## Jiotta:

Powerpoint template?

## Over a11y

Algemeen

* Over mij
  + Projecten
  + Certification
* Wat betekent A11y
* Misvattingen
  + Screenreaders ondersteunen javascript
  + Het kost niet meer tijd, wel extra aandacht
  + Je doet het niet alleen voor blinden- of slechtzienden
  + Het is meer dan aria tags uit je hoofd leren
* Doel van de presentatie
  + Overtuigen van het nut
  + Enthousiast maken voor a11y
  + Dat a11y nu een standaard wordt voor componenten
  + Doel is niet dat we vanaf nu a11y sites op gaan leveren of gaan refactoren. Daar moet de klant voor betalen. Het is makkelijker om een nieuwe site om te zetten, dan een bestaande site. Zelfde geldt voor seo en responsive design.
* Nut en noodzaak
  + Site laagdrempeling maken
  + Doelgroep verbreden
  + Fix for one, solve for many
  + Semantisch beter opgezet
  + Site is futureproof (controllers, voice assistants)
* Introductie
  + Doelgroep
  + Toepassingen
  + Use cases
  + wat zijn screenreaders
  + POUR
* Hoe de klant te overtuigen
* Verantwoordelijkheid
  + Frontend
  + Design/Ux
  + Content

Technisch

* Richtlijnen wcag 2.1
  + Regels
  + Voorbeelden
* Hoe te testen
  + Software op mac en windows

Opdracht lelijke code optimaliseren (30? min)

## High level componenten (codepen)

* Documentatiestructuur
  + h1 – h6
  + sr-only
  + Buttons vs anchors
* Formulieren opzet
  + Validatie
  + Uitleg
  + Gebruik van placeholders
* Afbeeldingen
* Iconen
  + Indexeerbaarheid
* Sitemap

## In depth components

* Caroussel
* Overlay
* Video player
* Bring your own component

## Doel

Ervoor te zorgen dat de interface bruikbaarder is voor meer en diverse mensen.

## Toetsenbord:

Zorg ervoor dat de site prima benaderbaar is met het toetsenboard.

* Voor de poweruser
* Futureproof: over het algemeen zorg je er dan indirect voor dat hij goed te bedienen is bijv met stem of met een joystick
* touchdevices

### Maak het pour

* perceivable (ondertiteling, contrast, screenreader)
* operable (toetsenboard, stem)
* understandable (schrijf b1 niveau: korte zinnen, geen bijzinnen)
* robust (oude apparaten)

https://www.accessibility.nl/kennisbank/artikelen/eenvoudige-beschrijving-toegankelijkheidsrichtlijnen

Verschil tussen A/AA/AAA

Om het keurmerk drempelvrij te zijn, moet de site volledige toegankelijk zijn. Alle elementen op de site moeten voldoen aan de regels. Ook alle subdomeinen moeten hieraan voldoen, bijv de mijn omgeving en de

## Aria

https://www.accessibility.nl/kennisbank/artikelen/wai-aria-en-toegankelijkheid

# Screenreader

Screenreader leest de pagina voor.

Met sneltoetsen kun je filteren op bepaalde tags

* h voor headers
* k voor links

Screenreaders ondersteunen javascript

MacOs en iOs: VoiceOver

Google: Voice assistent

Windows: Narrator

Overig: JAWS, NVDA, Supernova

Screenreaders ondersteunen niet altijd dezelfde tags

<https://webaim.org/projects/screenreadersurvey7/#usage>

CMD + F5: switch voiceover

OPTION+CONTROL = VO

VO+Command+H = browse headings

VO+links/recht = browse controls

VO+space = click

VO+down = interact with site

VO+U = Rotor: landmark: section, main, nav

Meeste gebruikers navigeren via headings

# Basis

## Typography

Zorg voor de juiste headers. Een screenreader kan de tekst scannen door alle kopjes na te gaan. Als deze niet op de juiste symantiek staat, is de tekst onleesbaar.

Gebruik liever relatieve eenheden dan absolute eenheden als je in mark-uptalen waarden toekent aan attributen en eigenschappen in style sheets.

Tekst kan tot 200% vergroot worden zonder verlies van info en functionaliteit.

Vergroot/verklein knop is geen alternatief voor schaalmogelijkheden in typografie. Eigenlijk overbodig.

## HTML

Elke pagina bevat 1 Main nodig met de content.

Skiplinks

Gebruik een class sr-only voor elementen die alleen voor screenreaders leesbaar moeten zijn. Elementen die op Display none of visibility hidden staan, staan wel in de dom, maar worden ook overgeslagen door screenreaders.

Zet de juiste taal als lang attribute op de html tag lang=”en”

Test de symantiek van de site door de css uit te zetten

## Valide code

Websites met louter valide pagina's zijn niet automatisch toegankelijk. Omgekeerd kunnen invalide pagina's wel toegankelijk zijn.

* Elementen moeten correct geopend en gesloten zijn;
* Elementen zijn niet verkeerd genest;
* Elke ID is uniek;
* Attributen zijn correct opgemaakt;
* Attributen komen slechts 1 x per element voor.

## Buttons

Gebruik buttons bij interactieve elementen (toggle modal, open menu) die worden getriggered door js en anchor tags voor navigatie tussen pagina’s of anchors.

Buttons hebben een focusstate nodig, om aan te geven waar de gebruiker is bij gebruik van een toetsenboard.

Links hebben zowel een onderscheiding nodig met een kleur en met een lijntje. Voor een hover/active/visited state is een kleuronderscheid voldoende.

# Design

Gebruik de juiste contrast waarden.

Verschilt per fontsize:

Het contrast van normale tekst en afbeeldingen van tekst is 4,5:1.

Het contrast van grote tekst (18pt of 14pt dikgedrukt) en afbeeldingen van grote tekst is 3:1.

Testen: https://webaim.org/resources/contrastchecker/

# Video

Videos die autoplay gaan moeten de mogelijkheid hebben om te pauzeren (behalve decoratieve videos)

Videos moeten ondertitels hebben.

Videos moeten een alternatief hebben, in het geval de video niet werkt

* volledige uitgeschreven transcriptie
* stel de video ter beschikking, inclusief caption

Overzicht van de captionvereisten per type video: <https://www.accessibility.nl/kennisbank/artikelen/audiovideo>

# Content

Voorkom knoppen als ‘meer informatie’ of ‘lees verder’, maar gebruik duidelijke labels voor buttons. Bijv, ‘lees meer over de vw passat’, of ‘ga naar adresgegevens’. Bij ruimtegebrek kun je de tekst ‘meer info’ aanvullen met een sr-only span.

Links moeten minimaal via 2 ingangen te benaderen zijn. Een makkelijke manier om dat de omzeilen is het genereren van een sitemap.

Gebruik duidelijk en korte zinnen.

# PDF

PDFs moeten indexeerbaar zijn.

Ook goed voor de SEO

Eenvoudig te check door de tekst te selecteren

PDF moet ook semantisch correcnt zijn:

* tabellen moeten tabellen zijn
* lijsten
* functionele afbeeldingen moeten een beschrijving hebben.

# Afbeeldingen

Gebruik alt teksten om de inhoud van de afbeelding te beschrijven, niet om de titel van het item/de pagina te kopiëren.

Gebruik een lege alt tekst voor decoratieve afbeeldingen.

Zorg ervoor dat de alt tekst na de titel wordt getoond. Dit kun je doen met flexbox.

Gebruik geen iconfonts voor icons. Deze zijn onleesbaar voor screenreaders. Gebruik svg. Zet een title tag bij functionele iconen, houd het leeg bij decoratieve iconen. Leg uit wat het icoon doet/is bij functionele iconen. Bijv niet ‘friendface’, maar ‘Volg volkswagen op friendface’. Voeg daarbij ook een aria-labelledby attribute op de svg met een verwijzing naar die title. Bijv:

*<svg aria-labelledby="svgtitle1"> <title id="svgtitle1">Settings</title> [other svg code] </svg>*

# CSS

Interessante mediaqueries.

# Javascript

Screenreaders ondersteunen javascript.

Externe APIs moeten ook toegankelijk zijn, zoals twitterfeeds

Test javascript apis ook op accessibility.

# Formulieren

Gebruik geen tijdslimiet voor formulieren. Als dat nodig is, voor veiligheidsredenen, biedt de mogelijkheid aan om dit uit te zetten of aan te passen.

Gebruik een submit button, om ervoor te zorgen dat het formulier ook wordt gesubmit met enter. Hoeft niet als het formulier maar 1 veld heeft.

## Label

Gebruik een for attribute bij labels en koppel deze aan het formulierelement met het id. Daardoor is het label klikbaar en zet het de focus op het veld.

Indien het label niet getoond hoeft te worden, gebruik dan een label met class sr-only.

De tag title wordt niet als zodanig geindexeerd door screenreaders.

Plaats labels voor de inputvelden. Voor checkboxes en radiobuttons is het gebruikelijk om dat andersom te dom.

Plaats labels visueel boven inputvelden en niet ernaast. Renske en Joost hebben hier namens PON een UXdocument voor geschreven.

Placeholders worden niet voorgelezen door screenreaders. Gebruik dus nooit placeholders als labels. Indien dit een vereiste is van de ux-er, animeer de placeholder als labels bij focus.

## Captcha

Moet een alternatief aanbieden, bijv in audio formaat

Moet uiteraard met het toetsenboard te benaderen zijn

## Groeperen

Groepeer velden met een fieldset (naw, radiobutttons)

Gebruik een legend (en zet deze desnoods op sr-only). Sommige screenreaders lezen bij elk veld de legend voor, dus houd de legend kort.

Een tekst zoals “NAW gegevens” is niet belangrijk om te begrijpen wat je bij een online winkel in het veld “Woonplaats” moet invullen. NAW gegevens kan daarom beter als kop worden aangegeven. Een tekst zoals “Geboortedatum” daarentegen is wel belangrijk om genoemd te hebben bij bij elkaar horende velden met de labels "Dag", “Maand” en "Jaar". Daarvoor is legend de betere oplossing.

## Instructies

…

## ARIA

Met ariatags kun je standaard html attributen overschrijven.

…

## Table

Moet een caption hebben, direct na de <table>. Dat bevat de titel van de tabel.

Koppen van rijen/collumns zijn th

## Menu

…

## Caroussel

Caroussels zijn pittig om te optimaliseren. Gebruik liever dialog/summary. Deze is out of the box toegankelijk en is prima te stijlen.

Nadeel

* Je kunt (volgens mij) niet animeren)
* Werkt niet in IE11 en edge. Dan wordt alles direct weergegeven. Misschien komt edge met een update, nu ze naar Blink zijn overgestapt?

Indien deze toch zelf geschreven moet worden:

* Test deze goed op toetsenboard events.
* Zet de juiste aria tags om aan te geven dat er content is opengeklapt. Zet de focus op het nieuwe element?
* Gebruik nooit een accerdoen waar de opengeklapte item sluit bij het openen van een nieuw item. Voegt niets toe.

… voorbeeld

## Dialog

* Zorg dat deze te sluiten is met de escape key en bij het klikken op de achtergrond
* Behoud de focus binnen de modal. Zorg dat de focus niet de modal verlaat bij het indien de modal open is.

…

## Autocomplete

Zorg dat je door de items kunt navigeren met het toetsenboard.

# Carousel

…