finnes mange teknikker for å beskrive interaktive objekter. For vanlige skjemafelt brukes ofte en synlig ledetekst koblet til feltet. Dette holder oftest, dersom ledeteksten er lagt inn på riktig måte, men i visse situasjoner brukes også visuelt skjulte beskrivelser lagt inn med for eksempel aria-label. Det kan finnes mange grunner til å legge inn en visuelt skjult beskrivelse. Det kan for eksempel handle om at en bruker som ikke ser grensesnittet trenger en utvidet beskrivelse av feltet.

Når skjulte beskrivelser brukes er det dog viktig at de inkluderer selv det som vises visuelt. Et objekt med en etikett som inneholder teksten ”Kjøp” eller ”Les mer”, eller et bilde som illustrerer objektets funksjon skal i den visuelt skjulte etiketten eller beskrivelsen innledes med samme tekst. En knapp med beskrivelsen ”Kjøp billett” skal for eksempel også innledes med formuleringen ”Kjøp billett” i attributtet aria-label eller aria-labelledby.

Brukere med kognitive eller sterkt motoriske funksjonsnedsettelser kan bruke stemmen for å styre et grensesnitt, i stedet for å bruke tastatur eller berøringsskjerm for å navigere. Når visuelt synlige og skjulte etiketter samsvarer fungerer også stemmestyring på en god og forutsigbar måte.

Koblede retningslinjer

WCAG 2.1 - 2.5.3 (A)

NL120: Funksjonalitet som aktiveres eller styres gjennom bevegelse kan inaktiveres og styres på andre måter

Ingen feil funnet

Bakgrunn

Mobile enheter kan foruten om tastatur og tale, også bruke akselerometer, gyroskop og kamera som metode for inntasting fra brukeren, eller for å utføre en funksjon i enheten.

For brukere med motoriske funksjonsnedsettelser er det problematisk når funksjoner i et grensesnitt er basert på bevegelse. Det forutsetter at brukeren kan røre fritt på sin enhet, som for eksempel ikke er mulig for brukere som har sin enhet montert på rullestolen sin. Om brukeren ikke tilbys et alternativ kan det være umulig for visse brukere å styre et grensesnitt. Derfor skal grensesnittet, når det er mulig, kunne styres med knapper, lenker og andre objekter, som alternativ til bevegelse.

Det finnes unntak for funksjoner som er avhengig av bevegelse for å fungere, for eksempel skrittellere og vaterpass.

Koblede retningslinjer

WCAG 2.1 - 2.5.4 (A)

NL140: Markering av interagerbare objekter og nærliggende farger skal ha en kontrast på minst 3,0:1

Ingen feil funnet

Bakgrunn

Det er ikke bare tekst som trenger å være tydelig og ha tilstrekkelige kontraster i et grensesnitt. Interaktive objekter som knapper, lenker og skjemafelt skal tydelig kunne skilles ut, selv i dårlige lysforhold. For at også brukere med lettere synsnedsettelser skal kunne skille ut grensesnittkomponenter, er det viktig at rammer rundt sånne objekter har gode kontraster mot nærliggende farger. Her skal kontrastverdien være minst 3,0:1.

Kontrastkravet mellom ramme- og bakgrunnsfarge gjelder i flere situasjoner: når brukeren markerer et klikkbart objekt eller skjemaobjekt, når et objekt er i fokus når brukeren styrer med tastatur, eller når brukeren har valgt et spesifikt objekt.

Kontrastkravet gjelder ikke komponenter hvor utformingen ikke er tilpasset,

ettersom utseende på disse bestemmes av brukerens nettleser. Det gjelder heller ikke interaktive komponenter.

Koblede retningslinjer

WCAG 2.1 - 1.4.11 (AA)

NL170: Lenkegrupper og informasjonsområder er gruppert

Ikke aktuelt

Bakgrunn

For å gjøre det lettere for brukeren å identifisere grupper av objekter er det viktig at det er en tydelig gruppering i koden, for eksempel ved at menyer grupperes med punktlister. Da presenteres menylenkene som en samlet grupper selv for hjelpemidler for alvorlig synsskadde brukere. I menyer der undermenyer felles ut bør nøstede lister benyttes (lister i lister). Da blir selv den hierarkiske strukturen mulig å oppfatte for alvorlig synsskadde brukere.

I HTML5 finnes elementet nav som skal benyttes for å angi hva som er navigasjonen på siden. Elementet skal omfatte både hovedmenyen og undermenyen, eller så skal to ulike nav-elementer benyttes for å markere hovedmenyen og undermenyen hver for seg. Elementet kan også benyttes på en større meny i sidefoten.

```
<nav>... menylenker... </nav>
```

Landemerker

WAI-ARIA definerer ulike landemerker (eller ”landmarks” på engelsk), som også benyttes for å merke opp ulike typer innhold på en side. Disse tilfører ytterligere struktur for brukere som ikke ser grensesnittet.

Om du bygger med HTML5 skal du først og fremst benytte de strukturelementene som finnes i dennes standarden. Du kan også legge på WAI-ARIA sine landemerker, men unngå å legge role=”navigation” på elementet nav, ettersom det forårsaker problemer for enkelte skjermlesere.

Slik legger du inn landemerker med WAI-ARIA:

```
<div role=”search”>... søkefunksjonen ...</nav>
```

Benytt følgende landemerker der det er relevant:

role="search”, som markerer søkefunksjonen

role="main", som markerer hovedinnholdet på siden

role="complementary", som benyttes for innhold som komplementerer hovedinnholdet på siden. Dertil hører alle typer av ”relatert informasjon”, det vil si den typen innhold som typisk bruker å ligge i høyrespalten på siden.

role="banner”, som benyttes for sidehodet

role="contentinfo”, som benyttes for informasjon om innholdet, for eksempel copyright-informasjon og juridisk informasjon.

Det finnes flere landemerker:

role=”application”, som benyttes for områder med applikasjonslignende interaksjon, for eksempel en tekstediter. Unngå dette landemerket med mindre du har svært sterke argumenter for å benytte det.

role=”form”, unngå dette ettersom det ikke tilfører noe for brukerne per i dag.

Information om WAI-ARIA (W3C’s nettsted)

.

Koblede retningslinjer

WCAG 2.0 - 1.3.1 (A)

WCAG 2.1 - 1.3.1 (A)

NL190: Lenkers mål framgår tydelig i sin kontekst

− Ikke aktuelt

Bakgrunn

Tydelig formulerte lenketekster hjelper alle brukere å finne riktig lenke. En tydelig formulert lenke er også ofte en forutsetning for at alvorlig svaksynte brukere og en del brukere med kognitive funksjonsnedsettinger skal finne den informasjonen de søker. Lenker må derfor gi tydelig informasjon om hvilke sider de leder til og hva slags innhold de sidene har.

For en alvorlig svaksynt bruker er det umulig å få en visuell oversikt av siden. Det innebærer at det ikke er mulig å identifisere de lenker som finnes på siden og hvilke tekstavsnitt de er koblet til på en rask måte. Alvorlig svaksynte brukere får i stedet ta seg rundt på siden og se hva som finnes med støtte av sitt hjelpemiddel, en såkalt skjermlesere.

Skjermleseren presenterer bare den informasjonen brukeren har fokus på. Er fokus på en lenke, leses bare lenken opp, ikke teksten rundt. Brukeren kan ikke

visuelt koble lenken til noen tekst i nærheten. Det gjør at lenken i seg må gi all relevant informasjon om hvor den leder.

Mange hjelpemiddel har dessuten en funksjon som løfter lenkene ut av sin sammenheng og presenterer dem i en separat lenkeliste. Dette forenkler navigasjonen og gjør det mulig å finne spesifikke lenker som ”Logg inn” eller ”Handlevogn” raskere. Når en lenke tas ut av sin kontekst, hjelper det ikke brukeren om det ved siden av lenken står hvor den leder, så lenkens mål må framgå av selve lenketeksten. Lenker som heter ”Les mer” er derfor ikke tilstrekkelig.

Lenkede bilder må gi all relevant informasjon om lenken i alt-teksten (kalles også i blant for ”beskrivelse” eller ”tooltip” i ulike publiseringsverktøy).

For e-postlenker er det viktig at selve e-postadressen framgår i lenketeksten og også informasjon om mottakeren, for eksempel mottakerens funksjon og/eller navn. At e-postadressen framgår er viktig for at brukere som har lånt en datamaskin eller ikke vil sende e-post akkurat nå skal kunne kopiere adressen for å bruke den senere.

Title-text

Legg ikke noen title-tekst (kallas i blant for ”beskrivelse” eller ”tooltip”) på lenker. Lenketeksten skal gi brukeren all relevant informasjon om lenken. Tidligere har en del nettsteder komplettert lenketekstene med en title-tekst som vises når brukeren holder musepekeren over lenken. Denne løsningen formidler ikke den samme informasjonen til brukere som navigerer med tastaturet eller med en berøringsskjerm, bare til de som benytter mus.

Lenker er forståelige i kontekst

WCAG på nivå AA krever at lenker er forståelige i sin kontekst. Det innebærer at lenken sammen med teksten den ligger i, eller den forrige overskriften skal være forståelig. Har du en overskrift som angir ”Firmainformasjon” og under overskriften en lenke som heter ”Les mer”, så er lenken forståelig i sin kontekst. Det holder for å møte kravet i WCAG på nivå AA, og det holder for at dette punktet skal være godkjent. Men som vi har skrevet ovenfor er det ikke optimalt for brukerne, og kan gi problemer.

Koblede retningslinjer

WCAG 2.0 - 2.4.4 (A)

WCAG 2.1 - 2.4.4 (A)

NL360: Når samme meny eller lenkesamling vises på forskjellige sider skal sorteringen være konsekvent dersom det ikke vil bli oppfattet som ulogisk av brukere

Ingen feil funnet

Bakgrunn

Ikke endre rekkefølgen i lister av lenker, verktøy eller menyer på ulike sider. Samme meny eller lenkesamling må presenteres i samme rekkefølge på alle stedene hvis det ikke skal oppleves som ulogisk for brukeren.

Koblede retningslinjer

WCAG 2.0 - 3.2.3 (AA)

WCAG 2.1 - 3.2.3 (AA)

Farger og presentasjon

CP10: Teksters forgrunns- og bakgrunnsfarger gir til sammen tilstrekkelig kontrast

Ingen feil funnet

Bakgrunn

For at besøkeren enkelt skal kunne ta del i innholdet, er det viktig at kontrastene (spesifikt lyskontrast) mellom tekst og bakgrunn er bra. På en kontrastskala som går fra 1:1 (ingen forskjell) til 21:1 (svart mot hvitt) skal kontrasten være minst 4.5:1.

For å måle kontraster brukes en kompleks formel som finnes på W3C's nettsted:

Formel for kontraster (W3C's nettsted, engelsk)

Om du ikke vil sitte selv og regne på dette, anbefaler WAI blant annet verktøyet "Colour Contrast Analyser" som kan lastes ned kostnadsfritt fra Paciello Groups nettsted:

Colour Contrast Analyser (The Paciello Group nettsted, engelsk)

Koblede retningslinjer

WCAG 2.0 - 1.4.3 (AA)

WCAG 2.0 - 1.4.6 (AAA)

WCAG 2.1 - 1.4.3 (AA)

WCAG 2.1 - 1.4.6 (AAA)

CP20: Grafiske objekter som formidler informasjon, og tekster som forekommer i bilder, illustrasjoner og filmer har tilstrekkelig kontrast

Ingen feil funnet

Bakgrunn

Kontraster i illustrasjoner, diagrammer, bilder, ikoner og filmer skal være gode. Selv dårlige lysforhold eller moderat svekket syn skal ikke forhindre brukeren i å se og tolke den grafiske informasjonen. Forsøk derfor å skape så gode kontraster som mulig. Grenseverdien når det gjelder kontraster for materiale av denne typen skal være minst 3:1 i henhold til WCAG 2.1. Verdiene referer til fargen på det grafiske motivet mot bakgrunnen. Ett unntak for kravet er når motivet er supplert med tekst.

Koblede retningslinjer

WCAG 2.1 - 1.4.11 (AA)

CP50: All informasjon går å nå når brukeren har stillt inn større avstand mellom bokstaver, ord, rader eller avsnitt

Ingen feil funnet

Bakgrunn

For brukere med dysleksi og nedsatt syn er det viktig å kunne tilpasse hvordan tekstene vises. Det finnes ulike plugins for nettlesere, som gjør det mulig å styre størrelse, font og avstand mellom rader, ord, tegn og avsnitt. Endringer av avstand i tekst skal kunne gjøres uten at funksjoner blir utilgjengelige eller at innholdet skjules eller blir avskåret. Resultatet må ikke være perfekt, men endringer skal være mulige å utføre i brukerens nettleser, og skal ikke blokkeres. Brukeren skal kunne endre avstand i henhold til de målene som angis nedenfor.

- Linjehøyde (line height) satt til minst 1.5 ganger fontens størrelse
- Avstanden etter teksten er satt til minst 2 ganger fontens størrelse
- Avstanden mellom bokstaver (letter spacing) er satt til minst 0.12 ganger fontens størrelse
- Avstanden mellom ord (word spacing) er satt til minst 0.16 ganger fontens størrelse

Koblede retningslinjer

WCAG 2.1 - 1.4.12 (AA)

CP60: Nettstedet presenteres uten forstyrrende skjermflimmer, og uten bevegelige og blinkende elementer som ikke kan stoppes

Ingen feil funnet

Bakgrunn

På lignende måte som en urolig bakgrunn til tekst kan forstyrre konsentrasjonen kan også bevegelser i grensesnittet dra til seg besøkerens oppmerksomhet og gjøre det vanskeligere å konsentrere seg på teksten. Derfor bør du unngå bevegelser, blinking og flimmer. Om du absolutt vil ha dette, skal det enten stoppe automatisk innen 3 sekunder eller skal det finnes en tydelig måte å stoppe bevegelsen.

Vær oppmerksom på at blinking som er raskere enn 3 ganger per sekund aldri er tillatt.

Koblede retningslinjer

WCAG 2.0 - 2.2.2 (A)
WCAG 2.0 - 2.3.1 (A)
WCAG 2.0 - 2.3.2 (AAA)
WCAG 2.1 - 2.2.2 (A)
WCAG 2.1 - 2.3.1 (A)
WCAG 2.1 - 2.3.2 (AAA)

CP70: Sidene har unike og relevante sidetitler

Underkjent

Bakgrunn

Sidens tittel er det første som møter brukeren. Den brukes også som tekst når brukeren legger inn siden blant sine favoritter, og søkemotorer benytter sidetittelen når de indekserer sider. Det er derfor viktig at alle nettstedets sider har unike sidetitler som beskriver sidens innhold og funksjon på en tydelig måte.

Vi anbefaler at du først i tittelen angir sidens unike innhold, og at du deretter angir organisasjonens-/nettstedets navn. Dette gjør du med elementet title, som alltid skal stå inne i elementet head:

```
<head>
...
```

```
<title>Forskning og innovasjon - Våre oppdrag - Funka</title>
...
</head>
```

Husk at også sider som presenteres inne i en iframe skal ha slik tittel, blant annet fordi enkelte skjermlesere presenterer denne i stedet for title-attributtet på den aktuelle iframe.

Kommentar

Ingen overskrifter funnet i applikasjonen

Anbefalinger

Forsikre at hver side har en overskrift, vi anbefaler at "Smittestopp" er overskriften på hovedsiden. Nå ser "Smittestopp" ut som en overskrift men er i realtiten kun ett bilde av logoen og teksten uten noen form for alternativ tekst.

Koblede retningslinjer

WCAG 2.0 - 2.4.2 (A)
WCAG 2.1 - 2.4.2 (A)

CP110: Sidens hovedspråk er angitt i koden

× Underkjent

Bakgrunn

Hovedspråk skal angis i koden på samtlige sider. Dette gjør det mulig for opplesende hjelpemiddel å avgjøre på hvilket språk teksten er skrevet, og dermed velge riktig stemme for det aktuelle språket. Hovedspråk skal angis direkte i elementet HTML, øverst i sidens kode.
I HTML5 angis hovedspråk med lang-attributtet:

```
<HTML lang=”no”>
```

I XHTML angis hovedspråk med attributtet xml:lang. Dette skal dog kompletteres med HTML-måten å skrive for bakoverkompatibilitet (unntatt i XHTML 1.1).
Eksempel:

```
<HTML xml:lang=”no” lang=”no”>
```

Norske talesynteser er normalt satt for å takle både bokmål og nynorsk uten nærmere språkdefinering. Du kan likevel vurdere å spesifisere nynorsk eller

bokmål, og da kan det samtidig være relevant å supplere med landskoden for å tydeliggjøre dette.

Eksempel for spesifisering av nynorsk:

```
<HTML lang=”nn-NO”>
```

Kommentar

Appen annonserer ikke sitt eget språk i kode, vår VoiceOver testing viste at VoiceOver for iOS ikke klarte å automatisk kjenne igjen hvilket språk appen benyttet.

Koblede retningslinjer

WCAG 2.0 - 3.1.1 (A)

WCAG 2.1 - 3.1.1 (A)

CP120: Språk som avviker fra sidens hovedspråk er spesifisert i koden

− Ikke aktuelt

Bakgrunn

Når språkendringer gjøres i innholdet skal disse markeres. Det kan for eksempel være om det finnes et sitat på siden som er på et avvikende språk, eller om siden gir informasjon på ett språk mens menyene og nettstedets struktur er på et annet språk. Slike endringer skal angis i koden. Dette kan gjøres på de fleste element, for eksempel span, li, div, a og p. Enkeltstående termer og ord behøver ikke merkes opp.

I HTML5 angis avvikende språk med lang-attributtet:

```
<p lang="en">
```

Dette angir at det aktuelle avsnittet er skrevet på engelsk. Attributtet kan brukes på alle HTML-element i HTML5.

I XHTML angis språk med attributtet xml:lang. Dette skal dog kompletteres med HTML-måten å skrive for bakoverkompatibilitet (unntatt i XHTML 1.1). For eksempel slik:

```
<p xml:lang="en" lang="en">
```

Kommentar

Ingen tilfeller av andre språk brukt funnet utenom i developer tools som ikke skal med til produksjonsversjonen.

Koblede retningslinjer

WCAG 2.0 - 3.1.2 (AA)

WCAG 2.1 - 3.1.2 (AA)

Struktur og formatering

ST20: Områdeoverskrifter beskriver områdets funksjon klart og konsistent

Ingen feil funnet

Bakgrunn

Akkurat som med overskrifter i innholdet på siden, er det viktig at områdeoverskriftene hjelper brukerne. En områdeoverskrift skal kort og så spesifikt som mulig gjengi hva området inneholder. Bruk konkrete overskrifter som "E-tjenester", "Brosjyrer og fordypning" eller "Kontakt".

Anbefalinger

Forsikre at alle overskrifter som visuelt fungerer som overskrifter også har ett tekslig ekvivalent.

Koblede retningslinjer

WCAG 2.0 - 2.4.6 (AA)

WCAG 2.1 - 2.4.6 (AA)

ST40: Innholdsoverskrifter har dekning i teksten og tiltrekker lesing

Ingen feil funnet

Bakgrunn

Overskriften er det første som leseren leser. Det kan også bli det eneste som leseren leser, dersom den ikke hører sammen med teksten på en korrekt måte eller leder til "feil" innhold.

Derfor er det viktig at overskriftene gir relevant informasjon om den tilhørende teksten. Overskriftene bør inneholde det eller de viktigste nøkkelordene i teksten. De skal også være innbydende, slik at de lokker leseren til å lese mer.

Koblede retningslinjer

WCAG 2.0 - 2.4.6 (AA)

WCAG 2.1 - 2.4.6 (AA)

ST60: Tekst som visuelt fungerer som overskrifter er også kodet som overskrifter

× Underkjent

Bakgrunn

Overskrifter er noe av det første som brukeren ser og leser på en ny side. God bruk av overskrifter er derfor svært viktig for å nå ut med informasjonen. Men alle brukere kan ikke se overskriftene, en del er avhengig av et hjelpemiddel for å ta til seg informasjonen. For at hjelpemiddelet skal forstå hva som er overskrifter på en side, og formidle dette til brukeren, kreves det at de er korrekt kodet med h-element.

Det er flere ting som er viktige, men en grunnforutsetning er at det som ser ut som, og derfor visuelt fungerer som, overskrifter også er kodet med korrekte overskriftselement.

I HTML kodes overskrifter med elementene h1, h2, h3, h4, h5 og h6. Sifrene indikerer nivåene. Det er også mulig å kode overskrifter med WAI-ARIA. Da gir man teksten role=”heading” og angir nivå med attributtet aria-level. Vi fraråder fra dette ettersom det ikke er full støtte for dette blant hjelpemidlene, og det øker risikoen for problemer.

Det finnes et unntak til dette punktet. Hvis du har overskrifter som vises så fort siden lastes, og som ligger før innholdets hovedoverskrift i strukturen, så skal de ikke kodes som overskrifter. Eksempel på dette er en visuell overskrift ovenfor venstremenyen på et nettsted der navigasjonen kommer før innholdet i strukturen. Den skal ikke kodes som en overskrift. Derimot, om du har en utfoldbar meny som bare vises når brukeren klikker på en knapp eller lenke, og den har visuelle overskrifter, da skal de være kodet som ordentlige overskrifter.

Kommentar

Appen har overskrifter på alle sidene, men på forsiden fungerer overskrift kun visuelt som en overskrift. Den er del av ett bilde og derfor ikke mulig å navigere til med skjermleser.

Anbefalinger

Forsikre at alle overskrifter enten er ren tekst eller har en tekslig ekvivalent som kan nås av skjermlesere som om det er ett vanlig elemenet i appen.

Den lettest løsningen kan være å skille smittestopp ikonet fra overskriften, og la overskriften fungere som ett vanlig tekstlig element mens bildet kune inneholder logoen.

Koblede retningslinjer

WCAG 2.0 - 1.3.1 (A)

WCAG 2.1 - 1.3.1 (A)

ST80: Overskriftsstrukturen innledes med en hovedoverskrift og hopper ikke over nivåer

Ikke aktuelt

Bakgrunn

Overskriften skal forme en korrekt struktur. Det innebærer at første overskriften skal være kodet som en h1-overskrift, som kan følges av en annen h1-overskrift, eller en h2-overskrift. En h2-overskrift kan etterfølges av h1, h2 eller h3 og så videre.

Eksempel:

```
<h1>Hovedoverskrift</h1>
   <h2>Overskrift nivå 2</h2>
     <h3>Overskrift nivå 3</h3>
     <h3>Overskrift nivå 3</h3>
   <h2>Overskrift nivå 2</h2>
```

Du får ikke hoppe over nivåer (gå fra h1 til h3 for eksempel), det fører til at brukeren ikke stoler på strukturen og det blir vanskelig å forstå hvordan informasjonen henger sammen.

Koblede retningslinjer

WCAG 2.0 - 1.3.1 (A)

WCAG 2.1 - 1.3.1 (A)

ST90: Overskriftstrukturen er logisk og gjenspeiler innholdets hierarki

Ikke aktuelt

Bakgrunn

Overskriftsstrukturen skal være logisk og speile strukturen i sidens innhold. Innled alle sider med en korrekt hovedoverskrift (element h1) øverst i innholdet. Deretter skal alle overskrifter komme i en hierarkisk korrekt ordning, som gjenspeiler innholdets struktur. Tenk på at tallet ikke skal gjenspeile hvor viktig tittelen er, men hvordan den forholder seg til andre overskrifter. Er det en mellomtittel til forrige overskrift, eller bør den være på samme nivå?

Forrige punkt handler om at strukturen er formelt korrekt. Dette punktet handler om at det er en struktur som faktisk matcher innholdet på siden. Dette innebærer i utgangspunktet at sidens spesifikke innhold bare skal ha én h1, med unntak av eventuelt innhold som er lagt inn med iframe. Det er likevel tillatt med flere h1 på siden, for eksempel for å skape et tydelig skille mellom sidens spesifikke innhold og det generelle innholdet som følger etter, ofte kalt sidefot eller kolofon.

Koblede retningslinjer

WCAG 2.0 - 1.3.1 (A)

WCAG 2.1 - 1.3.1 (A)

ST120: Lister er korrekt kodet og brukes på rett måte

Ikke aktuelt

Bakgrunn

En del informasjon passer å legge inn i lister. Det gjelder for eksempel oppramsinger, definisjonslister og beskrivelser over ulike steg i en prosess.

For å legge inn disse lister skal de element som finnes i HTML for dette formålet brukes. Dette er en forutsetning for at et hjelpemiddel skal tolke og presentere dem som korrekte lister for brukeren.

For å innlede en punktliste brukes elementet ul, for at innlede en nummerert liste brukes elementet ol. Hvert listeelement skapes deretter med elementet li.

For lister hvor hvert av punktene har to ledd (på engelsk *description list*) brukes elementet dl for å innlede listen, dt for å angi punktets beskrivelse og dd for å gi punktets verdi.

Eksempel:

```
<dl>
  <dt>Søknadsfrist:</dt>
  <dd>1. september 2017</dd>
  <dt>Arbeidssted:</dt>
  <dd>Grenseveien 12, Oslo</dd>
  <dt>Omfang:</dt>
  <dd>Heltid</dd>
  <dt>Varighet:</dt>
  <dd>1 års vikariat</dd>
</dl>
```

Lister kan også brukes for å strukturere menyen. Ved å legge inn menyen i en nøstet liste kan du formidle menyens hierarki på en enkel måte også for brukere som ikke kan se den.

Eksempel:

```
<ul>
  <li><a href="bs.hmtl">Barn & skola</a></li>
  <li><a href="bb.html">Bo & bygga</a>
  <ul>
  <li><a href="blov.html">Bygglov</a></li>
  <li><a href="proj.html">Aktuella byggprosjekt</a></li>
  </ul>
  </li>
  <li><a href="tur.html">Turism</a></li>
 </ul>
```

Kommentar

Ingen lister funnet

Koblede retningslinjer

WCAG 2.0 - 1.3.1 (A)

WCAG 2.1 - 1.3.1 (A)

ST180: Avsnitt er laget med elementet p og brukes på rett måte

× Underkjent

Bakgrunn

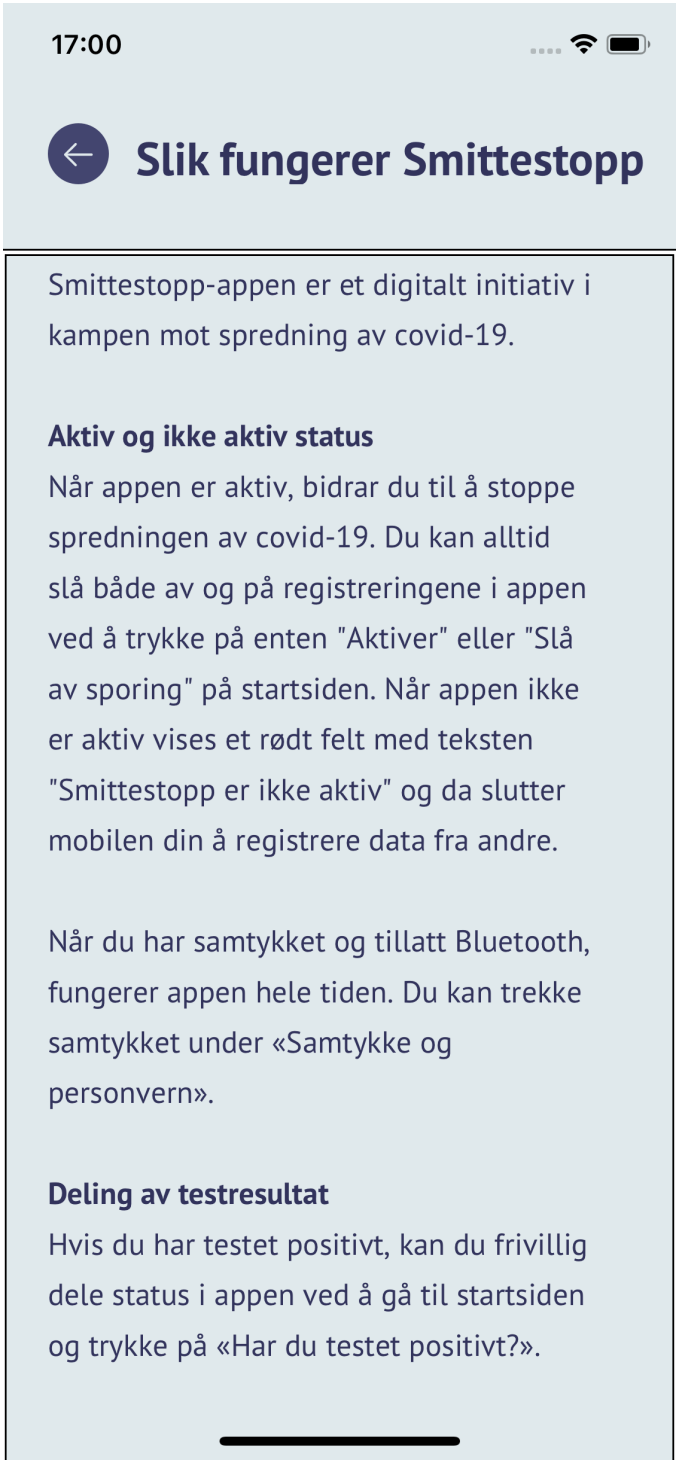
For å legge inn tekstavsnitt på en side skal elementet p brukes. Dette gjør dokumentet lettere å skumlese med et opplesende hjelpemiddel. Brukeren kan da i mange av disse hoppe fra avsnitt til avsnitt. Dette er ekstra viktig i lovtekster der de ulike avsnittene må kunne skilles fra hverandre.

Bruk ikke elementet div, br eller hr for å skille tekstavsnitt.

Kommentar

Siden "Slik fungerer Smittestopp" har visuelt overskrifter og avsnitt, men leses opp som ett kontinuerlig tekstelement av VoiceOver.

Dette gjør det vanskelig for skjermlesere å navigere imellom tekst, vil de ha noe annonsert igjen av VoiceOver må de starte på toppen av hele tekstbolken igjen istedenfor å kunne navigere en opp til forrige paragraf. Det er heller ikke mulig for skjermleserbrukere å raskt navigere mellom paragrafene slik seende brukere gjerne gjør fordi alt er i en tekstbolk.



Anbefalinger

Forsikre at all tekst er programmatisk delt opp slik den er visuelt delt opp.

Koblede retningslinjer

WCAG 2.0 - 1.3.1 (A)

WCAG 2.1 - 1.3.1 (A)

ST200: Sitat er markert med q eller blockquote

Ikke aktuelt

Bakgrunn

Merk opp lengre sitat med elementet blockquote.

Visuelt skaper elementet ett innrykk i teksten, noe som en del publiseringssystem utnytter for å tilby en enkel funksjon for innrykk i teksteditoren. Dette skal unngås. Blockquote får bare brukes for sitat.

Bruk elementet q for at merke opp og formatere kortere sitat.

I både blockquote og q skal attributtet cite brukes for å angi en kilde i form av en url om det finnes en slik kilde.

Eksempel:

```
<blockquote cite=”http://www.yr.no/vaer/prognose241215.html”>

 <p>Det finnes en 50-50 sjanse for snø i Norge til jul. Det bygger dog på at lavtrykket som nå ligger over de britiske øyene styrer nordover.</p>

</blockquote>
```

Koblede retningslinjer

WCAG 2.0 - 1.3.1 (A)

WCAG 2.1 - 1.3.1 (A)

Bilder

IM10: Tekst presenteres som tekst, ikke som bilder av tekst

× Underkjent

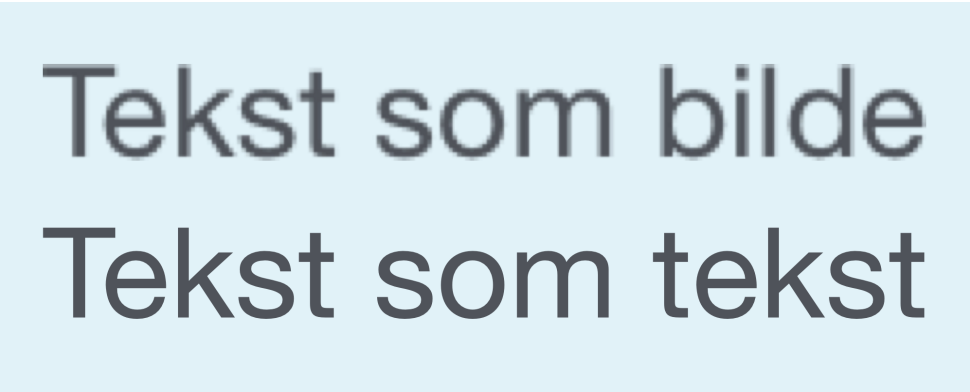
Bakgrunn

Bilder er på mange måte en bedre måte å kommunisere informasjon med enn vanlig tekst. Med bilder kan man forenkle og tydeliggjøre komplekse sammenhenger og vanskelige begrep. Bilder bistår også i lesingen ved å bryte opp teksten, og får brukeren til å oppleve teksten som mindre intensiv og krevende.

Bilder av tekst

Bilder skal altså brukes der det er gunstig, men de skal ikke brukes for å representere tekst. Skal tekst legges ut i innholdet, menyene eller på andre steder på sidene, skal teksten legges ut som ren tekst. Bilder av tekst skaper mange ganger store problem for brukerne.

Når bilder forstørres, blir hvert billedpunkt større. Tekst er derimot vektorgrafikk, noe som innebærer at hvert tegn får flere billedpunkt når teksten forstørres. Forskjellen blir da at bildet av tekst blir kantete og ofte oppleves som uklar, mens vanlig tekst forblir skarp. Dette blir ekstra tydelig i forstørrende hjelpemiddel. Under har vi lagt inn to bilder, først et bilde med tekst som pikselgrafikk, og deretter et bilde av faktisk tekst som altså er vektorgrafikk. I begge tilfellene er teksten 12-punkts skrift som er forstørret til 300 %.



Et annet problem med bilder av tekst er at teksten i et bilde ikke kan tilpasses av hjelpemiddel for dyslektikere, og den kan heller ikke kopieres, som en del hjelpemiddel for denne gruppen krever. Selv om man setter en alt-tekst på bildet av tekst er det ikke sikkert at et hjelpemiddel for dyslektikere kan håndtere dette.

Unntak

I visse situasjoner kan det være berettiget å bruke tekst i et bilde. Det kan for eksempel gjelde for organisasjonsskjema eller ulike typer av flytskjema der teksten må være en del av bildet.

Når tekst brukes i slike bilder, må den gjøres minst dobbelt så stor som den vanlige teksten på nettstedet, fordi teksten i bildet ikke kan forstørres på en bra måte av nettlesere og hjelpemiddel. Samtidig må kontrastene være bra, og bildet må inneholde en alt-tekst.

Om ikke beskrivelsen i alt-teksten kan inneholde all informasjon som bildet viser, må det enten finnes en utførlig beskrivelse som legges inn via attributtet longdesc eller en utførlig beskrivelse i tilknytning til bildet. Om det for eksempel er et organisasjonsskjema kan organisasjonen med dens ulike avdelinger beskrives i tekst under bildet. Bruk overskrifter og lister for å strukturere informasjonen på en så tydelig måte som mulig.

Kommentar

Overskriften i appen er i ett bilde istedenfor som ett tekstlig element. Dette fører til at skjermlesere brukere kun kommer til "er aktiv" eller "er ikke aktiv" .

"Er aktiv" i skjermbildet under er første element etter menyen, skjermleserbrukere får altså lest opp "er aktiv" uten no som helst kontekst om hva som er aktivt.

Alle sidene har også "FHI" logoen, dette er ett element det ikke er mulig å navigere til med skjermleser. Skjermleserbrukere får derfor ikke, ihvertfall fra forsiden, informasjonen seende brukere får om at dette er en app utgitt av FHI.

Anbefalinger

Forsikre at all tekst i bilde også har ett tekstlig ekvivalent - eller ikke bruk tekst i bilder.

Koblede retningslinjer

WCAG 2.0 - 1.4.5 (AA)

WCAG 2.0 - 1.4.9 (AAA)

WCAG 2.1 - 1.4.5 (AA)

WCAG 2.1 - 1.4.9 (AAA)

IM30: Likeverdige tekstbeskrivelser finnes for alle meningsbærende grafiske element på nettstedet

× Underkjent

Bakgrunn

For at hjelpemiddel skal gjengi innholdet i bilder til brukeren, må bildene utstyres med en alternativ beskrivelse. Dette gjøres vanligvis med hjelp av attributtet alt (kan i ulike publiseringssystem kalles for tooltip eller beskrivelse). Den alternative beskrivelsen skal kortfattet redegjøre for bildets motiv. Jo viktigere bildet er for siden og forståelsen, desto mer detaljert må beskrivelsen være.

Alt-teksten skal holdes under 150 tegn, men i blant rekker ikke det for å gjengi all informasjon i bildet. Da må alt-teksten kort angi hva bildet viser, og samtidig henvise til en lengre beskrivelse eller til tilsvarende informasjon. For eksempel:

"*Diagram over befolkningsutviklingen i Norge. Se tabell etter bildet*".

Meningsbærende grafiske objekt er i prinsipp alle bilder som inneholder informasjon, illustrasjoner, fotografier, ikoner, bildeknapper, bildelenker og så videre. Ikke meningsbærende grafiske objekt er for eksempel strek, tomrom, fargede bakgrunner og lignende. Slike bilder skal helst legges inn med CSS.

Lenkede bilder krevet noe ekstra ettertanke. Hvis du har en lenke som bare består av et bilde, så må bildets alt-tekst først og fremst fortelle lenkens mål. Hvis du har en lenke som både inneholder et bilde og en tekst må du tenke på at lenkebeskrivelsen kommer til å bestå av lenketeksten og bildets alt-tekst. Avhengig av situasjonen kan du da behøve å ha en tom alt-tekst. Det er for eksempel bare irriterende å høre: Skriver Skriv ut, om du har en skriv ut lenke med ikon som har alt-tekst.

Selv om du har skjema og bruker input type=”image” så skal denne ha en alt-tekst som forklarer objektets funksjon. Notér at det oftest er bedre å bruke input-varianter som er tilpasset formålet, for eksempel er det bedre å bruke input type=”submit” eller et button-element for knapper enn et bilde.

Bruker du elementene figure og figcaption så skal bildet fortsatt ha en alt-tekst, eksempel:

```
<figure>
    <img … alt=”Sverige år 1790” />
    <figcaption>
```

```
      Figur 1 – Bilden visar Sveriges utbredning 1790...
   </figcaption>
</figure>
```

Kommentar

Se IM10

Koblede retningslinjer

WCAG 2.0 - 1.1.1 (A)

WCAG 2.1 - 1.1.1 (A)

IM40: Alt-teksten beskriver bildet eller dets formål

× Underkjent

Bakgrunn

Alle meningsbærende grafiske objekter skal inneholde en kort alternativ beskrivelse (alt-tekst) som tilsvarer bildets funksjon i dokumentet. Alternative beskrivelser skal:

- Beskrive bildets motiv
- Beskrive bildets funksjon (hvis det er klikkbart)

Bilder som hovedsakelig har et dekorativt formål trenger kun en kort beskrivelse. Bilder som formidler viktig informasjon må ha en mer detaljert beskrivelse.

Det er ofte gunstig å beskrive hva slags grafisk objekt som brukes. For eksempel: ”Joachim Henstad fra Funka Nu AS, foto” eller ”Mann med hammer, illustrasjon”.

Grafiske objekt som ikke er meningsbærende (for eksempel layout-elementer) bør ikke beskrives. Derimot er det viktig å kvalitetssikre at slike objekt har et tomt alt-attributt.

Kommentar

I standard rotor-navigasjon med iOS kan man ikke navigere seg mellom bilder, men man kan gjøre det om man setter det på via rotor instillinger.

Med de rette instillingene kan man navigere mellom bilder og få deres label eller accessibilityLabel annonsert av skjermleseren. FHI logoen i appen har ingen label og ble ikke kjent igjen som ett bilde av VoiceOver i vår testing. Det er derfor helt umulig for skjermleserbrukere å få den samme informasjonen om at "FHI" står bak denne appen som andre brukere får.

Ingen av bildene på forsiden, altså smittestopp og FHI-logo bildene ble gjenkjent som bilder eller elementer med accessibilityLabel i vår testing

Anbefalinger

Forsikre at bilder er kodet som bilder og inneholder en alternativ tekstlig beskrivelse i form av accessibilityLabel eller liknende.

Koblede retningslinjer

WCAG 2.0 - 1.1.1 (A)

WCAG 2.1 - 1.1.1 (A)

IM60: Klientbaserte bildekart har tekstbeskrivelser både for lenker og for bilder

Ikke aktuelt

Bakgrunn

Bildekart (eng. Imagemap) er bilder med klikkbare områder. Det finnes to typer bildekart, klientbasert og serverbaserte.

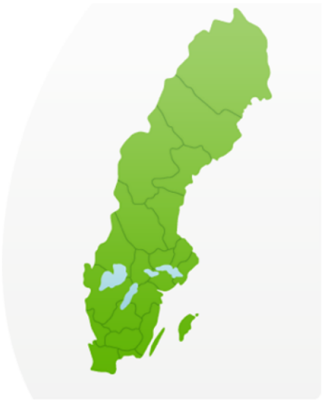
Klientbaserte bildekart er i bunn og grunn bare et vanlig bilde der flatene i bildet defineres som lenkede områder. Et serverbasert bildekart er et bilde der du kan klikke hvor som helst i bildet og siden sender koordinatene til serveren, som deretter bestemmer hva som skal sendes tilbake til nettleseren. Dette brukes for for eksempel for mange karttjenester der det går an å zoome, slik som Google Maps og Eniro.

For brukere med hjelpemidler er klientbaserte bildekart et mindre problem enn servicebaserte bildekart. Det er en mye enklere teknikk, og det er relativt enkelt å sørge for at alle områdene får en passende beskrivelse.

Tilgjengelige, klientbaserte bildekart

Det bildet som brukes som bildekart skal ha en kort beskrivelse av hva bildet viser, akkurat som hvilket som helst meningsbærende bilde.

De ulike klikkbare områdene (area-elementene) skal beskrives i attributtet title, men kan i tillegg også beskrives i alt-attributtet. Det er også mulig å legge vanlige tekstlenker til samme sidene som lenkene i bildet leder til, i tillegg til tilkoblingen til bildet. Avhengig av hvor små de klikkbare områdene er kan det bli nødvendig med en liste med lenker parallelt til bildekartet. Slik kan det se ut:



I eksempelet kan brukeren klikke på fylket i listen, eller i kartbildet til høyre.

Slik kan koden for en enkel klientbasert kart se ut. Her har vi valgt både å bruke

alt-tekst og title-tekst i de klikkbare områdene, ettersom støtten for begge attributtene varierer i elementet area:

```
<img src="Kart.gif" usemap="#Kart" alt="Norgeskart">

<map id="Kart" name="Kart">

<area shape="poly"
coords="125,304,125,194,168,81,264,2,322,3,338,30,330,67,198,
163,181,288" href="#Nord-Norge" alt="Se prosjekt i Nord-
Norge" title="Se prosjekt i Nord-Norge">

<area shape="poly" coords="125,304,36,430,166,400,181,288"
href="#Midt-Norge" alt="Se prosjekt i Midt-Norge" title="Se
prosjekt i Midt-Norge">

<area shape="poly"
coords="36,430,83,420,105,461,95,512,139,546,118,577,84,577,5
1,546" href="#Vest-_og_Sorlandet" alt="Se prosjekt på Vest-
og Sørlandet" title="Se prosjekt på Vest- og Sørlandet">

<area shape="poly"
coords="83,420,166,400,191,481,184,529,139,546,95,512,105,461
" href="#Ostlandet" alt="Se prosjekt på Østlandet" title="Se
prosjekt på Østlandet"></map>
```

Koblede retningslinjer

WCAG 2.0 - 1.1.1 (A)
WCAG 2.0 - 2.4.4 (A)
WCAG 2.0 - 2.4.9 (AAA)
WCAG 2.1 - 1.1.1 (A)
WCAG 2.1 - 2.4.4 (A)
WCAG 2.1 - 2.4.9 (AAA)

IM70: Serverbaserte bildekart brukes kun når det øker brukeropplevelsen, og da tilbys alternative veier for å få frem den samme informasjonen på

Ikke aktuelt

Bakgrunn

Serverbaserte bildekart kan gi en betydelig merverdi for brukerne. Det er for eksempel mye enklere å zoome ut og inn i kart hvor en adresse er markert, og dermed orientere seg. Derfor anbefaler vi at du bruker serverbaserte bildekart når det finnes en merverdi for brukeren. Hva du bør tenke på er følgende:

- Det må være mulig å håndtere kartet med tastaturet.
- Det skal finnes en alternativ måte å finne eller benytte informasjonen som kartet formidler. Hvis det er et kart som viser hvor et kontor ligger, skal adressen til kontoret også finnes i tekstform. Er det et kart med mange ulike interessepunkter må det finnes en mulighet å søke frem et interessepunkt og å liste interessepunkter i en tekstbasert liste.
- Tilgjengeligheten i kartet må ses over. Det er vanskelig å unngå at brukernes evner til å oppfatte farger i et kart påvirker opplevelsen, men med et gjennomtenkt bruk av ikoner, farger og symboler kan kartene gjøres ganske tydelige for så mange som mulig.
- Ha en gjennomtenkt strategi for hvordan kartet skal fungere i mobile enheter. Brukere som scroller på berøringsskjerm skal ikke havne i en uendelig scrollbar fordi kartet tar hele skjermen. Det må gå å skrolle nedover kartet på siden (om ikke kartet utgjør hele siden).

Koblede retningslinjer

WCAG 2.0 - 1.1.1 (A)

WCAG 2.1 - 1.1.1 (A)

Skjema

FS10: Skjemaobjekt er kodet med korrekt skjemakode

Ingen feil funnet

Bakgrunn

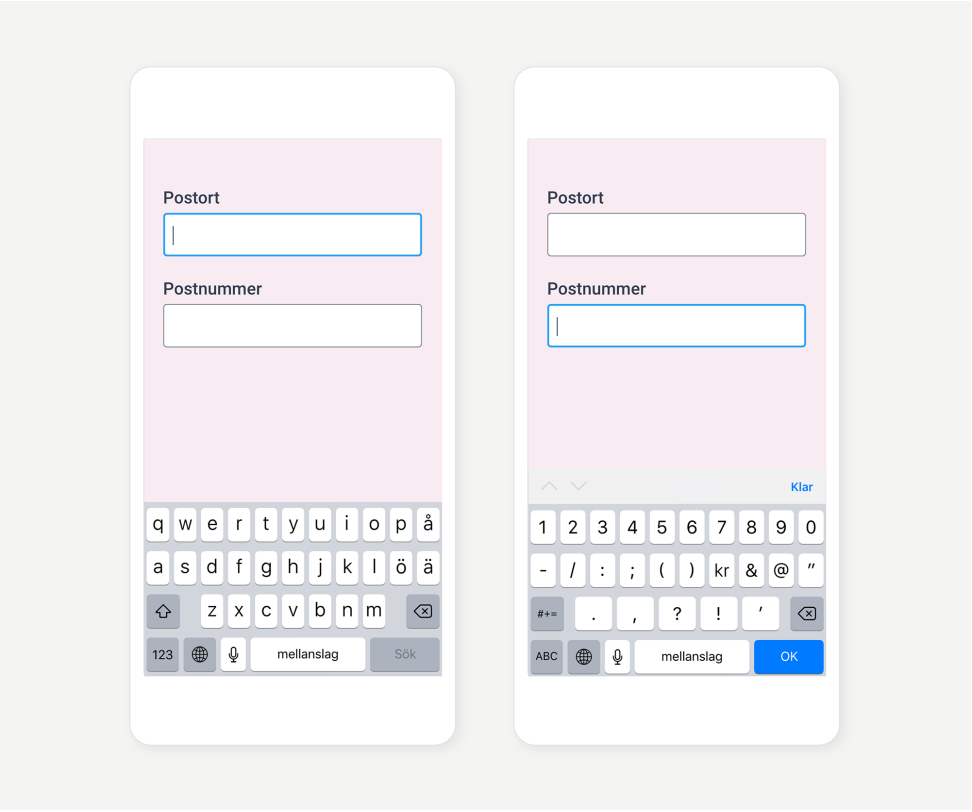
Det er viktig at alle deler av et skjema er lagt inn som faktiske skjemaobjekt, da de ellers ikke blir presentert sammen med øvrige deler av skjemaet i et hjelpemiddel. Iblant forekommer for eksempel knapper som er laget som lenkede bilder i stedet for som korrekte skjemaknapper. Da listes ikke disse blant øvrige skjemaobjekt, og det blir unødvendig vanskelig å sende skjemaet for en del brukere.

Det er også viktig at riktig type element brukes. Hvis for eksempel bare ett valg er tillatt, skal du bruke radioknapper, og ikke legge inn avkrysningsruter sammen med et script som hindrer at mer enn ett valg er avkrysset.

I HTML5 finnes flere nye typer skjemaobjekt. Disse har mange fordeler for brukerne, ikke minst i mobilen der tastaturet kan tilpasse seg etter den type informasjon som skal legges inn. Under et eksempel på hvordan ulike typer skjemafelt henter frem ulike tastatur i mobilen.

```
<label for="epost">Epostadresse</label>
 <input type="email" name="epost" id="epost">
```

```
<label for="mobil">Mobilnummer</label>
<input type="tel" name="mobil" id="mobil">
```



Koblede retningslinjer

WCAG 2.0 - 1.3.1 (A)

WCAG 2.0 - 4.1.2 (A)

WCAG 2.1 - 1.3.1 (A)

WCAG 2.1 - 4.1.2 (A)

FS50: Det finnes ledetekster og instruksjoner som beskriver for brukeren hvordan og hva de skal fylle i skjemaet

Ingen feil funnet

Bakgrunn

Når brukeren skal velge eller legge inn informasjon må det finnes ledetekster og instruksjoner som tydelig angir hva som skal velges eller legges inn. Instruksjonene skal også inneholde beskrivelse om hvilket format informasjonen skal angis i, og annen viktig informasjon som brukeren trenger for å kunne legge inn informasjonen på riktig måte.

Koblede retningslinjer

WCAG 2.0 - 3.3.2 (A)

WCAG 2.1 - 3.3.2 (A)

Elmer 3 - 1.2 Veiledning

Elmer 3 - 3.3 Utforming av spørsmål

Elmer 3 - 3.6 Frivillige felter

FS60: Ledetekster er knyttet til respektive skjemaobjekt

Ikke aktuelt

Bakgrunn

Ledeteksten er den teksten som forteller brukeren hva som skal fylles inn i et visst felt, hva en avkrysningsrute eller radioknapp har slags betydning, eller hva brukeren skal velge i en liste. Disse ledetekstene er ofte visuelt tydelig plassert nærme skjemaobjektet (tekstfeltet/avkrysningsruten/radioknappen/listen/knappen) det gjelder, noe som gjør at brukere som ser siden og kan lese teksten blir tydelig informert om formålet med skjemaobjektet.

For at dette skal fungere også for brukere som benytter ulike hjelpemiddel for å ta til seg informasjonen, må det finnes en tydelig kobling i koden mellom ledeteksten og skjemaobjektet. Dette gjennomføres med elementet label. Benytt label-element for å legge inn samtlige ledetekster. Label-elementet må også kobles korrekt til respektive skjemaobjekt med attributtet for som skal peke på skjemaobjektets id.

Label-elementet skal ikke bare inkludere ledeteksten men også eventuell tydeliggjørende informasjon og eventuelle feilmeldinger som er koblet til skjemaobjektet. Husk at det bare er tillatt med én label per skjemaobjekt.

Om grensesnittet kodes med HTML5, skal skjemafeltet være merket med hvilken type informasjon som er forventet.

Eksempel på skjema:

```
<form action="http://www.funka.com/survey" method="post">

 <label for="namn">Namn</label>
 <input type="text" name="namn" id="namn" />

 <label for="country">Land</label>
 <select name="selectCountry" id="country">
   <option disabled>Välj</option>
   <option>Norge</option>
   <option>Sverige</option>
 </select>
```

```
<fieldset>
  <legend class="hidden">Godkänn</legend>
  <label for="yes">
  <input type="radio" name="accept" id="yes">Ja, jag
godkänner</label>
  <label for="no">
  <input type="radio" name="accept" id="no">Nej, jag
godkänner inte</label>
</fieldset>

<label for="comments">Kommentar</label>
  <textarea name="comments" id="comments">
  </textarea>
  <input type="submit" value="Skicka svar" />
</form>
```

Koden genererer følgende skjema:

Namn

Land

Välj

Ja, jag godkänner

Nej, jag godkänner inte

Kommentar

Skicka svar

Koblede retningslinjer

WCAG 2.0 - 1.3.1 (A)
WCAG 2.0 - 2.4.6 (AA)
WCAG 2.0 - 3.3.2 (A)
WCAG 2.0 - 4.1.2 (A)
WCAG 2.1 - 1.3.1 (A)
WCAG 2.1 - 2.4.6 (AA)
WCAG 2.1 - 3.3.2 (A)
WCAG 2.1 - 4.1.2 (A)

FS70: Inngangsfelt i skjemaer kan identifiseres ved korrekt oppmerking i autocomplete attributtet

! Ingen feil funnet men kan forbedres

Bakgrunn

Lange og komplekse skjemaer utgjør ofte et stort problem for mange brukere, ikke minst brukere med kognitive funksjonsnedsettelser. Jo enklere skjemaet er, og jo mindre brukeren trenger å fylle i, desto mindre er risikoen for problemer. Det er mange grupper som har ekstra store problemer med skjemaer. Utenom brukere med kognitive funksjonsnedsettelser utgjør skjemaer også problemer for personer med skrivevansker og brukere med sterkt nedsatt syn.

Attributtet autocomplete gjør det mulig å forenkle utfylling i skjemafelt som inneholder ofte tilbakevendende informasjon, som navn, e-postadresse og telefonnummer. Brukeren trenger kun fylle i informasjonen ved ett tilfelle, og kan la nettleseren lagre informasjonen til neste gang brukeren skal fylle i samme informasjon i andre skjemafelt. Autocomplete brukes på skjemafelt av ulike typer, som for eksempel text, email, number, password.

```
<form action="/abonnere.html">
Fornavn:<input type="text" name="fnavn" autocomplete="given-
name" />
Etternavn: <input type="text" name="enavn"
autocomplete="family-name" />
E-post: <input type="email" name="epost" autocomplete="email"
/>
<input type="submit" value="Abonner"
</form>
```

Det finnes over 40 mulige verdier å bruke i attributtet autocomplete. I visse situasjoner er det ikke ønskelig at nettleseren fyller i skjemafeltet. Eksempler på slike situasjoner er når brukeren skal angi en engangskode ved innlogging til en e-tjeneste eller engangspassord som sendes når et passord må fornyes. Da kan den automatiske kompletteringen skrus av gjennom å sette attributtet autocomplete til off på det spesifikke skjemafeltet.

Hjelpemidler for personer med kognitive funksjonsnedsettelser kan fange opp attributtet autocomplete i skjemafeltet og gi brukeren en enklere og mer lettforståelig beskrivelse av den informasjonen som skjemafeltet skal inneholde.

Et eksempel er når autocomplete settes til verdien ”tel”, som erstatter eller kompletterer skjemafeltets ledetekst med et telefon-ikon.

For å møte kravet på dette punktet skal skjemafeltet som finnes i WCAG sin liste "Input Purposes for User Interface Components" ha attributtet autocomplete satt til de verdiene som angis i listen.

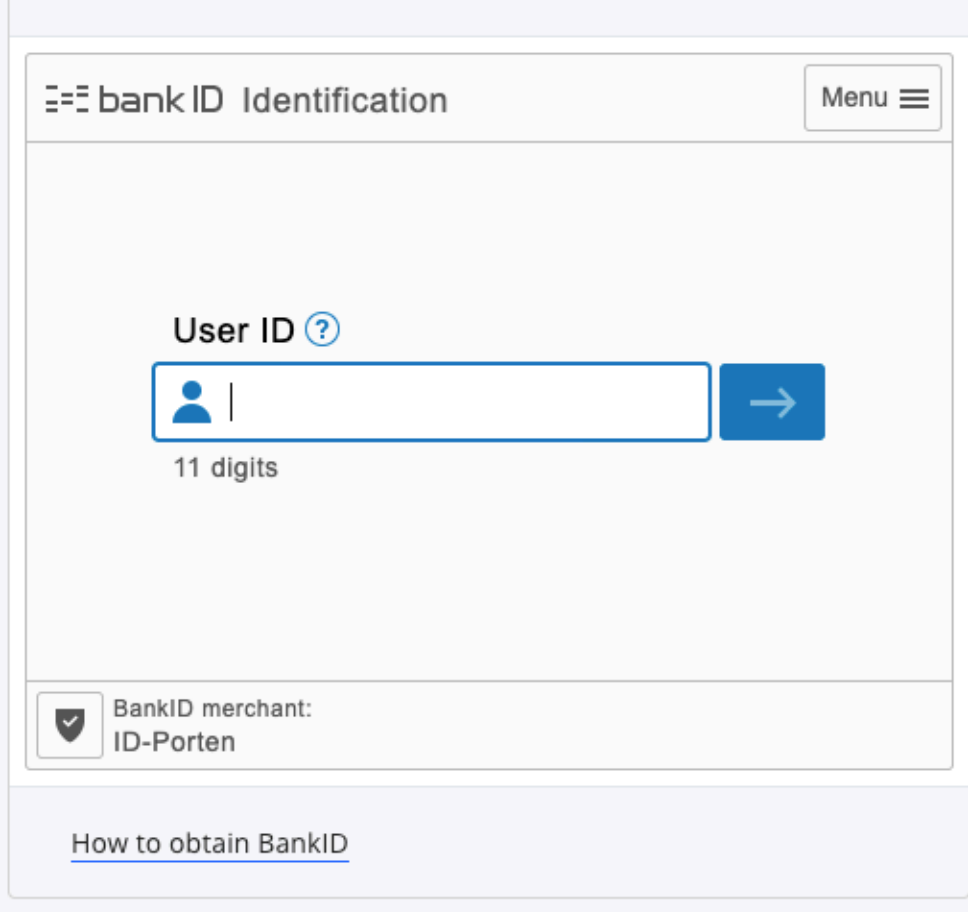
Input Purposes for User Interface Components

Kommentar

BankID login siden har autocomplete skrudd av for personnummer og personlig passord, autocomplete er svært nyttig for mange brukere.

Noen sliter med å huske personnummeret og passordet sitt, andre kan ha problemer med å bruke ett tastatur og derfor ikke ønsker å skrive inn denne dataen.

Det finnes ikke noe HTML autocomplete felt for personnummer, men det finnes ett for passordet som benyttes. Dette kan hjelpe brukere av for eksempel passordbehandlere til å fylle ut passord i feltet.



Ingen feil ble funnet i selve appen

Koblede retningslinjer

WCAG 2.1 - 1.3.5 (AA)

FS80: Utvidede beskrivelser av et spesifikt skjemaobjekt er koblet til dette skjemaobjektet

Ikke aktuelt

Bakgrunn

Det er ikke alltid at en kort beskrivelse er tilstrekkelig for å forklare for brukeren hva, hvordan og hvorfor visse opplysninger i skjemaet må fylles ut. Når det er behov for en utvidet forklaring, for eksempel for å avklare datoformat, eller hvorfor du spør brukeren å fylle ut litt informasjon, skal forklaringen være koblet til skjemaobjektet like tydelig som for vanlig ledetekst.

Den enkleste måten er å la den utvidede beskrivelsen være en del av label-elementet. Det er den mest robuste metoden. Du kan også jobbe med wai-aria attributten aria-labelledby og aria-describedby, men da kommer en del brukere til å miste informasjonen. Hvis det er et problem eller ikke må avklares fra tilfelle til tilfelle, fordi det avhenger av situasjonen og brukeren systemet retter seg mot.

Koblede retningslinjer

WCAG 2.0 - 1.3.1 (A)

WCAG 2.1 - 1.3.1 (A)

FS100: Viktige skjema gir brukeren mulighet til å lese et sammendrag av informasjonen som skal sendes før denne bekrefter og sender inn

Ikke aktuelt

Bakgrunn

Når en nettbasert funksjon brukes for juridisk bindende formål, eller for å overføre penger, er det ekstra viktig at brukeren faktisk forstår hva som kommer til å skje. Det er også lett å velge ”OK” i en nettjeneste uten at være sikker på konsekvensene. For å unngå unødvendige problemer, bør brukeren få mulighet til å lese gjennom et sammendrag før endelig innsending, og deretter bekrefte innsendingen. Dette kan handle om:

- Juridisk bindende handlinger
- Økonomiske transaksjoner
- Endring og sletting av lagret informasjon

Koblede retningslinjer

WCAG 2.0 - 3.3.4 (AA)

WCAG 2.1 - 3.3.4 (AA)

Elmer 3 - 5.2 Gjennomlesing

FS120: Knappers funksjon framgår tydelig

Ingen feil funnet

Bakgrunn

Det er viktig at brukere forstår hva som kommer til å skje når de klikker på en knapp. Knapper bør derfor være selvforklarende slik at brukeren forstår deres funksjon. En knapp som heter ”OK” er for eksempel ikke like tydelig som en knapp som heter ”Kommenter” eller ”Abonner”.

Koblede retningslinjer

WCAG 2.0 - 1.1.1 (A)

WCAG 2.0 - 4.1.2 (A)

WCAG 2.1 - 1.1.1 (A)

WCAG 2.1 - 4.1.2 (A)

Elmer 3 - 5.3 Send-knapp

FS130: Seksjoner i skjema er gruppert

Ikke aktuelt

Bakgrunn

Lengre skjema må deles opp og grupperes. Dette gjør det enklere å se hvilke deler som hører sammen, både visuelt og i et hjelpemiddel. Grupperinger av objekt gjøres med hjelp av fieldset og legend.

Eksempel på et skjema med grupperinger:

```
<form action="http://funka.com/survey" method="post"
onsubmit=”return checkForm(f);”>

 <fieldset>

  <legend>Leveringsadresse</legend>
```

```
  <label for="name1">Navn:</label>
  <input type="text" name="name1" id="name1">

  <label for="adress1">Adresse:</label>
  <input type="text" name="adress1" id="adress1">

  <label for="pnum1">Postnummer:</label>
  <input type="text" name="pnum1" id="pnum1">

  <label for="sted1">Sted:</label>
  <input type="text" name="sted1" id="sted1">

 </fieldset>

 <fieldset>

  <legend>Faktureringsadresse</legend>

  <label for="name2">Navn:</label>
  <input type="text" name="name2" id="name2">

  <label for="adress2">Adresse:</label>
  <input type="text" name="adress2" id="adress2">

  <label for="pnum2">Postnummer:</label>
  <input type="text" name="pnum2" id="pnum2">

  <label for="sted2">Sted:</label>
  <input type="text" name="sted2" id="sted2">

 </fieldset>

 <input type="submit">

</form>
```

Hvis du har en gruppe med radioknapper eller avkrysningsruter, må du alltid bruke fieldset og legend for å gi gruppen en felles beskrivelse. Uten dette blir det, litt forenklet, som å skulle svare på et spørsmål du ikke vet hva er.

Koblede retningslinjer

WCAG 2.0 - 1.3.1 (A)

WCAG 2.1 - 1.3.1 (A)

FS140: Brukeren blir tydelig informert når det oppstår feil

Ingen feil funnet

Bakgrunn

Hvis det finnes en mulig til å gjøre feil, kommer enkelte brukere til å gjøre feil. Da er det viktig at brukeren får vite at noe har blitt feil.

Gi en samlet feilmelding øverst

Det finnes flere ulike måter for å vise at et feil har oppstått, men for at så mange brukere som mulig skal oppfatte det, skal de fleste skjema tilby en samlet feilmelding øverst. Dette skal tydelig angi:

- At det blitt feil. Bruk gjerne både en overskrift og et ikon for å angi at det er en feilmelding.
- Hvilke feil som har skjedd. Gjerne i form av en punktliste.
- Hvor feilen har skjedd. Navngi skjemaobjektene der feilen finnes.
- Om mulig også hva brukeren skal gjøre for å korrigere feilen.
- Beskriv feilet ved respektive felt

På de steder i skjemaet der det har blitt feil, må det finnes en tydelig feilmelding, gjerne med en ikon. Også her skal feilen beskrives for brukeren.

Hva kan jeg gjøre mer?

Enda bedre blir det om du også tilbyr livevalidering slik at feil oppdages etter hvert som brukeren fyller i feltene. Tenk da på at ikke alle brukere kommer til å se dette, det må derfor gå å sende inn skjemaet også uten at brukeren har korrigert feilen. Når brukeren sender inn, skal hun få en samlet feilmelding som beskrevet over.

I den samlede feilmeldingen kan du med fordel også lenke ned til de enkelte feilene i skjemaet, slik at brukeren slipper å lete.

Andre typer feil

Et annet eksempel på feil som kan oppstå, er når brukeren forsøker å laste inn en

side som ikke finnes. Da må nettstedet gi en tydelig feilmelding som forteller at siden ikke finnes, og gi brukeren mulighet å finne informasjonen på annet måte. Dette løses normalt ved å designe sin egen side for 404-feil.

Koblede retningslinjer

WCAG 2.0 - 3.3.1 (A)

WCAG 2.1 - 3.3.1 (A)

Kvalitet på nett - 6.1

Elmer 3 - 4.3 Feil i besvarelse

FS150: Feilmeldinger er koblet til respektive skjemaobjekt

× Underkjent

Bakgrunn

Når feilmeldinger legges inn i skjemaet og er relatert til et spesifikt skjemaobjekt eller en gruppe av objekt (for eksempel en gruppe av radioknapper), må de knyttes til skjemaobjektet eller gruppen. Det gjøres normalt sett ved å legge inn feilmeldingen som en del av label-elementet eller legend-teksten.

Eventuelt kan det innebære at feilmeldingen legges inn to ganger, en gang som vises visuelt og en som ligger skjult i den eksisterende ledeteksten. I så fall er det bra å skjule det visuelle feilmeldingen for hjelpemiddel og se til at det visuelt skjulte feilmeldingen leses.

Dette kan for eksempel gjøres med følgende HTML-kode:

```
<label for=”navn”>Navn: <span class=”skjult”>Feil: Du har glemt å skrive inn navnet ditt</span></label>

<input id=”navn” type=”text”>

<span aria-hidden=”true”>Du har glemt å skrive navnet ditt</span>
```

CSS-kode:

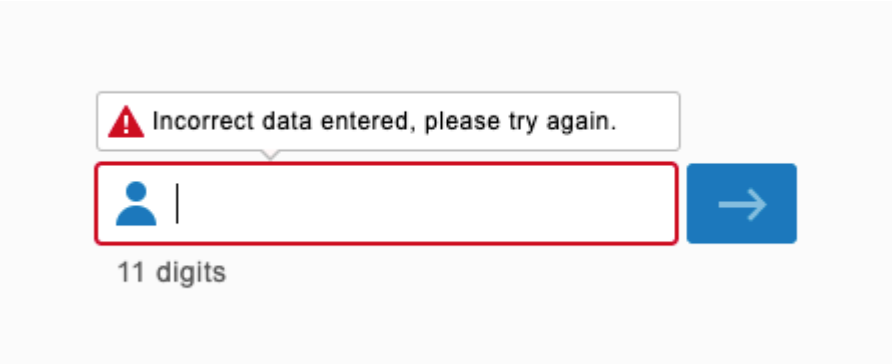
```
.skjult { position: absolute; left: -20000px;}
```

Selvfølgelig er det beste om du bare behøver legge inn feilmeldingen én gang og

gjøre den synlig for alle.

Kommentar

Ingen feil funnet i selve appen, men feilmeldingen for feil inntastning av bankID data er ikke knyttet til skjemafeltet. Vi opplevde at fokus automatisk ble satt på feilmeldingen når den oppstod i VoiceOver for iOS. Men HTML koden indikerer ikke at feilmeldeningen er knyttet til noe skjemafelt.



Anbefalinger

Knytt alle feilmeldinger til respektive skjemaobjekter med bruk av av <label> elementer knyttet til skjemafeltet med en ID eller aria-describedby

Koblede retningslinjer

WCAG 2.0 - 1.3.1 (A)

WCAG 2.1 - 1.3.1 (A)

FS170: Alle feil beskrives med tekst

Ingen feil funnet

Bakgrunn

Feilmeldingen kan inneholde bilder eller til og med filmer, men det er viktig at det alltid finnes en feilmelding i form av tekst. Det holder altså ikke med et ikon for å indikere at noe er feil, det skal alltid uttrykkelig stå i tekst hva som er feil.

Koblede retningslinjer

WCAG 2.0 - 3.3.1 (A)

WCAG 2.1 - 3.3.1 (A)

Elmer 3 - 4.3 Feil i besvarelse

FS180: Når det er mulig, bør det være klart hvordan feilene kan rettes opp

Ingen feil funnet

Bakgrunn

Beskriv så detaljert som mulig hva som er feil og hvordan brukeren kan ordne det.

For eksempel kan du gi forslag på hva som kan være feil, for eksempel:

Du må angi en epostadresse som inneholder et @-tegn

I stedet for

Du må angi en gyldig e-postadresse

Koblede retningslinjer

WCAG 2.0 - 3.3.3 (AA)

WCAG 2.1 - 3.3.3 (AA)

Elmer 3 - 4.3 Feil i besvarelse

Tabeller

TB40: Tabelldata er kodet med korrekt tabellkode

Ikke aktuelt

Bakgrunn

Kod alltid tabeller med korrekt tabellkode. Det er mulig å skape innhold som ser ut som tabeller, men som ikke er kodet som faktiske tabeller. Det finnes også HTML5-roller og WAI-ARIA attributt som kan brukes for å skape tabeller i stedet for tradisjonell HTML-kode, men det er løsninger som i dag ikke fungerer bra for brukere med hjelpemidler. Tabeller skal derfor alltid bygges opp med elementene table, tr, td og th.

Koblede retningslinjer

WCAG 2.0 - 4.1.2 (A)

WCAG 2.1 - 4.1.2 (A)

TB50: Overskrifter til tabeller er kodet med elementet caption

× Underkjent

Bakgrunn

Tabeller er en av de mest komplekse teknikkene som finnes for å presentere data innen HTML. For at en bruker skal ha en mulighet til å forstå tabellen må det finnes tydelige tabelloverskrifter som er korrekt lagt inn.

Hvis det ikke finnes spesielle grunner skal elementet caption brukes for å legge inn en overskrift til tabellen. Denne overskriften skal kort forklare hva slags informasjon tabellen inneholder. Caption-elementet legges øverst i tabellen på samme nivå som tabellrader (tr-elementer):

```
<table>
   <caption>Flypriser Stockholm-Luleå</caption>
   <tr>
 …
```

Koblede retningslinjer

WCAG 2.0 - 1.3.1 (A)

WCAG 2.1 - 1.3.1 (A)

TB60: Rad- og kolonneoverskrifter er kodet med elementet th

× Underkjent

Bakgrunn

For at brukerne skal kunne tolke innholdet i tabellene korrekt er det viktig at tabellene har tydelige kolonne- og radoverskrifter. Disse overskriftene skal ikke bare være visuelt tydelige, de må også være strukturelt riktig kodet med elementet th. Th forteller at cellen er en ”Table Header cell”, altså en celle for tabelloverskrift, i motsetning til elementet td som angir at cellen er en ”Table Data cell”, altså en datacelle. Alle tabelloverskrifter skal legges inn med elementet th.

Koblede retningslinjer

WCAG 2.0 - 1.3.1 (A)

WCAG 2.1 - 1.3.1 (A)

TB70: Attributet scope blir brukt for å indikere retningen til hver rad- og kolonneoverskrift

Ikke aktuelt

Bakgrunn

Tabelloverskriftene må også knyttes korrekt med de dataceller (og eventuelle under-/mellomoverskrifter) som er underordnet overskriften. For enkle tabeller holder det om tabelloverskriften i attributtet scope angir at den relaterer seg til kolonnen (scope=”col”) eller raden (scope=”row”). For tabeller i mer enn to logiske nivåer kreves en mer avansert oppmerking (se TB100).

Eksempel på tabell med rad og kolonneoverskrifter:

```
<table>
  <caption>Flypriser Oslo-Stockholm</caption>
  <tr>
    <th scope="col">Tidspunkt</th>
    <th scope="col">30.12.2011</th>
    <th scope="col">31.12.2011</th>
    <th scope="col">01.01.2012</th>
  </tr>
  <tr>
    <th scope="row">07:40</th>
    <td>529,-</td>
    <td>Utsolgt</td>
    <td>449,-</td>
  </tr>
  <tr>
    <th scope="row">14:00</th>
    <td>Ingen avgang</td>
    <td>379,-</td>
    <td>579,-</td>
  </tr>
  <tr>
    <th scope="row">17:00</th>
    <td>379,-</td>
    <td>699,-</td>
    <td>Ingen avgang</td>
```

```
  </tr>
</table>
```

Med litt stiling gir dette følgende tabell:

Flypriser Oslo-Stockholm

Tid	30.12.2011	31.12.2011	01.01.2012
07:40	529,-	Utsolgt	449,-
14:00	Ingen avgang	379,-	579,-
17:00	379,-	699,-	Ingen avgang

Koblede retningslinjer

WCAG 2.0 - 1.3.1 (A)

WCAG 2.1 - 1.3.1 (A)

TB100: Komplekse tabeller kompletteres med relevant kode

Ikke aktuelt

Bakgrunn

En tabell regnes som kompleks dersom ett eller flere av følgende er tilfelle:

- En overskriftcelle dekker flere kolonner
- En overskriftcelle dekker flere rader
- Radoverskriftene ikke står i første kolonne fra venstre
- Det er flere nivå av overskrifter på samme kolonne og/eller rad
- Det er overskriftceller som forholder seg til dataceller som verken står i samme rad eller i samme kolonne som overskriftcellen.

Komplekse tabeller må ha mer spesifikke koblinger mellom overskrifter og dataceller enn enkle tabeller. Det er to metoder som er mulige å benytte for dette:

- Lage kolonnegrupper og benytte scope="colgroup" på de aktuelle overskriftcellene. Denne er bare relevant dersom den eneste betingelsen som gjelder fra listen over er at en overskriftcelle dekker flere kolonner.
- Kombinere id-verdier på overskriftcellene og header-attributtet på datacellene.

Hvis du har en kompleks tabell, bør du i første omgang vurdere å dele den opp i

flere, enklere tabeller eller på annen måte forenkle oppstillingen slik at du klarer deg med bare ordinære rad- og koloneoverskrifter for å assosiere overskrifter og dataceller.

Kolonnegrupper defineres med <colgroup>-elementet, som skal være barn av <table>-elementet, men etter <caption>. Deretter setter du scope=”colgroup” på overskriftceller som matcher kolonnegruppene. Dette forutsetter at alle datacellene kan hente sine tilhørende overskrifter enten rett opp eller rett til venstre.

```
<table>
 <caption>Flypriser Oslo-Stockholm</caption>
 <colgroup>
  <col span=”1”>
  <col span=”3”>
  <col span=”3”>
 </colgroup>
 <tr>
  <td></td>
  <th colspan="3" scope=”colgroup”>Ryk og Reis Airlines</th>
  <th colspan="3" scope=”colgroup”>Klar Ferdig Fly</th>
 </tr>
 <tr>
  <td></td>
  <th scope=”col”>30.12.2011</th>
  <th scope=”col”>31.12.2011</th>
  <th scope=”col”>01.01.2012</th>
  <th scope=”col”>30.12.2011</th>
  <th scope=”col”>31.12.2011</th>
  <th scope=”col”>01.01.2012</th>
 </tr>
 <tr>
  <th scope=”row”>Morgenfly</th>
  <td>829,-</td>
  <td>Utsolgt</td>
  <td>449,-</td>
  <td>466,-</td>
```

```
  <td>563,-</td>
  <td>463,-</td>
 </tr>
 ...
</table>
```

Den samme tabellen løst med attributtene id og headers:

```
<table>
 <caption>Flypriser Oslo-Stockholm</caption>
 <tr>
  <td></td>
  <th colspan="3" id="rra">Ryk og Reis Airlines</th>
  <th colspan="3" id="kff">Klar Ferdig Fly</th>
 </tr>
 <tr>
  <td></td>
  <th id="d30122011a" headers="rra">30.12.2011</th>
  <th id="d31122011a" headers="rra">31.12.2011</th>
  <th id="d01012012a" headers="rra">01.01.2012</th>
  <th id="d30122011b" headers="kff">30.12.2011</th>
  <th id="d31122011b" headers="kff">31.12.2011</th>
  <th id="d01012012b" headers="kff">01.01.2012</th>
 </tr>
 <tr>
  <th id="morgen">Morgenfly</th>
  <td headers="rra d30122011a morgen">829,-</td>
  <td headers="rra d31122011a morgen">Utsolgt</td>
  <td headers="rra d01012012a morgen">449,-</td>
  <td headers="kff d30122011b morgen">466,-</td>
  <td headers="kff d31122011b morgen">563,-</td>
  <td headers="kff d01012012b morgen">463,-</td>
 </tr>
 <tr>
  <th id="middag">Middagsfly</th>
  <td headers="rra d30122011a middag">Ingen avgang</td>
```

```
  <td headers="rra d31122011a middag">379,-</td>
  <td headers="rra d071225a middag">579,-</td>
  <td headers="kff d071223b middag">773,-</td>
  <td headers="kff d071224b middag">Utsolgt</td>
  <td headers="kff d071225b middag">1685,-</td>
 </tr>
 <tr>
  <th id="kveld">Kveldsfly</th>
  <td headers="rra d071223a kveld">379,-</td>
  <td headers="rra d071224a kveld">699,-</td>
  <td headers="rra d071225a kveld">Ingen avgang</td>
  <td headers="kff d071223b kveld">563,-</td>
  <td headers="kff d071224b kveld">Ingen avgang</td>
  <td headers="kff d071225b kveld">563,-</td>
 </tr>
</table>
```

Flypriser Oslo-Stockholm

	Ryk og Reis Airlines			Klar Ferdig Fly		
	30.12.2011	**31.12.2011**	**01.01.2012**	**30.12.2011**	**31.12.2011**	**01.01.2012**
Morgenfly	829,-	Utsolgt	449,-	466,-	563,-	463,-
Middagsfly	Ingen avgang	379,-	579,-	773,-	Utsolgt	1685,-
Kveldsfly	379,-	699,-	Ingen avgang	563,-	Ingen avgang	563,-

Koblede retningslinjer

WCAG 2.0 - 1.3.1 (A)

WCAG 2.1 - 1.3.1 (A)

Script og WAI-ARIA

SW20: Bruk WAI-ARIA for å formidle informasjon som ikke kan formidles med HTML, og som øker brukeropplevelsen

Ingen feil funnet

Bakgrunn

Standarden WAI-ARIA inneholder et sett med attributter som kan brukes for å tydeliggjøre ulike html-objekters funksjon, verdi og rolle. En del av det som inngår i WAI-ARIA har allerede bra løsninger i HTML5, og da skal HTML-løsningen brukes. Men mye i wai-aria tydeliggjør og øker tilgjengeligheten ytterligere utover det som er innebygd i HTML5.

Et eksempel der HTML5 og WAI-ARIA overlapper gjelder navigering. Du skal helst benytte HTML5 attributtet <nav> for å vise at noe er en meny, men det finnes en mulighet i WAI-ARIA å også angi det gjennom attributtet role=”navigation”. Vår anbefaling er altså at du helst bruker løsningen som finnes i HTML5.

Ikke overbruk WAI-ARIA

Selv om WAI-ARIA inneholder mye som er bra, så det er viktig at du ikke overbruker standarden. Du skal bare legge til objekter om du vet at det øker brukernytten. Unngå å legge på WAI-ARIA om HTML-koden allerede tydeliggjør funksjonen, for eksempel er det unødvendig å skrive <button role=”button”>, det risikerer bare å skape dobbel opplesing for brukerne. Bruk ikke heller role=menu og role=menuitem for vanlige nettstedsmenyer, dette er tenkt for applikasjoner, for eksempel teksteditorer, og brukerne blir bare forstyrret over å høre hjelpemidlene lese ”menyobjekt” for hver lenke i menyen.

Kommentar

I bankID logindelen har ikke feilmeldingen en aria-live som annonser at feilmeldingen er kommet opp.

Ingen feil funnet i selve appen.

Koblede retningslinjer

WCAG 2.0 - 4.1.2 (A)

WCAG 2.1 - 4.1.2 (A)

SW50: Når nye elementer legges inn i siden, skal de plasseres på rett plass både visuelt og strukturelt

Ingen feil funnet

Bakgrunn

Iblant legges nytt innhold eller nye funksjoner inn på siden uten at den lastes om. Når det skjer er det viktig at den nye funksjonen eller informasjonen kommer opp på riktig sted både visuelt og strukturelt. Om søkeforslag dukker opp i et nytt område på skjermen når brukeren begynner å skrive i søkefeltet er det viktig at dette området er plassert visuelt tydelig der det hører hjemme, men også på rett plass i strukturen. Da vil også hjelpemidler strukturelt kunne presentere informasjonen på rett sted. Den beste løsningen på dette er om det gjøres på en slik måte at den nye koden settes inn på rett plass, som følgende eksempel:

```
<label for="sok">Søk</label>
<input type="search" id="sok" />
<!- Søkeforslag lastes inn her -->
<input type="submit" value="Søk" />
```

Det finnes andre måter å løse dette på også, som for eksempel gjennom ved å benytte script gjenkjenne hva brukeren gjør og flytte fokus slik at effekten for brukeren blir den samme. Hvordan dette fungerer burde alltid testes i praksis med de vanligste hjelpemidlene.

Et vanlig problem er at et nytt område plasseres visuelt korrekt men havner sist på siden i strukturen. Om et hjelpemiddel ikke finner noe før sidefoten er det en stor risiko for at brukeren aldri kommer til å finne det.

Koblede retningslinjer

WCAG 2.0 - 1.3.2 (A)

WCAG 2.0 - 2.4.3 (A)

WCAG 2.1 - 1.3.2 (A)

WCAG 2.1 - 2.4.3 (A)

SW60: Når nytt innhold eller nye områder åpnes, får brukeren direkte fokus på dette

! Ingen feil funnet men kan forbedres

Bakgrunn

Om et dekkende lag, eller et nytt område åpnes dynamisk på siden, er det viktig at brukere med en gang får fokus på dette (om det ikke forstyrrer brukeren). Eksempel på unntak er søkeforslag. Brukerens fokus skal ikke flyttes automatisk til søkeforslagene så fort de dukker opp, ettersom det vil være forstyrrende.

For en bruker som navigerer med tastaturet er det viktig å komme direkte inn i det nye laget eller innholdet uten å behøve tabbe til slutten av siden. Om det er et skjema kan fokus settes i det første feltet, ellers skal det første tabsteget etter lenken som åpnet laget eller resulterte i at det nye området lastes inn, lede inn til første lenken i området.

Tenk også på at leserekkefølgen må bli korrekt for brukere med opplesende hjelpemiddel. Når de fortsetter lesingen nedover på siden fra den lenken som åpnet laget, skal lesingen fortsette inne i laget det gjelder. Dette kan kreve at du tester det med hjelpemiddel.

Om det gjelder meldinger og dialogvinduer bør WAI-ARIAs roller for meldinger og meldingsdialoger benyttes. Om det er et skjema bør det være rollen alertdialog:

```
role=”alertdialog”
```

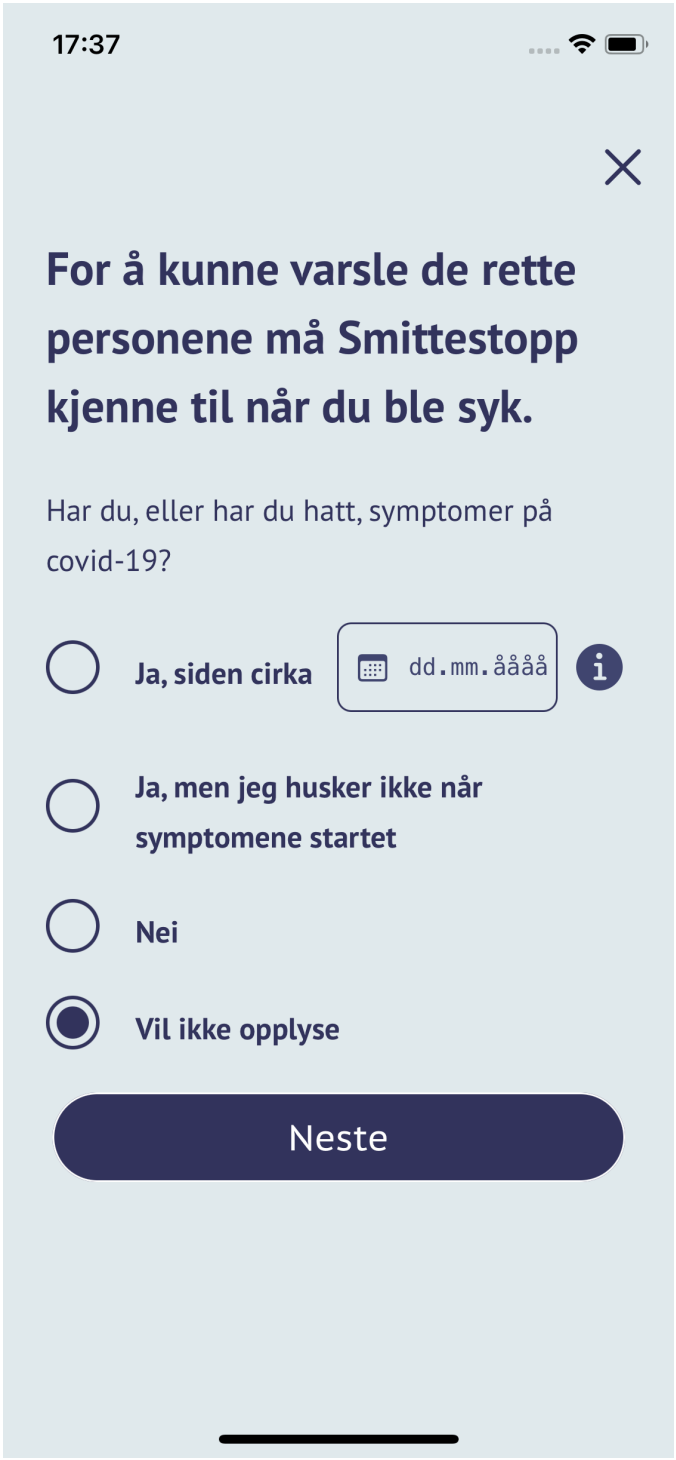
Ellers rollen alert:

```
role=”alert”
```

Information om WAI-ARIA (W3C’s nettsted)

Kommentar

På alle sider med en "X" for å lukke siden settes fokus automatisk på X-elementet som lukker siden fremfor overskriften. Dette kan føre til at brukere ved uhell lukker en side med svært viktig informasjon når de først kommer til den.



Anbefalinger

Sett fokus på overskriften på alle sider som åpnes, ikke lukk-knappen.

Koblede retningslinjer

WCAG 2.0 - 2.4.3 (A)

WCAG 2.1 - 2.4.3 (A)

SW70: Når status endres, informeres tilleggsbrukere uten å flytte fokusmerket

Ingen feil funnet

Bakgrunn

Det er svært vanskelig for brukere med sterkt nedsatt syn å oppfatte oppdateringer i grensesnittet som skjer uten at siden lastes om. Det kan være at det skjer en forandring i en pågående prosess, for eksempel en oppdatert køplass på et nettsted for salg av konsertbilletter. Det kan også handle om å informere brukeren om hvor mange søketreff et søk har resultert i, eller validering av feil i et skjema. Det er problematisk for brukeren å raskt flytte seg til en annen del av grensesnittet for å rekke å lese en statusmelding innen det er utdatert eller har blitt oppdatert.

Eksempler på statusendringer er:

- Informasjon om antall søketreff i et søk
- En forandring skjer i en ventende applikasjon
- En forandring skjer i en pågående prosess
- En feil oppstår i et skjema

Når en statusmelding vises på skjermen skal brukeren få informasjon om det uten å trenge å flytte seg til et annet område på siden. Attributtet role=”alert”, aria-live og aria-busy kan brukes for å informere skjermleserbrukeren om statusmeldingen som vises på andre deler av siden enn der brukeren nå befinner seg. Dette innebærer at brukeren får statusmeldingen, uten å selv trenge og lete etter forandringer eller å oppdatere siden. Hjelpemiddel for brukere med kognitive funksjonsnedsettelser kan tydeliggjøre, forsinke eller dempe statusmeldingen, avhengig av hva brukeren foretrekker.

```
<div class="medium-12 columns">
<h2 aria-live="assertive" aria-atomic="true">
Ditt søk på <strong><mark>universell utforming</mark>
</strong>resulterte i 538 treff
</h2>
</div>
```

WAI-ARIA-attributt kan brukes i kombinasjon for å gjøre det tydeligere for brukeren at en statusoppdatering er spesielt viktig. Et eksempel på det er å bruke

attributtet role=”alert” og aria-live=”assertive” i meldingen som informerer brukeren om at denne holder på å logge ut.

Attributtet aria-live har tre nivåer som bestemmer hvor høy prioritet en oppdatering i et område skal ha:

- Assertive = En viktig melding som skal avbryte hjelpemiddelet så snart som mulig.
- Polite = En mindre viktig melding som skal presenteres om brukeren pauser.
- Off = En melding som ikke trenger å presenteres for brukeren.

Denne retningslinjen gjelder ikke feilmeldinger i modale vinduer, innhold som vises eller skjules med utfoldbare objekter, eller innhold som krever inntasting fra brukeren. Informasjonen om cookies regnes altså ikke som en statusmelding ettersom det ofte krever at brukeren bekrefter sitt samtykke.

Koblede retningslinjer

WCAG 2.1 - 4.1.3 (AA)

SW80: Innhold som vises når et objekt er fokusert, er fullt nyttig og oppnåelig til brukeren beveger fokuset

Ingen feil funnet

Bakgrunn

Brukere med forstørrende hjelpemidler ser bare en liten del av skjermen om gangen, som gjør det vanskelig å få overblikk over store områder. Det er også problematisk for brukeren å finne innhold som ligger utenfor det området som blir forstørret. Brukere som navigerer med mus eller tastatur og forstørrende hjelpemidler, som følger musepekeren og tastaturets bevegelser, må derfor kunne flytte fokus uten at et område forsvinner. Dette retningslinjen gjelder både ved hover med musepekeren, og fokus med tastatur.

Et eksempel på dette er megamenyer, altså menyer som består av flere nivåer og presenteres i store områder som vises når brukeren kretser over menyen. De dekker ofte annet innhold, og forsvinner når brukerens musepeker går utenfor området. For at brukere med forstørrende hjelpemidler skal kunne komme til alt innhold, skal brukeren kunne flytte fokus fritt innen området uten at det stenges. Brukeren skal også ha mulighet til å stenge området på en enkel måte, for eksempel med escape-tasten.

Området som ikke skjuler annet innhold trenger ikke ha en stengingsmekanisme.

Et eksempel er menyer som i stedet for og åpnes i et dekkende lag, åpnes ovenfor sidens innhold, som flyttes ned og plasseres nedenfor menyen. Heller ikke feilmeldinger som beskriver inntastingsfeil trenger å ha en stengingsmekanisme.

Denne retningslinjen gjelder ikke modale vinduer eller tootips som legges inn med attributtet title, ettersom disse implementeres av selve nettleseren.

Koblede retningslinjer

WCAG 2.1 - 1.4.13 (AA)

SW90: Når et modalt vindu vises, skal fokuset stå i modalen til den blir lukket

Ingen feil funnet

Bakgrunn

Hvis du bruker modale vinduer i grensesnittet, er det viktig at alle brukerne kan komme tilbake til hovedinnholdet på en enkel måte, uten å bli sittende fast i modalt vindu.

En vanlig feil når du bruker modalt vindu er at brukeren ved hjelp av tastaturet eller hjelpemidler kan ende opp bak vinduet, selv om det ikke er lukket. Med andre ord er det et krav at brukere som navigerer med tastatur eller hjelpemidler bare kan orientere seg inne i vinduet til brukeren har lukket den.

Når det gjelder å blokkere hjelpemidler fra bakgrunnen, anbefaler vi å bruke attributtet aria-hidden=”true”, som du kan sette så høyt i strukturen som mulig for å dekke alt innholdet som ikke er i modalvinduet.

Det er derimot mer krevende å blokkere bakgrunnen for navigasjon med tastatur, fordi du må endre standardoppførselen for hver enkelt objekt ved å sette tabindex = "- 1" på alle koblinger, knapper og skjemaobjekter, og eventuelt også til andre typer elementer som allerede har tabindex-attributtet.

Koblede retningslinjer

WCAG 2.0 - 2.4.3 (A)

WCAG 2.1 - 2.4.3 (A)

SW100: Bruk aria-expanded for områder som kan foldes ut og sammen

− Ikke aktuelt

Bakgrunn

Attributtet aria-expanded skal benyttes på områdes som kan foldes ut og foldes sammen. Attributtet forteller om området er utfoldet eller ikke. Dette kan gjelde en FAQ-liste der svarene foldes ut om brukeren klikker på spørsmålet, eller på en meny som dynamisk folder seg ut til neste lag. Aria-expanded skal bare benyttes om dette skjer dynamisk, altså ikke når det skjer ved at en side lastes inn på nytt.

```
<h3>
 <a aria-controls="svar1" aria-expanded="true" href="#svar1"
onclick="javascript:expand-svar1();" role="button">
 Når skal jeg benytte aria-expanded?
 </a></h3>

<p id="svar1">Du skal benytte det når et område kan foldes ut
og sammen dynamisk uten at siden lastes om.</p>

<h3>
<a aria-controls="svar2" aria-expanded="false" href="#svar2"
onclick="javascript:expand-svar2();" role="button">
Et annet spørsmål?
</a></h3>

<p id="svar2">Et annet svar.</p>
```

Koblede retningslinjer

WCAG 2.0 - 4.1.2 (A)

WCAG 2.1 - 4.1.2 (A)

Automatiske hendelser

AE20: Nye vindu og dekkende lag åpnes bare når brukeren har valgt å gjøre det

Ingen feil funnet

Bakgrunn

Nye vinduer og lag som delvis dekker den egentlige nettsiden skaper problem for flere grupper av brukere. De skaper forvirring og er mange ganger vanskelige å stenge. Derfor skal brukeren alltid få informasjon om at ett nytt vindu eller dekkende lag kommer til å åpnes. Brukeren bør også få mulighet til å velge om et nytt vindu eller lag skal åpnes.

Dette punktet er rettet mot lag og vinduer som åpnes automatisk, som for eksempel nettstedsundersøkelser som legger seg over det vanlige vinduet når besøkeren viser en side, eller reklamevinduer som åpner når brukeren kommer inn på nettstedet. Vanlige lenker som åpner nye vinduer skal indikere dette til brukeren, men bedømmelsen av den informasjonen gjøres under punkt P11 i den pedagogiske revisjonen.

Normalt sett skal ikke nye vinduer og dekkende lag dukke opp uten at brukeren aktivt har klikket på en lenke som tydelig angir hva som kommer til å skje, men et alternativ kan være å la brukeren sette en innstilling der denne aksepterer at denne typen hendelser inntreffer.

Koblede retningslinjer

WCAG 2.0 - 3.2.1 (A)
WCAG 2.0 - 3.2.2 (A)
WCAG 2.0 - 3.2.5 (AAA)
WCAG 2.1 - 3.2.1 (A)
WCAG 2.1 - 3.2.2 (A)
WCAG 2.1 - 3.2.5 (AAA)

AE60: Hvis automatiske hendelser eller tidsgrenser brukes finnes det en mulighet for å pause eller forlenge tidsintervallet

Ingen feil funnet

Bakgrunn

For hver tidsgrense må det være mulig å enten:

- Deaktivere
- Endre. Brukeren kan selv styre hvor lang tidsgrensen skal være og maksverdien er minst ti ganger lenger enn grunnverdien
- Forlenge. For eksempel gjennom at en advarsel kommer opp noen minutter innen tidsgrensen løper ut og brukeren gis mulighet til å forlenge

Brukeren kan for eksempel få mulighet til å deaktivere, endre eller forlenge tidsgrensen gjennom en knapp som vises i sammenheng til tidsgrensen. Et annet alternativ er å vise en melding noen minutter innen utlogging der brukeren kan velge å forlenge innloggingstiden.

Tidsgrensen skal alltid forholde seg til seneste brukeraktivitet som systemet kan kjenne til, ikke til innloggingstidspunktet. Om brukeren for eksempel går fra steg 1 til steg 2 i en tjeneste så skal det nullstille nedtellingen til utloggingen.

Om tidsgrensen er slik at meningen med funksjonen forsvinner helt om denne forlenges eller endres kan det være et akseptabelt unntak. Likeså om tidsgrensen er på minst 20 timer.

Koblede retningslinjer

WCAG 2.0 - 2.2.1 (A)
WCAG 2.0 - 2.2.2 (A)
WCAG 2.0 - 2.2.4 (AAA)
WCAG 2.0 - 3.2.5 (AAA)
WCAG 2.1 - 2.2.1 (A)
WCAG 2.1 - 2.2.2 (A)
WCAG 2.1 - 2.2.4 (AAA)
WCAG 2.1 - 3.2.5 (AAA)

AE80: Hvis et område oppdateres uten at siden lastes om, skal dette være markert med WAI-ARIA-attributtet aria-live

Ikke aktuelt

Bakgrunn

Aria-live angir hvor viktige forandringer i grensesnittet er. Det avgjør for eksempel om et hjelpemiddel skal avbryte brukeren eller ikke.

Benytt følgende verdier:

- Off: Benytt for noe som oppdateres ofte, for eksempel aksjeinformasjon.
- Polite: Benytt om en endring bare skal gi beskjed om brukeren ikke har noe viktig for seg.
- Assertive: Benytt når det skjer noe viktig som brukeren burde gjøres oppmerksom på så snart som mulig, men som ikke umiddelbart må avbryte brukeren uansett omstendigheter.

Koblede retningslinjer

WCAG 2.0 - 4.1.2 (A)

WCAG 2.1 - 4.1.2 (A)

Rammer

FR20: Formålet til hver ramme er beskrevet i attributtet title

Ingen feil funnet

Bakgrunn

Formålet til hver enkelt ramme må beskrives i attributtet title. Denne informasjonen presenteres av hjelpemidler for svaksynte og hjelper dem med å forstå om de skal inn i rammen og lese den nettsiden, eller fortsette på den eksisterende nettsiden utenfor rammen.

Koblede retningslinjer

WCAG 2.0 - 4.1.2 (A)

WCAG 2.1 - 4.1.2 (A)

Søkefunksjoner

SF130: Det er mer enn en måte å finne informasjon på

Ikke aktuelt

Bakgrunn

For å oppfylle kravene i WCAG 2.1 på nivå AA, må nettstedet ditt tilby flere måter å finne en bestemt side på (gjelder ikke for trinn i e-tjenester eller prosesser). Om du ikke har en søkefunksjon må du tilby et alternativ, for eksempel et nettstedskart eller en A-Å-funksjon.

Koblede retningslinjer

WCAG 2.0 - 2.4.5 (AA)

WCAG 2.1 - 2.4.5 (AA)

Dokumenter og andre medieformater

DM10: Bakgrunnslyd kan enkelt stenges av manuelt, eller avslutter automatisk innen 3 sekunder

Ikke aktuelt

Bakgrunn

På samme måte som bevegelser i grensesnittet påvirker besøkerens evne å ta til seg informasjonen, påvirker også bakgrunnslyd konsentrasjonsevnen. Derfor skal bakgrunnslyd som starter automatisk, stoppe etter maksimalt 3 sekunder, alternativt må det finnes en tydelig og enkel måte å stoppe lyden.

Koblede retningslinjer

WCAG 2.0 - 1.4.2 (A)

WCAG 2.1 - 1.4.2 (A)

DM20: Det finnes en passende tekstbeskrivelse av materiale i komplekse format i tilknytning til dette materialet

Ikke aktuelt

Bakgrunn

Komplekse format er ofte bra for brukeren. Det gir mulighet til å forklare kompliserte sammenhenger på en mer forståelig måte. Det er derimot ikke alltid brukeren kan eller vil benytte seg av komplekse format. Det kan bero på at brukerens utrustning ikke støtter teknikken eller at oppkoblingen er dårlig. Det må derfor alltid finnes en tydelig beskrivelse av innholdet i disse komplekse formatene. Når du lenker til for eksempel en film må du derfor beskrive filmens innhold i tekst. Hvor omfattende beskrivelsen skal være avhenger helt på filmens innhold og relevans. En beskrivelse kan også inneholde lenker til andre deler på nettstedet der besøkeren kan lese tilsvarende informasjon som filmen gir.

Lyd

Når informasjon presenteres med hjelp av et lydklipp, må tilsvarende informasjon også finnes i tekstform på nettstedet. Det må også finnes en tydelig kobling mellom lydklippet og den alternative teksten.

Ved livesendinger skal det som minimum finnes en tydelig beskrivelse av hva som sendes, og om mulig bør det også være en tekstgjengivelse. Det kan for eksempel skje ved at notatene til et møtes protokollfører samsendes live med lyden.

Film

For innspilt film skal det finnes en tekstversjon av innholdet tydelig lenket i tilknytning til filmen. En tekstversjon behøver ikke være en ordrett gjengivelse av innholdet, men skal gi likeverdig informasjon. Alternativt må filmen både utstyres med en teksting (caption) som gjengir den informasjon som gis med lyd og en speakerstemme (audio description) som gjengir visuell informasjon som ikke er åpenbar ut fra lyden i filmen.

Ved livesendinger skal det som minimum finnes en tydelig beskrivelse av hva som sendes og om mulig kan du også gi en tekstgjengivelse. Det kan for eksempel skje ved at notatene til et møtes protokollfører samsendes live med filmen.

PDF

Også PDF regnes som et komplekst medieformat, og skaper iblant problemer for

brukerne. Du bør derfor alltid vurdere om du kan bruke HTML i stedet for PDF, både for informasjon og for skjema. I det minste bør du vurdere om det bør være et sammendrag av PDF-ens innhold i HTML, og det må alltid være tydelige lenketekster til PDF-filer. Tenk også på at selve PDF-filen må være universelt utformet.

Koblede retningslinjer

WCAG 2.0 - 1.1.1 (A)
WCAG 2.0 - 1.2.1 (A)
WCAG 2.1 - 1.1.1 (A)
WCAG 2.1 - 1.2.1 (A)

DM30: Lydinformasjon i forhåndsinnspilt video- og lydklipp gjengis som tekst

− Ikke aktuelt

Bakgrunn

Hvis du har lyd som en del av en video eller som en lydfil, så skal innholdet finnes tekstet. For video holder det med en tekststreng som er synkronisert med lyden. Har du et lydklipp kan det være vanskelig å legge inn en tekststreng, og i stedet kan du tilby brukeren å lese en tekstutskrift av lydinformasjonen.

Tenk på at tekstingen skal gjengi lydinformasjon, som betyr tale, men kan også være annen informasjon. Hvis det for eksempel ringer på døren i filmen og skuespilleren derfor går til døren og åpner, så skal tekstingen også formidle at det ringer på døren.

Koblede retningslinjer

WCAG 2.0 - 1.2.2 (A)
WCAG 2.0 - 1.2.9 (AAA)
WCAG 2.1 - 1.2.2 (A)
WCAG 2.1 - 1.2.9 (AAA)

DM40: Lydinformasjon i direktesendinger gjengis som tekst

Ikke aktuelt

Bakgrunn

Også når du har lydinformasjon i direktesendinger, så bør informasjonen tekstes. Vi anbefaler at du gjør en vurdering om det er fornuftig å tilby liveteksting i hvert aktuelle tilfelle, men for å møte kravet i WCAG 2.0 nivå AA kreves det at direktesendinger tekstes.

Dette punktet omfattes ikke av Likestillings- og Diskrimineringsloven

Koblede retningslinjer

WCAG 2.0 - 1.2.4 (A)

WCAG 2.1 - 1.2.4 (A)

DM100: Visuell informasjon i video forklares med lyd

Ikke aktuelt

Bakgrunn

Video bør synstolkes, slik at en bruker som ikke ser videoen har mulighet til å forstå hva som skjer. Dette kan i mange tilfeller gjøres gjennom at en vanlig fortellerstemme forklarer hva som skjer, men i mer komplekse filmer kan det være nødvendig med et separat lydspor for bare synstolkingen.

Også i det tilfelle bør du gjøre en vurdering om det er fornuftig å tilby dette, men også her er det slik at det skal finnes en synstolking hvis dere skal møte WCAG 2.0 nivå AA.

Dette punktet omfattes ikke av Likestillings- og Diskrimineringsloven.

Koblede retningslinjer

WCAG 2.0 - 1.2.3 (A)

WCAG 2.0 - 1.2.5 (AA)

WCAG 2.1 - 1.2.3 (A)

WCAG 2.1 - 1.2.5 (AA)

DM200: Det er tydelig for brukerne hvor de kan finne alternativ til informasjon som presenteres på annen måte enn tekst

Ikke aktuelt

Bakgrunn

En film, bilde eller lydfil kan ofte gjengi kompleks informasjon på en enklere måte enn hva en tekst kan. Du bør derfor strebe etter å benytte grafikk og lyd for å gi informasjon når det er gunstig.

Så fort du benytter grafikk, film eller lyd for å gi informasjon, må du også gi et alternativ til de brukerne som ikke kan se eller høre informasjonen. For eksempel kan du tilby tekstversjon av en film, mens informasjon i et diagram kan du gjengi som tabell med samme data. For vanlige bilder er det nok med en alt-tekst (attributtet alt). Da ligger beskrivelsen direkte på bildet, men så fort du jobber med mer komplekse alternativ enn vanlige alt-tekster må du tydelig koble sammen de ulike alternativene.

Det er et vanlig problem at alternative versjoner er vanskelig for brukeren å finne. Grunnregelen er at alternativ skal lenkes eller presenteres i direkte tilknytning til det de er alternativ til. For eksempel kan fliker ved siden av hverandre la brukeren veksle mellom et diagram og en tabell, som vist i eksempelet under. Et tekst- eller tegnspråksalternativ til en film bør ligge direkte under filmen. Det er derimot ikke tilstrekkelig bare å legge inn en lenke i høyrespalten eller forvente at brukerne finner siden ”Tilgjengelighet” i menyen.

Koblede retningslinjer

WCAG 2.0 - 1.1.1 (A)

WCAG 2.1 - 1.1.1 (A)

Hjelp og overgang til manuell service

MS40: Hvis det finnes en forklaring for hvordan en bruker med hjelpemiddel skal håndtere en viss funksjon, bør forklaringen og funksjonen være sammenkoplet

Ikke aktuelt

Bakgrunn

Iblant er det nødvendig med en forklaring for alvorlig svaksynte brukere om hvordan en funksjon skal brukes. Legg inn en slik forklaring med WAI-ARIA-attributtet aria-describedby eller aria-label:

- Bruk aria-describedby om forklaringen ligger synlig for alle brukere. For eksempel om forklaringen ligger som brødtekst under funksjonen. Om dette er en forklaring til ett skjemaobjekt bør den i stedet ligge i label-elementet. Noter at støtten for aria-describedby er noe varierende i dag, slik at du ikke bare bør belage deg på det.
- Bruk aria-label for å legge inn en forklaring som er skjult for alle brukere utom skjermleserbrukere.

Koblede retningslinjer

WCAG 2.0 - 1.3.1 (A)

WCAG 2.1 - 1.3.1 (A)

Forståbarhet

US10: Forståeligheten er ikke avhengig av brukerens evne til å oppfatte forskjellige farger

Ingen feil funnet

Bakgrunn

Mange ganger er farge en tydelig og funksjonell måte å formidle informasjon på. Rød farge indikerer advarsler og feilmeldinger, blå farge er sterkt forbundet med lenker. Men selv om det er bra å bruke ulike farger for å formidle informasjon, kan du ikke bare basere deg på farge. Omkring 200 000 nordmenn regnes som fargeblinde. Det betyr ikke at alle ser i gråskala, i denne gruppen inngår også for eksempel alle med rød-grønn fargeblindhet, men det er en betydelig gruppe. Dessuten må man ikke regnes som fargeblind for å ha problemer med å oppfatte ulike farger i ulike situasjoner. For eksempel er det betydelig vanskeligere enn normalt å oppfatte fargeendringer i for eksempel løpende tekst om du sitter med mobil eller nettbrett utendørs en solrik dag.

For at ikke brukerens evne til å oppfatte farger skal bli avgjørende, kreves det at du alltid kompletterer med minst én annen måte for å gi informasjonen. Lenker kan for eksempel understrekes, knapper i ulike farger kan utstyres med tekst og så videre.

Det skal heller ikke kreves at brukerens datamaskin er ”riktig” innstilt. I mange nettlesere finnes en innstilling for å ”alltid” understreke lenker, som medfører at om nettstedet ikke har spesifisert om lenker skal være understreket eller ikke, blir lenkene understreket. For en person som låner en datamaskin, for eksempel på biblioteket, er det ikke alltid mulig å påvirke innstillingene, og man kan da ikke være sikker på at ”riktig” innstilling er gjort. Derfor skal du tydelig angi i stilmalene at lenker skal være understreket når det er slik du vil markere lenker som komplement til for eksempel en spesifikk farge.

Koblede retningslinjer

WCAG 2.0 - 1.4.1 (A)

WCAG 2.1 - 1.4.1 (A)

Kvalitet på nett - 5.10

US20: Instruksjoner og henvisninger er forståelig uavhengig av brukerens evne til å se og høre

Ingen feil funnet

Bakgrunn

For at alle besøkere skal forstå instruksjoner og henvisninger er det viktig at dere ikke bare stoler på besøkerens evne til å se eller høre. Å henvise med for eksempel farge kan være en bra måte at tydeliggjøre for brukeren, men henvisningen må kompletteres med annen informasjon. I stedet for å skrive ”klikke på den grønne knappen” kan instruksjonen være ”klikke på den grønne knappen med teksten Aksepter under”. Da finnes det foruten fargen også en tekst og en posisjon som hjelper brukeren med å finne knappen.

Henvis heller ikke til objekt som ligger ”til høyre”. Dels er ikke henvisningen mulig å forstå for brukere som ikke ser grensesnittet. Dessuten havner objekt i blant på andre steder avhengig av skjermbredde i responsive grensesnitt.

De fleste instruksjonsfilmer forutsetter at brukerne både kan se og høre. Slike filmer må derfor alltid kompletteres med en tekstversjon.

Koblede retningslinjer

WCAG 2.0 - 1.3.3 (A)

WCAG 2.1 - 1.3.3 (A)

Web Accessibility Directive (WAD)

WAD10: Tilgjengelighetserklæring

Underkjent

Bakgrunn

Ifølge EUs webdirektiv skal alle nettsteder som omfattes av loven ha en Tilgjengelighetserklæring publisert. Tilgjengelighetserklæring bør være lett å finne på startsiden, men helst på alle sider, for eksempel i bunnteksten. Linken til denne siden hete "Tilgjengelighetserklæring".

Rapporten skal publiseres i et universelt utformet format og primært rette seg mot og hjelpe nettstedets sluttbrukere, men også tilsynsmyndigheten. Rapporten skal inneholde:

- Myndighetens navn.
- Den nasjonale lovens navn.
- Nettstedets navn.
- Krav som ikke oppfyller loven:
- Beskrivelse av og link til Tilbakemeldingsfunksjon.
- Link til tilsynsmyndighetens klagefunksjon.
- Publiserings- og oppdateringsdato for erklæringen.
- Metodevalg for å ta frem Tilgjengelighetserklæringen.

Anbefalinger (ikke krav):

- Link til granskningsrapport.
- Beskrivelse av hvordan organisasjonen jobber med universell utforming.
- Formell godkjennelse på erklæringen fra ansvarlig person i organisasjonen.
- Dato da nettstedet ble publisert.
- Dato for siste betydelige oppdatering eller redesign.
- Spesifikk kommunikasjonstjeneste for personer med funksjonsnedsettelser.

Kommentar

Ingen tilgjengelighetserklæring funnet i appen eller på helsenorge.no

WAD20: Tilbakemeldingsfunksjon

× Underkjent

Bakgrunn

Brukere skal ha mulighet til å påpeke feil og mangler knyttet til universell utforming ved å gi nettstedseier og ansvarsperson direkte tilbakemeldinger.

- Beskriv Tilbakemeldingsfunksjonen.
- Kontaktinformasjon til avdelingen eller personen som er ansvarlig for nettstedets universelle utforming og håndtering av tilbakemeldinger.

Kommentar

Ingen tilgjengelighetserklæring eller mulighet for å rapportere tilgjengelighetsfeil funnet i appen eller på helsenorge.no

WAD30: Klager til tilsynsmyndigheten

× Underkjent

Bakgrunn

Hvis den besøkende har gitt tilbakemeldinger om brudd på universell utforming til nettstedseier uten å få svar, skal man ha mulighet til å registrere klagen hos tilsynsmyndigheten. Den skal beskrive hvordan klager håndteres i Tilgjengelighetserklæringen og det skal lenkes til tilsynsmyndighetens klagefunksjon.

Kommentar

Ingen informasjon eller skjema funnet for å gi tilbakemelding på tilgjenlighetsfeil funnet.

WAD40: Søkefunksjon

– Ikke aktuelt

Bakgrunn

Mange foretrekker å bruke nettstedets søkefunksjon for å finne informasjon. Mens andre har behov for den, for eksempel brukere som syns det er vanskelig å klikke seg gjennom menystrukturen.

Søkefunksjonen er derfor en av de viktigste funksjonene på et nettsted.

Et lite nettsted med begrenset innhold kan klare seg uten en søkefunksjon, men det finnes generelt ingen god grunn til ikke å ha søkefunksjon.

Dette punktet oppsummerer søkefunksjonen basert på de evaluerte punktene ovenfor. For eksempel kontraster, korrekt skjemakode og fungerende WAI-ARIA.

WAD50: Dokumenter

– Ikke aktuelt

Bakgrunn

Universell utforming i ulike dokumenter, som for eksempel PDF-dokumenter, er viktig og må ikke glemmes. I dette punktet gransker vi minst ett dokument per tjeneste/hovedinngang, samt noen tilfeldig valgte.

Illustrasjonen nedenfor viser et eksempel på hvordan en automatisk tilgjengelighetssjekk av et PDF-dokument kan se ut i Adobe Acrobat Pro.

Du har mye å tjene på å stille krav til leverandørene dine om at PDF-dokumentene skal være tilgjengelige for alle. Du må også kvalitetssikre at malene som genererer PDF-dokumentene i de interne systemene, holder et høyt nivå av universell utforming.