

Checklistor för olika roller

För alla

Alla har ett ansvar för att göra information tillgänglig. Vi kommunicerar dagligen med andra genom e-post, videomöten, presentationer, diagram och digitala whiteboards. Därför behöver vi se till att det vi skapar går att förstå för alla. Det gäller oavsett om du skriver ett mejl, gör en presentation eller tar fram ett tekniskt diagram.

- ☐ Använd ett tydligt och koncist språk. Undvik:
 - Långa eller obekanta ord.
 - Komplexa och långa meningar
 - Otydliga formuleringar, såsom förkortningar
- ☐ Undvik textrader som ligger för tätt, ett radavstånd som är lite större förbättrar läsbarheten.
- ☐ Dela upp långa textstycken. Det är bra att strukturera text i avsnitt med rubriker, kortare stycken och listor där det är lämpligt.
- ☐ Använd inte enbart färg för att förmedla något betydelsefullt.
- ☐ Kontrollera att det finns tillräcklig kontrast mellan innehåll och bakgrund. Använd verktyg för att kontrollera detta.
- ☐ De flesta verktyg har inbyggda funktioner för tillgänglighet och stavningskontroll (PowerPoint, Word och Adobe Acrobat Pro). Säkerställ att dessa används när de finns tillgängliga.
- ☐ Undvik bilder som innehåller text.
- ☐ När du skickar skärmbilder, säkerställ att du lägger till beskrivande text som representerar bilden.

När du presenterar

- ☐ Kontrollera om någon behöver materialet i ett annat format, till exempel stor text.
- ☐ Kontrollera att ditt innehåll är synligt. Tänk på att inte ha så mycket text på samma slide. Använd stor text, gärna minst 20 punkters storlek.

- ☐ Kontrollera att du hörs. Använd mikrofon om det finns tillgängligt.
- ☐ Beskriv bilder när det är relevant (om bilden har mer än enbart dekorativ betydelse).
- ☐ Uppmärksamma din publik på att det finns undertexter och syntolkning för media där det är tillämpligt (Teams har inbyggda dynamiska undertexter).

Checklista för utvecklare

Allmänt

- ☐ Följ "best practise" och kodstandard
- ☐ Skriv semantisk HTML-kod
- ☐ Ge stöd för tangentbordsnavigering för allt innehåll
- ☐ Använd etablerade standarder även för egna widgets
- ☐ Var försiktig när du använder WAI-ARIA, använd det rätt.
- ☐ Ta fram en rutin för att välja komponenter från förbyggda kodbibliotek, ramverk etc. Se till att du har en struktur på plats för hur du ska utvärdera varje komponent du funderar på att använda.
- ☐ Se till att det finns textalternativ till innehåll som inte presenteras i text, som bilder och grafik.
- ☐ Skapa en struktur i innehållet med hjälp av rubriknivåer och landmärken.
- ☐ Använd responsiv design
- ☐ Använd JavaScript för progressiv förbättring
- ☐ Använd automatiska verktyg för test och validering
- ☐ Tillåt användaren att zooma till 400%, utan att innehåll försvinner eller överlappas.
- ☐ Undvik rörelse i innehållet, om något rör sig, måste det gå att stänga av rörelsen.

Semantik och struktur

- ☐ Se till att DOM-ordningen är logisk, intuitiv och följer sidans faktiska innehåll och struktur
- ☐ Använd i första hand HTML landmärken.
- ☐ Använd korrekta rubriknivåer som speglar hierarkin på sidan
- ☐ Använd och tillämpa HTML-taggar för sin ursprungliga semantik, när det är möjligt?

- ☐ All struktur och alla relationer som syns visuellt också måste finnas uttryckta i HTML-koden förmedlas genom visuell presentation (i rubriker, listor, menyer, formulär etc.)

Ordning

- ☐ Har läsordningen testats för att säkerställa att allt innehåll på sidan är tillgängligt i en logisk ordning för olika hjälpmedel?"

Användarinställningar

Förändring av teckenstorlek

- ☐ Använd relativa storlekar för teckenstorlek (font-size). Använd rem så att storleken är relativ till webbläsarens grundstorlek för text. Ändring av textstorlek ska fungera både med standardfunktionen för att öka textstorleken och med webbläsarens inställning för att öka teckenstorleken. Använd inte px för font-size.
- ☐ Använd relativa storlekar för radavstånd (line height). Använd relativa värden i stället för en fast storlek.
- ☐ Använd padding för textavstånd och undvik fasta höjder eller bredder så att behållare kan anpassa sig när textstorleken ökar.

Inställning av färger och mörkt läge

- ☐ Låt användaren inställningar i webbläsaren slå igenom på webbsidan. Det gäller Mörkt läge och Högkontrastläge.
- ☐ Tvinga inte ett ljust eller mörkt färgschema om användaren har valt något annat i sina systeminställningar.
- ☐ Blockera inte högkontrastläge eller andra kontrastinställningar genom att låsa färger eller bakgrunder.
- ☐ Säkerställ att text, ikoner och knappar förblir synliga och användbara när mörkt läge eller högkontrastläge är aktiverat.

Tangentbords- och pekskärmsnavigering

- ☐ Säkerställ att all funktionalitet är tillgänglig med enbart tangentbord, att tabbordningen är logisk.
- ☐ Säkerställ att innehåll är tillgängligt med enbart tangentbord, att det går att nå med tangentbordet och att all funktionalitet kan styras med tangentbord.

- ☐ Tabbfokus ska vara tydligt vid tangentbordsnavigering, fokus ska också uppfylla kontrastkrav och alltid vara synligt, alltså inte vid något tillfälle hamna bakom annat innehåll.
- ☐ Se till att fokusordningen följer en logisk ordning, samma ordning som innehållet som visas visuellt.
- ☐ Säkerställ att alla interaktioner med enbart tangentbord och pekskärm följer förväntade mönster, så att användare förstår hur de ska interagera med alla widgets på sidan.
- ☐ Undvik att tangentbordsanvändare fastnar med tabben i något läge, alltså undvik att de fastnar i en tangentbordsfälla.

Tabeller

- ☐ Koda tabeller korrekt med dataceller, td, och rubrikceller, th.
- ☐ Koppla dataceller till rubrikceller så att skärmläsaranvändare kan navigera effektivt i tabeller.
- ☐ "Are data tables marked up to convey the correct relationships between data cells and their associated column or row header cells?"
- ☐ Add a caption(title) that indicates what the table data is about
- ☐ Tabeller ska användas för data, inte för layout. Om en tabell används för layout ska tabellen kunna läsas radvis och den får inte innehålla kolumn eller radrubriker, th. Den ska inte heller ha någon caption-tag. Lägg gärna till `<table role="presentation">`

Läs mer: <https://www.w3.org/WAI/tutorials/tables/>

Formulär och interaktiva element

- ☐ Varje inmatningsfält ska ha en etikett, helst synlig. Har formulärfält synliga etiketter/labels, och är etiketten och fältet korrekt kopplade i koden?
- ☐ Använd korrekta input-typer för formulärfält, helst i HTML.
- ☐ Koppla ytterligare instruktioner till varje formulärfält direkt i koden, där det behövs.
- ☐ Gruppera relaterade formulärfält, exempelvis kryssrutor och alternativknappar, företräddelsevis med fieldset och legend.
- ☐ Använd autocomplete-attribut för att underlätta ifyllnad av vanliga fält som exempelvis namn och adress.

Felhantering

- ☐ Är felmeddelanden korrekt kopplade till sina formulärfält i koden?
- ☐ Uppmärksammas skärmläsaranvändare på felmeddelande som visas, även om inte fokus är på felmeddelandet?
- ☐ Om ett valideringsfel uppstår under användarens inmatning eller när fokus flyttas, uppmärksammas skärmläsaranvändare på felet?
- ☐ Om det finns flera fel, flytta fokus till ett samlingsmeddelande med alla fel, alternativt flytta fokus till det första felande fältet.
- ☐ Framgår det av felmeddelandet hur användaren ska rätta till felet. Ge ett korrigeringsförslag.

Egna kontroller och widgets

- ☐ Säkerställ att alla egna kontroller och scriptade komponenter har namn och roller som motsvarar inbyggda HTML-element. ("programmatically determinable")
- ☐ Baseras egna widgets på etablerade designmönster och beprövade interaktionsmodeller?
- ☐ Följ WAI-ARIA Authoring Practices när ni använder ARIA-roller, tillstånd och egenskaper.
- ☐ Tilldela korrekta roller, tillstånd och egenskaper så att widgetens funktion och status framgår för hjälpmedel.

- ☐ Skapa vid behov egna namn och etiketter för komplexa element, så att deras syfte och funktion framgår tydligt.

Kontextförändringar och Tidsbegränsningar

Kontextförändringar och dynamiskt innehåll

- ☐ Säkerställ att nya områden, modaler och vyer får fokus när de öppnas. Till exempel att React Router sätter fokus korrekt vid navigering mellan sidor.
- ☐ Se till att statusmeddelanden och feedback annonseras för skärmläsaranvändare
- ☐ Kontrollera att ingen större kontextförändring utlöses automatiskt när ett element får fokus.
- ☐ Kontrollera att ingen större kontextförändring sker när användaren ändrar inställningen för en komponent – om det inte tydligt kommunicerats i förväg.

Rörelse och tidsbegränsningar

- ☐ För innehåll som rör sig, blinkar, uppdateras automatiskt eller rullar – säkerställ att användaren kan stoppa, pausa, dölja eller kontrollera det.
- ☐ Om en tidsbegränsning finns för att slutföra en uppgift, ska användaren kunna stänga av, justera eller förlänga tiden om inte tidsbegränsningen är längre än 20 timmar.

Bilder och grafik

- ☐ Säkerställ att alla `<img>`-element har korrekta alternativtexter (alt-attribut). Det gäller länkade och informativa bilder. För dekorativa bilder ska alt-attributet lämnas tomt (alt = "").
- ☐ Säkerställ att alla informativa bakgrundsbilder har alternativ text som kan läsas av hjälpmedel.
- ☐ Dölj dekorativa bilder från skärmläsare.
- ☐ Undvik att använda bilder av text.
- ☐ Beskriv flödesscheman och liknande grafik i klartext, exempelvis som en steg-för-steg-lista.

Hanterbart med tangentbord och skärmläsare

Hanterbart med tangentbord

- ☐ Säkerställ att alla interaktiva element kan nås med tangentbord.
- ☐ Kontrollera att fokusordningen är logisk och följer sidans struktur.
- ☐ Använd aldrig positiva tabindex-värden.
- ☐ Använd tydligt tabbfokus på alla fokusbara element. Tabbfokus ska uppfylla kontrastvärde för komponenter.
- ☐ Testa att fokusmarkeringen är synlig på samtliga komponenter. De får inte vara gömda bakom något.

Hanterbart med skärmläsare

- ☐ Kontrollera att allt innehåll är tillgängligt med skärmläsare och att alla komponenter har maskinläsbara namn.
- ☐ Säkerställ att visuella tillstånd, statusmeddelanden och relationer förmedlas till skärmläsaren.
- ☐ Se till att skärmläsare inte kan hitta eller läsa dolda element.

Använd verktyg för automatiska tester

Automatiska tester hittar inte alla fel, men kan vara en hjälp att hitta tillgänglighetsproblem.

- ☐ Använd automatiska verktyg för att kontrollera kodkvalitet och validering.
- ☐ Testa koden för vanliga fel med verktyg som aXe, Wave, Accessibility insights eller andra verktyg för att testa tillgänglighet.
- ☐ Utnyttja tillgänglighetsfunktioner och tillägg i din IDE (kodeditor).

Modaler och dialoger

- ☐ Sätt fokus på ett element inuti modalen när den öppnas (t.ex. den första rubriken).
- ☐ Begränsa fokus till modalen så länge den är öppen.
- ☐ Säkerställ att stängningsknappen kan aktiveras med tangentbord.
- ☐ Tillåt att modalen kan stängas med Esc-tangenten.

- ☐ Ange role="dialog" och aria-modal="true" för att markera modalen korrekt.
- ☐ När modalen stängs, återställ fokus till det element som öppnade modalen.

Karuseller

Undvik karuseller om möjligt, eftersom de ofta ökar komplexiteten utan att tillföra tillräckligt värde.

- ☐ Säkerställ att föregående- och nästa-knappar finns och har tydliga, tillgängliga etiketter (t.ex. via aria-label om ikoner används).
- ☐ Placera slides i en -lista och gör varje slide fokuserbar så att innehållet kan läsas med tangentbord och skärmläsare.
- ☐ Placera bullets/indikatorer i en -lista och gör dem klickbara samt fokuserbara, med tydlig markering, både visuellt och semantiskt av vilken slide som är aktiv.
- ☐ Säkerställ att karusellen fungerar med tangentbord, inklusive att alla interaktioner följer förväntade mönster.
- ☐ På mobila enheter: säkerställ att användaren kan navigera via swipe eller via föregående/nästa-knappar.

Följ rekommendationerna från W3C:s exempel för tillgängliga karuseller.

www.w3.org/WAI/tutorials/carousels/working-example

Checklista för projektledare

Förespråka tillgänglighet

- ☐ Identifiera tillgänglighet som ett av projektets strategiska mål och involvera det tekniska teamet i diskussioner om vad målet innebär och varför det är viktigt. Förespråka att uppnå minst nivå AA i WCAGs riktlinjer, samt att uppfylla krav som finns i EN301 549.
- ☐ Det kan vara bra att visa exempel på andra organisationer som redan har skapat tillgängliga och lättanvända webbplatser. Det hjälper teamet att förstå vad som fungerar bra och vad de kan sträva efter.
- ☐ Kommunicera tydligt vart teamet kan vända sig för att få stöd från personer med kunskap om tillgänglighet.
- ☐ Ta fram och dela exempel på riskerna med bristande tillgänglighet, som förlorade kunder, negativ publicitet och rättsliga åtgärder.
- ☐ Koppla tillgänglighet till andra fördelar. Det är ofta värdefullt att visa att tillgänglighet ger många positiva effekter utöver att göra webben användbar för personer med funktionsnedsättning, exempelvis förbättrad SEO och en bättre användarupplevelse för alla.

Planering

- ☐ Avsätt tid för testning, åtgärder och omtestning.
- ☐ Tillgänglighet kräver uppföljning och underhåll, precis som allt annat.
- ☐ Säkerställ att ni har rätt tillgänglighetskompetens.
- ☐ Involvera tillgänglighetsexperter tidigt och regelbundet. De kan snabbt upptäcka risker och hjälpa er undvika onödigt extraarbete.
- ☐ Identifiera era största tillgänglighetsrisker.
- ☐ Ha en tydlig rutin för hur ni väljer komponenter från kodbibliotek och ramverk. Se också till att ni har en struktur för att bedöma varje komponent utifrån tillgänglighet innan ni använder den.
- ☐ Inkludera tillgänglighet i "definition of done" och acceptanskriterier.
- ☐ Användartester: Låt produkten löpande utvärderas av externa personer som använder olika sätt att ta del av innehållet (endast

tangentbord, skärmläsare, färgblindhet med mera), så snart det finns fungerande utkast eller prototyper.

- ☐ Prioritera publicering av en tillgänglighetsredogörelse.
- ☐ Se till att det finns tid och resurser för att utbilda organisationen. Ansvar för utbildningen ska inte ligga enbart på projektledaren, utan ska delas tydligt mellan utsedda personer.

Checklista för designers

Hjälp ditt team att förstå varför tillgänglighet är viktigt

- ☐ Hitta ett verktyg för att simulera funktionsnedsättningar, till exempel Silktide.
- ☐ Bjud in teamet att genomföra användbarhetstester med användare som har funktionsnedsättningar.

Innehåll på webbplatsen

Sidtitel

- ☐ Beskriver sidtiteln (webbläsarens titel för sidan, inte huvudrubriken eller <h1>) sidans ämne eller syfte?

Rubriker

- ☐ Är all text som ser ut och fungerar som en rubrik markerad som en rubrik?
- ☐ Används rubriknivåerna så att de förmedlar korrekt hierarkisk ordning i innehållet, och inte enbart för visuell styling?

Navigation

- ☐ Finns det en metod för att hoppa över repetitiv navigation och gå direkt till huvudinnehållet?
- ☐ Är innehållets läsordning och fokusordning, enligt kodens struktur, logisk och intuitiv?

Länkar

- ☐ Beskriver länktextern tydligt länkens syfte eller destination? Länken bör kunna plockas ur sitt sammanhang och ändå gå att förstå.

Färg och kontrast

- ☐ Använd inte bara färg för att visa information. Kombinera färg med text, till exempel för att visa fel i formulär eller för att märka kategorier i diagram.
- ☐ Har all text en minsta färgkontrast mot sin bakgrund på minst 4,5:1 (3:1 för stor text)?
- ☐ Skiljer sig länktext från övrig text med mer än enbart färg?

- ☐ Har interaktiva element en tydlig och synlig fokusmarkering när användare navigerar med Tab eller piltangenter?
- ☐ Används tillräcklig kontrast på gränssnittskomponenter och meningsbärande ikoner?

Bilder

- ☐ Ger alternativtexten för informativa bilder samma information som bilden?
- ☐ Beskriver alternativtexten, för interaktiva bilder som bildlänkar, knappar eller bildkartor, tydligt vart länken leder eller vad knappen gör?
- ☐ Förklaras komplexa bilder eller infografik fullständigt i sidans innehåll samt med en kort alternativtext?
- ☐ Är dekorativa bilder korrekt markerade som sådana i koden?
- ☐ Används vanlig text i stället för text inbäddad i bilder?
Undantag kan finnas, exempelvis text i logotyper och dekorativa textbilder eller skärmbilder.

Fokusmarkering

- ☐ Utforma och testa en tydlig fokusmarkering med hög kontrast, minst 3:1. Kontrollera att kontrastkraven mot bakgrunden uppfylls i de flesta fall.
- ☐ Kantlinjer. För att uppfylla WCAG 2.2:s nya krav på fokusindikatorer: säkerställ att kantlinjen är minst 1 px bred.

Tabeller

- ☐ Om sidan innehåller en datatabell, är kolumner och/eller rader korrekt uppmärkta?
- ☐ Har komplexa tabeller förenklats för att minimera eller eliminera behovet av sammansatta kolumn- eller radhuvuden?
- ☐ Är tabellen tydlig, visuellt? Kanske behövs skuggning av varannan rad, eller annat sätt som hjälper ögat att hålla sig till rätt rad. Detta är viktigast i tabeller med många dataceller.

Formulär

- ☐ Har alla formulärfält en etikett som alltid är synlig?
- ☐ Är alla formuläretiketter tillräckligt beskrivande och tydliga? Finns all information som användaren behöver för att fylla i formuläret tillgänglig på sidan?

- ☐ Är alla formuläretiketter och instruktioner placerade direkt intill sitt formulärfält så att användare (inklusive användare av skärmförstoring) enkelt kan koppla fältet till dess etikett och/eller instruktioner?
- ☐ Är alla kontroller placerade i nära anslutning till det innehåll de styr? Till exempel, är knapparna "Redigera" och "Ta bort" placerade bredvid det innehåll de ändrar?
- ☐ Ger felmeddelanden tillräcklig information för att användare ska kunna rätta sitt fel?

Dynamiskt innehåll

- ☐ Meddelas användaren när nytt innehåll läggs in på sidan, eller placeras det nya innehållet direkt efter det element som utlöste det, i en logisk läs- och tabbordning?
- ☐ Fungerar alla tangentbords- och pekskärmsinteraktioner på ett sätt som användare förväntar sig, så att de lätt förstår hur de ska använda alla komponenter på sidan?

Design för tangentbord och hjälpmedel

Inmatningsenheter

- ☐ Designa för tangentbordsinteraktion
- ☐ Tillhandahåll synliga fokusmarkeringar
- ☐ Säkerställ tillräckligt stora klick- och trycktor för mus- och pekanvändare

Design för hjälpmedel

- ☐ Tillhandahåll instruktioner för hur man interagerar med komplexa komponenter
- ☐ Skriv användbara etiketter för skärmläsaranvändare

Layout

- ☐ Skapa en sidlayout som är förutsägbar och lätt att följa. Undvik att användaren måste hoppa mellan innehåll på olika delar av skärmen.

Tydlig interaktion

- ☐ Utforma interaktiva element så att de är enkla att förstå och interagera med

Rörelse och tidsbegränsningar

- ☐ Undvik onödig rörelse och animation som användaren inte kan stänga av.
- ☐ Undvik tidsgränser och timeouter som användaren inte kan skjuta upp eller avbryta.

Innehåll och instruktioner

- ☐ Skriv allt innehåll på ett tydligt och begripligt sätt
- ☐ Använd begripliga länk- och knapptexter
- ☐ Använd bilder, video och ljud som komplement till text när det är lämpligt
- ☐ Tillhandahåll hjälpsamma felmeddelanden och instruktioner