

INCLUSIEVE AANBEVELINGEN EN TECHNISCH
REFERENTIEKADER OM DIGITALE OVERHEIDSDIENSTEN
TE VERBETEREN EN TE OPTIMALISEREN VOOR MENSEN
VOOR WIE EEN DIGITALE KLOOF DREIGT

_ DIGITALE INCLUSIE

INHOUD

I.	Inleiding.....	2
II.	Deel I: Aanbevelingen voor inclusieve digitale overheidsdiensten	4
A.	Instandhouding van alternatieven voor digitale kanalen	4
B.	Digitale toegankelijkheid	5
C.	Lay-out.....	6
D.	Structuur en navigatie	9
E.	Redactionele inhoud	13
F.	Een profiel aanmaken, aanmelden en afmelden	16
G.	Informatie invoeren en formulieren invullen.....	18
H.	Documenten uploaden en downloaden	21
I.	Betalende procedures uitvoeren	22
J.	Multimedia (video- en geluidsbestanden).....	23
K.	Een afspraak maken.....	24
L.	Contact	25
M.	Proces voor de ontwikkeling van DOD's	26
N.	Een testplatform voor elke DOD.....	27
III.	Deel II: Technisch referentiekader voor IT-ontwikkelaars.....	28
O.	Toegankelijke formulieren	28
P.	Semantiek en structuur	86
Q.	Tabellen.....	115
R.	Dynamische componenten.....	134
S.	Afbeeldingen.....	143
IV.	Conclusie.....	153
V.	Checklist voor toegankelijke en inclusieve DOD's	154

I. Inleiding

Paradigm heeft CAWaB de opdracht gegeven om inclusieve aanbevelingen te formuleren in het kader van de ontwikkeling van digitale overheidsdiensten (DOD's) in het Brussels Hoofdstedelijk Gewest.

In een eerste verslag werd een grondige analyse gemaakt van het gebruik van twee DOD's (MyActiris en IRISbox) door twaalf doelgroepen waarvoor een digitale kloof dreigt, als volgt verdeeld:

- 1) **4 mensen met dyslexie;**
- 2) **4 mensen met een verstandelijke beperking;**
- 3) **6 blinden:** blinde mensen die spraaksynthese gebruiken;
- 4) **2 slechtzienden:** slechtziende mensen die spraaksynthese- en/of vergrotings-/contrastsoftware gebruiken;
- 5) **7 doven:** mensen die vergaand doof zijn;
- 6) **2 slechthorenden:** mensen die een gehoorapparaat dragen;
- 7) **5 toetsenbordgebruikers:** mensen die enkel een toetsenbord gebruiken om op het internet te surfen;
- 8) **5 ouderen:** mensen boven 65 jaar¹ die thuis geen computer hebben;
- 9) **4 anderstaligen:** mensen die geen Frans spreken of begrijpen. Omdat er tolken voorhanden moesten zijn, is hier gekozen voor het Spaans en het Arabisch;
- 10) **4 kansarmen:** mensen met weinig financiële middelen die in het bezit zijn van een getuigschrift lager onderwijs;
- 11) **5 mensen in een alfabetiseringstraject:** mensen die schriftelijke en geschreven noties hebben van het Frans;
- 12) **7 mensen op een smartphone:** mensen die enkel een smartphone hebben om online te gaan, waaronder twee slechtzienden die spraaksynthese gebruiken.

Elke doelgroep bevatte ten minste één werkzoekende, één jongere onder 25 en één vrouw.

Om deze doelgroepen te bereiken werd een beroep gedaan op een netwerk van 23 aangesloten verenigingen.

Er zijn ook twee controlegroepen samengesteld. Deze bestonden uit mensen voor wie geen digitale kloof zou dreigen, om hun ervaringen te kunnen vergelijken met die van de twaalf bovengenoemde doelgroepen:

- 1) **6 mensen voor de controlegroep met een computer**
- 2) **3 mensen voor de controlegroep met een smartphone**

In totaal hebben 64 burgers deelgenomen.

Deze studie heeft het mogelijk gemaakt de specifieke behoeften van bepaalde doelgroepen bij het surfen op websites van overheidsdiensten objectief te beoordelen, maar ook veel meer algemene lessen te trekken.

¹ Er is één uitzondering gemaakt voor iemand die uiteindelijk 55 jaar bleek te zijn.

Een eerste vaststelling is duidelijk: de normen voor digitale toegankelijkheid opgelegd door de ordonnantie van 4 oktober 2018 inzake de toegankelijkheid van de websites en mobiele applicaties van de gewestelijke overheidsinstanties en de gemeenten², beantwoorden zelfs wanneer correct toegepast - wat helaas nog niet al te vaak het geval is - niet aan de behoeften van de gehele bevolking.

Dit verslag omvat dus 38 aanbevelingen die de toegankelijkheidsnormen (WCAG 2.1 AA) aanvullen om online overheidsdiensten inclusief te maken voor alle burgers.

Naast die aanbevelingen bevat het tweede deel van dit verslag een aanvullend referentiekader met technische criteria voor IT-ontwikkelaars om de aanbevelingen toe te passen.

² Deze ordonnantie is een omzetting van Richtlijn (EU) 2016/2102 van het Europees Parlement en de Raad van 26 oktober 2016 inzake de toegankelijkheid van de websites en mobiele applicaties van overheidsinstanties.

II. Deel I: Aanbevelingen voor inclusieve digitale overheidsdiensten

A. Instandhouding van alternatieven voor digitale kanalen

Aanbeveling 1: Naast de onlinediensten hebben de burgers de mogelijkheid om voor elke vraag naar een fysiek loket te gaan, waar een ambtenaar is die hen kan helpen met hun procedures

Alle gebruikers wensen dat het menselijk contact behouden blijft, hetzij om hen te helpen met hun administratieve procedures, hetzij bij technische problemen op de digitale platformen.

Deze maatregel wordt ook naar voren gebracht in de recente studie van Lire et Ecrire³: *“Een dicht netwerk van loketten moet in stand worden gehouden om een einde te maken aan de sociale ongelijkheden die zijn versterkt door zowel de gezondheidscrisis als door de digitalisering die daardoor in een stroomversnelling is gekomen. Ambtenaren bij eerstelijns overheidsdiensten hebben ervaring en knowhow die niet zo gemakkelijk kan worden vervangen door een instrument of door een digitale hulp die niet op de hoogte kan zijn van alle details in de administratieve procedures voor belastingaangiften, uitkeringsaanvragen of schoolinschrijvingen ...”*.

Als men voor deze diensten een afspraak moet maken, mag dit overeenkomstig aanbeveling 35 niet alleen online kunnen.

Het is aan te bevelen dat het personeel aan de loketten op de hoogte is van en getraind is in het onthaal van personen met een handicap (en de verschillende soorten handicaps). Aan deze loketten moet de informatie aan doven in gebarentaal (persoonlijk of op afstand) worden aangeboden, zodat zij met het personeel van de overheidsdienst kunnen communiceren.

³ Louise Culot, “Accessibilité numérique pour les personnes illettrées : la loi ne garantit pas l’égalité”, Lire et Ecrire Franstalige Gemeenschap, februari 2022.

B. Digitale toegankelijkheid

[Aanbeveling 2: De site voldoet aan de normen voor digitale toegankelijkheid en bevat een conforme toegankelijkheidsverklaring, opgesteld of gevalideerd door een bevoegde instantie](#)

Krachtens [de ordonnantie van 4 oktober 2018 inzake de toegankelijkheid van de websites en mobiele applicaties van de gewestelijke overheidsinstanties en de gemeenten](#) moet de openbare website de normen voor digitale toegankelijkheid WCAG 2.1 AA in acht nemen.

In dit verslag komen enkele technische aspecten en criteria aan bod, maar deze zijn niet uitputtend. Toch is een volledige analyse van de sites met betrekking tot de normen voor digitale toegankelijkheid van essentieel belang, aangezien dit een onmisbare basis vormt voor de toegankelijkheid van websites.

Om dit doel te bereiken is het raadzaam zich vanaf het begin van het ontwerp te laten begeleiden door een deskundige organisatie.

Bovendien moet, wanneer de website online komt, [de toegankelijkheidsverklaring](#) daarop staan. Deze moet worden opgesteld of gevalideerd door een organisatie met deskundigheid op het gebied van digitale toegankelijkheid.

Om verder te gaan

[Technisch referentiekader in het tweede deel van dit verslag](#)

C. Lay-out

[Aanbeveling 3: Uniformiseer de lay-out van alle digitale overheidsdiensten](#)

Wanneer burgers iets moeten opzoeken, aan- of afmelden, toegang willen krijgen tot hun profiel, hun taal moeten kiezen... is het gemakkelijker indien zij op de verschillende online overheidsdiensten op dezelfde plaatsen naar een link of een knop kunnen zoeken.

Opdat gebruikers elementen systematisch op dezelfde plaatsen terug zouden vinden, wordt aanbevolen voor alle online overheidsdiensten een uniforme lay-out te volgen. Alle basisprocedures waaruit een DOD is opgebouwd (aanmelden, afmelden, zoekmachine, persoonlijk profiel, afspraak maken, hulp en contact, de sitemap enz.) moeten steeds op een identieke plaats staan.

Uniformiteit in de structuur en de lay-out van de DOD's zal het voor de gebruikers eenvoudiger maken om te leren hoe zij deze sites moeten gebruiken en om de stappen voor het starten van een administratieve procedure gemakkelijker en sneller te onthouden.

Om verder te gaan

[Technische hulp over navigatie binnen de pagina](#)

[WCAG 2.1 AA – Consistente navigatie – criterium 3.2.3 \(externe link\)](#)

[Aanbeveling 4: De kleur van een tekst moet voldoende contrasteren met de achtergrondkleur](#)

Contrasten zijn zeer belangrijk om te zorgen dat slechtzienden, ouderen of iedereen die niet voldoende licht heeft de inhoud kan lezen.

Deze aanbeveling is ook van toepassing op bijvoorbeeld ondertitels bij videobeelden.

Hetzelfde geldt voor de focus van de browser: de zichtbaarheid ervan moet op alle onderdelen gewaarborgd zijn en mag niet worden belet door schaduwen en achtergrondafbeeldingen met dezelfde kleur als de focus.

Om verder te gaan

[WCAG 2.1 AA – Contrast \(minimum\) – criterium 1.4.3 \(externe link\)](#)

[Application Colour Contrast Analyser – \(externe link\)](#)

[Aanbeveling 5: Gebruik een toegankelijk lettertype en zorg ervoor dat de tekengrootte kan worden gewijzigd](#)

Een te kleine tekengrootte is extra vermoeiend voor ziende en slechtziende gebruikers die hun administratieve procedures online moeten afhandelen. Daarom is het beter om met grote tekengroottes te werken.

Bovendien is een schreefloos lettertype ideaal, zodat bijvoorbeeld mensen met dyslexie bepaalde letters niet door elkaar halen.

Tip: een goede gewoonte is om de tekenreeks aill1 in de verschillende lettertypen te testen, zodat geen enkel teken met een ander kan worden verward.

Om verder te gaan

[Wat is het beste lettertype voor dyslectische bezoekers? - Website Culture Dys \(externe link\)](#)

[A guide to Understanding What Makes a Typeface Accessible – Medium.com \(externe link\)](#)

[Atkinson Hyperlegible Font – brailleinstitute.org \(externe link\)](#)

[Testing fonts for accessibility – UX Collective \(externe link\)](#)

[Luciole lettertype \(externe link\)](#)

[Digital typography and accessibility \(externe link\)](#)

[Practical Typography \(externe link\)](#)

Indien de gebruiker werkt met hulpmiddelen om tekens en regelafstanden te vergroten mogen het kader en het lettertype van de knoppen niet vast zijn en moeten zij zich aanpassen aan de vergroting of verkleining.

Om verder te gaan

[WCAG 2.1 AA – Herschalen van tekst – criterium 1.4.4 \(externe link\)](#)

[WCAG 2.1 AA – Tekstafstand – criterium 1.4.12 \(externe link\)](#)

[Aanbeveling 6: Informatie mag niet alleen door kleur of vorm worden verstrekt](#)

Informatie mag niet alleen door middel van kleur of vorm worden verstrekt, anders is zij wellicht niet waarneembaar voor mensen met een visuele beperking.

Bovendien mag de huidige pagina in de tabbladen niet louter met een andere tekstkleur worden aangeduid, maar moet dit gecombineerd worden met een opmaak naar keuze, kader, onderstreping enz.

Om verder te gaan

[WCAG 2.1 AA – Gebruik van kleur – criterium 1.4.1 \(externe link\)](#)

[Technische hulp over betekenisvolle labels en omschrijvingen in de vorm van pictogrammen](#)

Aanbeveling 7: Pas het responsive design toe

Wanneer een website niet responsief is, is dit te merken aan het gebrek aan aanpassingsvermogen op de drager (mobiele telefoon of tablet): men moet in- of uitzoomen op de pagina omdat deze niet aan het scherm is aangepast, bepaalde knoppen werken niet, vensters, menu's of uitklaplijsten worden niet correct weergegeven of werken niet enz.

D. Structuur en navigatie

[Aanbeveling 8: Zorg voor ten minste de volgende drie navigatiepaden: het navigatiemenu, de sitemap en een zoekmachine](#)

Om de gebruiker toegang te geven tot de informatie die hij zoekt, moeten hem verschillende middelen worden aangeboden, waaronder ten minste het navigatiemenu, een zoekmachine en een sitemap.

Veel sites hebben geen sitemap meer, maar het is een navigatiesysteem dat de gebruiker enorm helpt om de structuur van de site te visualiseren. Deze sitemap moet toegankelijk zijn vanaf dezelfde plaats op elke pagina van de site.

Meerdere navigatiepaden aanbieden om dezelfde informatie of hetzelfde formulier te vinden kan bedoeld zijn om de zoektocht te vergemakkelijken, maar kan ook een overvloed aan informatie genereren en informatie moeilijker toegankelijk maken. De bezoeker kan dan geheel verloren lopen en gedemotiveerd raken om verder te gaan.

Daarom is het belangrijk om de procedures (zoveel mogelijk) te vereenvoudigen en te standaardiseren, en bij het ontwerpen of invoeren van een nieuwe procedure de functionaliteiten te testen met een steekproef van het publiek dat aan de beoordelingen in het kader van deze opdracht heeft deelgenomen ([zie aanbeveling 37](#)).

Om verder te gaan

[Technische hulp over de technische aspecten van de navigatie](#)

[RGAA – Navigatie – Navigatiesystemen – Criterium 12.1 \(externe link\)](#)

Aanbeveling 9: Voorzie een krachtige en spraakgestuurde zoekmachine

De zoekmachine is een instrument dat gebruikers erg op prijs stellen. Sommigen gebruiken haar zelfs als de enige manier om toegang tot informatie te krijgen. Het is dan ook noodzakelijk dat zij efficiënt en performant is.

Daarom moet de zoekmachine spelfouten aanvaarden, rekening houden met synoniemen, met fonetisch schrift, woordsuggesties geven en, indien mogelijk, spraakgestuurd zijn.

Er moet een filter worden toegevoegd om de resultaten van een zoekopdracht te kunnen sorteren.

Wanneer een gebruiker de site doorzoekt via de zoekmachine, moet het toetsenbord naar het eerste zoekresultaat navigeren en niet naar de bovenkant van de pagina verschuiven.

Voorts moet de gebruiker, wanneer het langer duurt om de resultaten te laden en hij er niet onmiddellijk naar kan navigeren, worden gewaarschuwd dat de resultaten aan het laden zijn.

Voor gebruikers die moeite hebben met het geschreven Frans of gewoon om tijd te besparen, kan het inschakelen van autocompletion erg nuttig zijn.

Autocompletion is een functie die de gebruiker woorden voorstelt op basis van de eerste getypte tekens of op basis van de persoonlijke informatie van de gebruiker die reeds in de browser is opgeslagen.

Om verder te gaan

[Technische hulp over autocompletion](#)

Zoekmachines moeten ook rekening houden met de toegankelijkheidscriteria voor formulieren, semantiek en navigatie.

Wij denken hierbij aan :

- [Toegankelijke labels](#)
- [Instructies en andere informatiemeldingen](#)
- [De paginatitel](#)
- [Toetsenbordvriendelijkheid](#)

[Aanbeveling 10: Respecteer de normen voor het structureren van websites](#)

Een goede websitestructuur is essentieel voor mensen die gebruik maken van hulptechnologieën en voor diegenen die enkel met een toetsenbord werken. Zij hebben geen muis en zijn dus afhankelijk van deze structuur om op een logische manier door de site te kunnen navigeren.

Respect voor de structuur betekent respect voor de niveaus van de titels, de manier waarop de lijsten met opsommingstekens zijn ontworpen, het identificeren van de navigatiegebieden van de site zoals menu, hoofdgedeelte, voettekst enz. Die hebben allemaal een specifieke code zodat gebruikers naar believen van gebied naar gebied kunnen gaan, en ongewenste gebieden kunnen vermijden.

Om verder te gaan

[Technische hulp over semantiek en structuur](#)

[Technische hulp over lijsten](#)

[Aanbeveling 11: Voer skip-links in](#)

Voor blinde en slechtziende gebruikers moeten skip-links worden voorzien, zodat de spraaksynthese niet systematisch alle rubrieken herhaalt alvorens de gewenste inhoud te bereiken.

Deze skip-link moet de eerste zijn in de tabvolgorde van elke webpagina en moet op elke pagina van de site staan. Als hij verborgen is, moet hij op zijn minst zichtbaar worden wanneer men erop focust.

Om verder te gaan

[Technische hulp over skip-links](#)

[Aanbeveling 12: Geef informatie weer om duidelijk te maken waar men zich bevindt in de boomstructuur van de site](#)

Om de bezoekers in staat te stellen hun weg in het proces te vinden, is het van belang op alle pagina's en in de in te vullen administratieve formulieren informatie weer te geven waarmee zij hun plaats in de boomstructuur van de site kunnen vinden (een broodkruimelspoor bijvoorbeeld). Deze functie vergemakkelijkt de navigatie en de mentale voorstelling van de structuur van de site en de status van elke administratieve aanvraag.

Om verder te gaan

[Technische hulp over navigatie tussen meerdere pagina's](#)

[Technische hulp over de identificatie van de stappen bij progressieve formulieren](#)

[Aanbeveling 13: De hele site moet toetsenbordvriendelijk zijn](#)

Heel vaak hebben mensen die alleen het toetsenbord gebruiken geen toegang tot bepaalde knoppen of rubrieken.

Om de weg op de pagina te kunnen vinden, moet deze navigeerbaar zijn via alle interactieve elementen die met het toetsenbord te bereiken zijn. Het is ook van essentieel belang dat al deze interactieve elementen kunnen worden geactiveerd via de "Enter"- of "Spatie"-toets.

Bovendien wordt de navigatie voor gebruikers van spraaksynthese of mensen die enkel met een toetsenbord werken verstoord als de tabvolgorde niet consistent is, d.w.z. als de leesrichting niet wordt gerespecteerd (van boven naar beneden en van links naar rechts voor het Frans). Zij moeten dan de hele site doorlopen voordat zij de gewenste rubriek, knop, link enz. bereiken.

Elk pop-upvenster moet met het toetsenbord kunnen worden gesloten door op de sluitknop te drukken of via de Esc-toets.

Tenslotte, wanneer de gebruiker door een van de interactieve componenten navigeert, mag de tab niet geblokkeerd geraken in de component en mag hij niet vast komen te zitten in de commando's van een image slider.

Om verder te gaan

[Technische hulp over het visualiseren en identificeren van hyperlinks](#)

[Technische hulp over toetsenbordvriendelijkheid](#)

[Aanbeveling 14: Gebruik expliciete termen die de actie van een element \(link/knop\) nauwkeurig beschrijven](#)

Sommige knoppen en links zijn niet erg expliciet of begrijpelijk voor gebruikers met een visuele beperking.

Voor ziende mensen kan de lay-out van knoppen en elementen immers extra informatie meedelen, die voor visueel gehandicapten niet waarneembaar is.

Bovendien kunnen mensen die hulptechnologieën gebruiken van het ene element (link of knop) naar het andere navigeren. De bewoordingen van deze elementen moeten daarom begrijpelijk en beschrijvend zijn. Daarom kunnen termen als "Meer weten", "Meer info", "Te lezen" enz. beter niet worden gebruikt.

Om verder te gaan

[Technische hulp over Labels](#)

[Technische hulp over de toegankelijke naam van hyperlinks](#)

E. Redactionele inhoud

[Aanbeveling 15: Beperk de lengte van de teksten en illustreer de inhoud met visuals](#)

Te veel tekstuele inhoud demotiveert de gebruiker, of jaagt hem zelfs weg, omdat hij zich meer moet inspannen en concentreren om de gewenste informatie te vinden, die op de site naar de achtergrond verdwijnt.

Om niet in deze val te trappen is het noodzakelijk de lengte van de teksten zoveel mogelijk te beperken en gebruik te maken van visuals die enerzijds de tekstuele inhoud van de site illustreren, of zelfs toelichten, en anderzijds de tekst verluchten, waardoor de inhoud beter te begrijpen is.

De te gebruiken visuals kunnen pictogrammen, afbeeldingen, schermafdrucken zijn om onlineprocedures uit te leggen enz.

Om verder te gaan

[WCAG – Richtlijn 3.1 begrijpen – Content leesbaar en begrijpelijk maken \(externe link\)](#)

[Informatie voor iedereen – Europese regels voor eenvoudig leesbare en begrijpelijke communicatie \(externe link – PDF 838 KO, FR\)](#)

[Aanbeveling 16: Vertaal de site in het Frans, Nederlands en Engels](#)

Een Engelse vertaling van de online diensten toevoegen zou een groot deel van de anderstaligen die geen Frans of Nederlands spreken of begrijpen, in staat stellen hun administratieve procedures uit te voeren.

[Aanbeveling 17: De gebruiker moet voortdurend in dezelfde taal kunnen blijven navigeren](#)

Op een meertalige site kan elke gebruiker zijn taal van voorkeur kiezen. Soms gebeurt het echter dat pagina's op dezelfde site tijdens het navigeren plotseling van taal veranderen, zelfs zonder dat daarvoor wordt verwittigd. De gebruikers raken dan de weg kwijt, zijn in de war en kunnen hun verzoek niet voortzetten, omdat zij de nieuwe taal niet begrijpen.

Elke taal moet dus homogeen zijn voor de gebruiker. Wanneer de gebruiker wordt doorverwezen naar een andere pagina (aanmelden, betalingsprocedure enz.), moet deze ook in de logische taal van de navigatie staan. Als de site in het Frans is, moet de taal van alle pagina's de Franse versie zijn.

Om verder te gaan

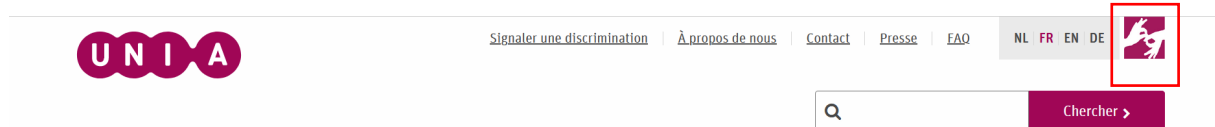
[Technische hulp over de taal van de pagina](#)

[Aanbeveling 18: Zet administratieve acties om in eenvoudige taal en in gebarentaal](#)

Geschreven Frans, en meer bepaald administratieve taal, is niet altijd begrijpelijk voor de gemiddelde burger. Het risico bestaat dan ook dat hij tijdens zijn administratieve procedures onjuiste of onvolledige informatie invult.

Om dit te verhelpen, moeten alle handelingen waarvoor hij informatie moet verstrekken (in te vullen formulier, in te vullen of te uploaden documenten enz.) via videocapsules beschikbaar zijn in eenvoudige taal en in en gebarentaal (VGT). Er moet ook worden uitgelegd welke de stappen zijn van het proces, wat het doel ervan is, hoe men moet antwoorden en aanvragen indienen ...

Dat er videobeelden met gebarentaal voorhanden zijn, moet duidelijk op de site vermeld staan. Het VGT-pictogram met de handen moet op alle pagina's van de site te zien zijn, steeds op dezelfde plaats. (bijvoorbeeld de website van Unia)



Ook hier geldt dat bij het ontwerpen of invoeren van een nieuwe procedure het nodig is de inhoud te testen met een steekproef van het publiek dat aan de beoordelingen in het kader van deze opdracht heeft deelgenomen ([zie aanbeveling 37](#)).

Om verder te gaan

[Informatie voor iedereen – Europese regels voor eenvoudig leesbare en begrijpelijke communicatie \(externe link – PDF 838 KO, FR\)](#)

[Europese regels voor eenvoudig leesbare informatie – Inclusion Europe \(externe link\)](#)

[Frans referentiekader voor het opstellen van alternatieve content met een eenvoudig taalgebruik \(externe link\)](#)

[Facile à Lire et à Comprendre \(FALC\) – FALC.be : een dienst om informatie voor iedereen toegankelijk te maken in het Frans. Vertaling, training en promotie \(externe link\)](#)

[Wablieft vzw: een dienst om informatie voor iedereen toegankelijk te maken in het Nederlands. Vertaling en training \(externe link\)](#)

[LSFB: een vertaaldienst voor videocapsules in Franstalige gebarentaal \(externe link\)](#)

[Visual box : een vertaaldienst voor videocapsules in Nederlandstalige gebarentaal \(externe link\)](#)

[Aanbeveling 19: Voeg een tekstalternatief toe voor afbeeldingen die informatie bevatten](#)

Voor mensen die spraaksynthese gebruiken, moeten alleen afbeeldingen die informatie bevatten welke niet op de pagina aanwezig is, toegankelijk worden gemaakt via een tekstalternatief. Omgekeerd moeten decoratieve beelden er geen hebben.

Om verder te gaan

[Technische hulp over alternatieven voor afbeeldingen](#)

F. Een profiel aanmaken, aanmelden en afmelden

[Aanbeveling 20: Implementeer de aanmeldprocedure via CSAM](#)

Een profiel aanmaken kan vervelend zijn omdat men een lang en moeilijk te onthouden wachtwoord moet kiezen, moet wachten op een bevestigingsmail dat het profiel is aangemaakt, of ook door de procedure om een vergeten wachtwoord terug te krijgen enz.

Om te voorkomen dat voor elke DOD een ander profiel moet worden aangemaakt, is het noodzakelijk de wijze van verbinding met de verschillende overheidsdiensten via digitale CSAM-sleutels te standaardiseren.

Opdat iedereen deze methode zou kunnen gebruiken, moet de uitleg over het gebruik (en het aanmaken van een profiel, onder meer voor Itsme) beschikbaar zijn in eenvoudige taal en, voor wat de videobeelden betreft, in gebarentaal.

Er moeten een e-mailadres en een telefoonnummer, met vertolking op afstand in gebarentaal, bestaan waar iemand in geval van technische problemen of voor vragen in verband met het gebruik van deze identificatiesleutel terecht kan.

[Aanbeveling 21: Geef een bericht dat de status van aanmelden/afmelden bevestigt](#)

Zodra de gebruiker zich heeft aangemeld/afgemeld, is het belangrijk dat de bevestiging of dit al of niet is gelukt duidelijk wordt weergegeven. Deze aanbeveling is van toepassing op alle handelingen zoals de betalingsprocedure, een afspraak maken enz.

Om verder te gaan

[Technische hulp over feedback-identificatie](#)

[Aanbeveling 22: Geef een waarschuwingsbericht als een gebruiker een venster sluit zonder eerst af te melden](#)

Mensen die niet vertrouwd zijn met ICT-tools beschikken niet over bepaalde automatisen, met name die om af te melden alvorens een pagina op een site te sluiten. Dit kan leiden tot veiligheidsproblemen als de persoon is aangemeld op een gedeelde computer (bibliotheek, EPN enz.).

Er moet een waarschuwingsbericht verschijnen wanneer de gebruiker het venster of tabblad probeert te sluiten, welk aangeeft dat deze actie hem onmiddellijk of binnen een bepaalde periode zal doen afmelden. Indien nodig moet de gebruiker worden geïnformeerd als hij op het punt staat de pagina te sluiten, maar nog wel is aangemeld en moet hem worden verteld hoe hij kan afmelden.

Om verder te gaan

[Technische hulp over feedback-identificatie](#)

[Technische hulp over waarschuwingen en meldingen](#)

[Technische hulp over contextwijzigingen](#)

[Aanbeveling 23: Verleng de maximale duur voor de aanmeldprocedure](#)

De gebruikers hebben drie minuten tijd om zich via de Itsme-app aan te melden op een DOD. Deze tijdslimiet is te kort voor verschillende doelgroepen, vooral als er via pictogrammen een extra verificatie moet gebeuren tussen de telefoon en de computer. Daarom moet de limiet worden verlengd, zodat de gebruiker voldoende tijd krijgt om zich aan te melden.

Om verder te gaan

[WCAG 2.1 – Bedienbaar– Richtlijnen voor genoeg tijd \(externe link\)](#)

G. Informatie invoeren en formulieren invullen

[Aanbeveling 24: Waarborg de toegankelijkheid van de formulieren](#)

Formulieren leveren regelmatig problemen op voor gebruikers van toetsenborden en spraaksynthese. Het kan moeilijk zijn om te weten wat zij in de velden moeten invullen, waar ze zich in het formulier bevinden of om de knop te vinden om hun antwoord te verzenden.

Beveiligingsmaatregelen zoals “vraag-antwoord authenticatie” (bijvoorbeeld captcha) zijn ook vaak ontoegankelijk voor een aantal gebruikers.

Alle formulierelden (invoervelden, keuzerondjes, aankruisvakjes enz.) moeten een zichtbaar label hebben dat visueel en technisch verbonden is.

Om verder te gaan

[Technische hulp over toegankelijke formulieren](#)

[CAPTCHA's en toegankelijkheid – Orange.com \(externe link, FR\)](#)

[Aanbeveling 25: Vraag alleen strikt noodzakelijke informatie](#)

Hoe meer informatie moet worden ingevoerd, hoe groter het risico dat de gebruiker een fout maakt of de procedure verlaat.

Daarom is het belangrijk ervoor te zorgen dat op de formulieren alleen strikt noodzakelijke informatie wordt gevraagd, in overeenstemming met [de algemene verordening gegevensbescherming \(AVG\)](#).

[Aanbeveling 26: Maak het mogelijk om persoonlijke gegevens manueel in te voeren en aan te passen](#)

Gebruikers vinden zich niet altijd terug in de keuzemogelijkheden die worden geboden in de vervolgkeuzelijsten van in te vullen formulieren of velden. Het risico bestaat daarom dat zij informatie bij benadering of zelfs onjuist invoeren.

Daarom moeten de gebruikers de toelating krijgen zoveel mogelijk informatie handmatig in te voeren (bijvoorbeeld via een veld “ander”).

Bovendien moet niet alle informatie verplicht worden gemaakt, omdat het kan gebeuren dat de gebruiker niets heeft om in te vullen. Het woord “Verplicht” is ook te verkiezen boven de asterisk (*), die niet iedereen begrijpt.

De gebruiker moet ook zoveel mogelijk van zijn persoonsgegevens kunnen wijzigen: als bepaalde gegevens niet rechtstreeks op de site kunnen worden aangepast, moet de gebruiker gemakkelijk (bijvoorbeeld via een knop) een verzoek kunnen indienen om zijn gegevens te wijzigen.

Om verder te gaan

[Technische hulp over expliciete labels](#)

[Technische hulp over instructies bij invoervelden van formulieren](#)

[Technische hulp over verplichte invoervelden](#)

[Technische hulp over dynamische formulieren](#)

[Hulp over WAI-ARIA design patterns](#)

[Aanbeveling 27: Specificeer het formaat van de gevraagde informatie](#)

Het formaat van de gevraagde informatie (telefoon, adres, geboortedatum enz.) verschilt van site tot site. Gebruikers verliezen soms veel tijd met het invoeren van hun gegevens, omdat zij niet weten in welk formaat dit moet.

Om de gebruikers in staat te stellen de gevraagde informatie correct in te vullen, moet met voorbeelden worden gespecificeerd in welk formaat zij dit dienen te doen.

Voor een formaat van een datum, telefoonnummer of e-mail moeten aan de hand van dummy's concrete voorbeelden worden gegeven: geef voor een e-mailadres het voorbeeld jan.peeters@gmail.com of voor een mobiel telefoonnummer +32 490 11 12 13.

Opgelet: vermijd formaten zoals 04XX/ XX XX XX of email@example.com, die niet altijd begrijpelijk zijn.

Om verder te gaan

[Technische hulp over expliciete labels](#)

[Technische hulp over instructies bij invoervelden van formulieren](#)

[Technische hulp over foutpreventie](#)

[Aanbeveling 28: Geef bij een fout duidelijk aan welke informatie moet worden gewijzigd](#)

Als de gebruiker informatie is vergeten of verkeerd heeft ingevoerd, verschijnt er een foutmelding om hem daarvan op de hoogte te stellen. Deze foutmelding is echter niet altijd duidelijk of zichtbaar.

Foutmeldingen moeten correct worden gekoppeld aan formulervelden met een expliciete boodschap: “Het is verplicht uw voornaam te schrijven” in plaats van “Dit veld is verplicht”. Het verplichte karakter van de velden moet ook worden vermeld voordat ze worden ingevuld.

Bovendien moet de focus van het toetsenbord liggen op het eerste formulerveld dat onjuist is, zodat het direct kan worden gelokaliseerd.

Om verder te gaan

[Technische hulp over expliciete labels](#)

[Technische hulp over instructies bij invoervelden van formulieren](#)

[Technische hulp over foutpreventie en -correctie](#)

[Aanbeveling 29: Geef een bericht dat de status van indiening van het formulier bevestigt](#)

Wanneer een gebruiker een formulier invult, is het niet altijd duidelijk of dit ook werkelijk is verzonden. Het kan dus gebeuren dat de gebruiker het formulier opnieuw invult, in de veronderstelling dat de eerste poging mislukt is.

Daarom moet bovenaan de pagina een bericht verschijnen dat de status van indiening van het formulier duidelijk aangeeft.

Om verder te gaan

[Technische hulp over de bevestiging van het verzenden van formulieren](#)

[Technische hulp over de paginatitel](#)

[Technische hulp over waarschuwingen en meldingen op de pagina](#)

H. Documenten uploaden en downloaden

[Aanbeveling 30: Identificeer de aard en het gewicht van het downloadbare document](#)

Om te voorkomen dat een document de computer van de gebruiker overbelast, moet deze eerst de aard en het gewicht van het document kennen om te kunnen beoordelen of hij het kan downloaden.

Daarom is het noodzakelijk het soort (pdf, Word, PowerPoint enz.) en het gewicht van het document (Kb, Mb, GB, enz.) te vermelden. Indien het document in een andere taal is opgesteld dan die van de pagina, moet ook de taal van het document worden gespecificeerd.

Om verder te gaan

[Technische hulp over externe links en downloadlinks](#)

[Aanbeveling 31: Leg uit hoe men documenten kan scannen en uploaden](#)

Mensen die niet vertrouwd zijn met ICT-tools weten niet hoe zij een document moeten scannen of uploaden.

Dit moet hen dus worden uitgelegd in eenvoudige taal en geïllustreerd met schermafbeeldingen als toelichting bij deze twee handelingen.

I. Betalende procedures uitvoeren

[Aanbeveling 32: Bied betaling aan met debetkaart \(Bancontact\) en kredietkaart \(Visa en Mastercard\)](#)

Alvorens zij hun administratieve vraag indienen moeten gebruikers weten welke procedures betalend zijn. Sommige mensen zijn immers niet in staat om online betalingen te verrichten en zouden hun aanvraag dus niet kunnen afronden.

Voor elke betalende administratieve procedure moet ten minste betaling met Bancontact, Visa en Mastercard worden aangeboden.

Om verder te gaan

[Opquast-richtlijn 47 – De aanvaarde betaalmiddelen en de bijbehorende procedures worden aangegeven \(externe link, FR\)](#)

[Opquast-richtlijn 57 – De website aanvaardt minstens twee betaalmiddelen \(externe link, FR\)](#)

[Technische hulp over alternatieven voor afbeeldingen](#)

[Aanbeveling 33: Geef een bericht dat de status van de onlinebetaling bevestigt](#)

Net als bij het aanmelden ([aanbeveling 21](#)) moet een bericht aangeven of de betaling al dan niet is geslaagd, om te voorkomen dat de gebruiker de handeling onnodig herhaalt.

Bovendien staan verschillende doelgroepen wantrouwend tegenover het veiligheidsniveau van websites en vooral tegenover onlinebetalingen. Zoals aangegeven in [aanbeveling 18](#) moet de betalingsprocedure worden uitgelegd en moet het veiligheidsniveau van de site worden aangegeven, zowel in eenvoudige taal als in een ondertitelde video met gebarentaal.

Om verder te gaan

[Opquast-richtlijn 60 – Na bevestiging van de bestelling krijgt de klant de transactiereferentie te zien \(externe link, FR\)](#)

[Opquast-richtlijn 62 – Elke factuur wordt via e-mail bevestigd \(externe link, FR\)](#)

[Technische hulp over feedback-identificatie](#)

[Technische hulp over de bevestiging van het verzenden van formulieren](#)

[Technische hulp over de paginatitel](#)

[Technische hulp over waarschuwingen en meldingen op de pagina](#)

J. Multimedia (video- en geluidsbestanden)

[Aanbeveling 34: Video- en geluidsbestanden moeten altijd voor iedereen toegankelijk zijn](#)

Voor alle multimedia-inhoud met geluidsinformatie wordt gesynchroniseerde ondertiteling verwacht in de taal van de video.

Ondertitels moeten niet in de video worden ingebed, maar apart worden toegevoegd via een ondertitelbestand (.srt of een ander) zodat gebruikers die dat nodig hebben ze kunnen personaliseren (bijvoorbeeld door de grootte aan te passen).

Bovendien moet er een videospeler worden voorzien waarmee men de ondertitels kan tonen of verbergen.

In de ondertitels moet alle nuttige informatie worden opgenomen, evenals situationele of contextuele informatie, zoals andere geluidselementen dan de stem, die nuttig zijn om de video te begrijpen.

Ondertiteling is een aanvulling op tekstuele transcriptie.

Audiobeschrijving moet ook worden verstrekt als visuele elementen niet op de website of in de video worden beschreven, maar wel informatie verschaffen.

Alle video- en geluidsbestanden moeten in gebarentaal worden vertaald.

Als een video geen tekstuele inhoud illustreert, moet een tekstueel transcript worden verstrekt voor gebruikers die de ondertitels van een video moeilijk kunnen volgen. Dit moet beschikbaar zijn naast de video, hetzij direct onder de video, hetzij via een link naar de video. Alle nuttige audio-informatie moet erin worden opgenomen.

Om verder te gaan

[WCAG 2.1 – Waarneembaar – Toegankelijkheid van op tijd gebaseerde media \(externe link\)](#)

[Richtlijn 8.3 AcceDe Web – Correct gebruik van toegankelijke video's \(externe link, FR\)](#)

[W3C-WAI – Video's ondertitelen \(externe link, EN\)](#)

[W3C-WAI – Audio- en videomedia begrijpelijk maken \(externe link, EN\)](#)

[Hoe maken we multimedadragers voor iedereen toegankelijk? – Emmanuelle Aboaf \(externe link, FR\)](#)

K. Een afspraak maken

[Aanbeveling 35: Afspraken moet men telefonisch en via een onlinekalender kunnen boeken](#)

Zoals hierboven reeds is uiteengezet, kan de gebruiker bij standaardisering van de basisprocedures van DOD's gemakkelijker en sneller leren hoe ze moeten worden gereproduceerd. Dit geldt ook voor het maken van afspraken.

Het moet mogelijk zijn telefonisch en via een online-agenda afspraken te maken, zodat de gebruiker zelf de dag en datum van de afspraak kan kiezen. Met deze functie kunnen doven een afspraakdatum en -tijd kiezen naar gelang van de beschikbaarheid van een tolk (voor vertolking op afstand) of een begeleider.

Overeenkomstig de aanbevelingen 21, 29 en 33 moet aan de gebruiker een bericht worden gezonden dat bevestigt of het al of niet is gelukt, een afspraak te maken. Er moet ook een bevestigingse-mail worden gestuurd, zodat de burger een schriftelijk bewijs van zijn afspraak heeft.

Om verder te gaan

[Technische hulp over feedback-identificatie](#)

[Technische hulp over de bevestiging van het verzenden van formulieren](#)

[Technische hulp over de paginatitel](#)

[Technische hulp over waarschuwingen en meldingen op de pagina](#)

[Technische hulp om WAI-ARIA design patterns in acht te nemen](#)

L. Contact

[Aanbeveling 36: Bied ten minste een toegankelijke e-mail of contactformulier en het telefoonnummer aan \(aangevuld met vertolking op afstand in gebarentaal\)](#)

Zoals bij elke administratieve procedure die men persoonlijk afhandelt, is het mogelijk dat de burger tijdens het online navigeren problemen ondervindt: er kan zich zo een technisch probleem voordoen, een ingewikkeld formulier kan snel onbegrijpelijk worden enz.

Om alle burgers bijstand te kunnen verlenen, moeten ten minste twee contactmiddelen worden aangeboden: e-mail en/of het contactformulier en de telefoon met vertolking op afstand in gebarentaal.

Om verder te gaan

[Opquast-richtlijn 100 – Het volledige adres en telefoonnummer zijn op elke pagina van de website beschikbaar \(externe link, FR\)](#)

[Opquast-richtlijn 102 – De website biedt minstens twee contactmogelijkheden aan \(externe link, FR\)](#)

M. Proces voor de ontwikkeling van DOD's

[Aanbeveling 37: Laat de DOD vanaf de ontwerpfase testen door een diverse groep burgers](#)

Om de DOD echt toegankelijk te maken voor alle burgers, moet ze vanaf de ontwerpfase worden getest door zoveel mogelijk verschillende groepen burgers, waaronder de op bladzijde 3 geschetste groepen.

Zoals wordt onderstreept in de video “*Service design: what is it?*”⁴ beschikbaar in de tools van het playbook van BOSA moet de burger bij het ontwerp van elk project op de eerste plaats komen: hoe eerder een project in handen van een gebruiker komt, hoe sneller er kan worden bijgestuurd als er zich een probleem voordoet. Ook kan vóór de officiële lancering van de DOD een nieuwe testfase worden georganiseerd om ervoor te zorgen dat de aanpassingen zijn doorgevoerd en echt aan de behoeften van alle gebruikers voldoen.

Als deze verschillende doelgroepen de site testen, kan dit de doeltreffendheid van de dienst versterken, de toegankelijkheid ervan verzekeren en zowel functionele als technische valkuilen blootleggen⁵.

[Aanbeveling 38: Maak een gebruiksaanwijzing voor de DOD's](#)

Zodra de structuur en de lay-out van alle DOD's uniform zijn, moet één universele gebruikersgids uitleggen hoe men op de sites kan navigeren en de administratieve procedures online kan afhandelen.

Deze handleiding moet worden aangeboden in videovorm (in eenvoudige taal en gebarentaal en ondertiteld) en in papieren vorm, downloadbaar en afdrukbaar, geschreven in eenvoudige taal. Beide formaten moeten vergezeld gaan van illustraties (schermafbeeldingen, pictogrammen enz.).

[Aanbeveling 39: Geef professionals een opleiding in inclusieve aanbevelingen](#)

Om ervoor te zorgen dat de inclusieve aanbevelingen correct worden toegepast op de DOD's, moeten alle professionals die verantwoordelijk zijn voor deze sites (webontwerpers, grafisch ontwerpers, redacteurs enz.) worden opgeleid in digitale toegankelijkheid.

Deze opleidingen moeten worden georganiseerd voor alle nieuwe medewerkers, om deze continuïteit in de toegankelijkheid van de DOD's te waarborgen.

⁴ Video beschikbaar op de website van het playbook van BOSA. Online: <https://digitalopen.belgium.be/nl/playbook/tools/service-design>

⁵ Faure, L & Brotoorne, P (2021). *Guide pour une conception inclusive des services numériques*. Idéalic.be. p.18.

N. Een testplatform voor elke DOD

[Aanbeveling 40: Stel een onlinetestplatform ter beschikking aan ondersteunings- en opleidingsinstanties waarmee gebruikers tijdens opleidingen onlinediensten kunnen uitproberen](#)

Om te leren werken met DOD's moeten overheidsdiensten een exacte (actuele) kopie van hun website ter beschikking stellen in een "test"-versie.

Deze leersites zullen ondersteunings- en opleidingsdiensten in staat stellen de platforms met gebruikers te testen, zonder de persoonlijke bestanden van de burgers te wijzigen, zonder dat er in het geval van dummy-aanvragen sprake is van daadwerkelijke betaling of onnodige administratieve rompslomp voor de diensten.

Daartoe zou het Gewest een eenvoudige procedure moeten invoeren om het panel op verzoek van de ontwerpers van websites voor overheidsdiensten te raadplegen, via een raamcontract of een aankoopcentrale.

Als bevoegde instanties verifiëren of de inclusieve aanbevelingen daadwerkelijk worden toegepast, wordt het bovendien mogelijk het inclusieve karakter van deze diensten te valideren.

III. Deel II: Technisch referentiekader voor IT-ontwikkelaars

O. Toegankelijke formulieren

Labels

Elk formulierveld moet een label krijgen dat voor alle gebruikers begrijpelijk is, ongeacht of ze de pagina met hun ogen kunnen raadplegen of niet.

a) Semantische labels

Labels moeten technisch aan hun respectievelijke elementen gekoppeld zijn

Elk formulierveld of bedieningselement moet een label of een zogenaamde 'toegankelijke naam' krijgen. Dat label moet technisch aan het overeenkomstige veld worden gekoppeld om zijn betekenis ondubbelzinnig duidelijk te maken.

Er bestaan verschillende manieren om toegankelijke labels te maken. De beste manier is het gebruik van de daarvoor bedoelde HTML-tags. Dat is de meest robuuste oplossing, die door alle browsers en hulptechnologieën wordt ondersteund. Bovendien stelt hun gebruik de gebruiker in staat om de focus op het veld te leggen door op het label te klikken. Zo wordt het klikoppervlak groter, wat een vlottere navigatie mogelijk maakt voor mensen met een motorische of visuele beperking waardoor ze moeilijk op een te klein doel kunnen richten.

Er bestaat een rangorde voor de manieren om een veld van een toegankelijke naam te voorzien:

- **Aria-labelledby:** hulptechnologieën gebruiken de tekst die wordt aangewezen door de ID-waarde van het aria-labelledby-attribuut als toegankelijke naam. Hoewel de tekst van aria-labelledby op het scherm zichtbaar is, brengt een klik op die tekst de focus niet naar het overeenkomstige veld. Gebruik deze methode dus slechts als <label> niet werkt.
- **Aria-label:** de tekst van het aria-label-attribuut wordt als toegankelijke naam gebruikt als er geen aria-labelledby bestaat. De tekst van aria-label is volledig onzichtbaar en is dus enkel nuttig voor gebruikers van een schermlezer. In de meest omstandigheden is deze methode dus niet geschikt.
- **<label>:** dit is de meest aanbevolen methode in alle omstandigheden. De inhoud van het label wordt gebruikt als toegankelijke naam als er geen aria-label of aria-labelledby bestaat.
- **Title-attribuut:** dit attribuut mag geen essentiële informatie bevatten en kan dus niet worden gebruikt als voornaamste methode om een toegankelijke naam te geven. De inhoud van het title-attribuut wordt pas zichtbaar als de muiscursor er overheen zweeft en niet als het veld de focus krijgt. Gebruikers die geen toegang hebben tot de muis (de meeste hulptechnologieën en mensen die enkel het toetsenbord gebruiken) krijgen geen toegang tot die informatie.
- **Placeholder:** de placeholder kan technisch gezien een toegankelijke naam geven, maar deze techniek wordt afgeraden om de volgende redenen. De inhoud verdwijnt zodra de gebruiker in het veld begint te schrijven; de standaardstijl voldoet niet aan de regels voor minimaal contrast; de placeholder mag slechts niet-essentiële inhoud bevatten.

Goede voorbeelden:

Expliciete labels

Aanbevolen methode in alle omstandigheden. De velden worden expliciet aan hun respectievelijke labels gekoppeld.

```
<p>
  <label for="fname_a">First Name:</label>
  <input type="text" name="fname_a" id="fname_a">
</p>
```

```
<p>
  <label for="lname_a">Last Name:</label>
  <input type="text" name="lname_a" id="lname_a">
</p>
```

Impliciete labels

Een impliciete koppeling wordt tot stand gebracht door het veld in het label te plaatsen.

Die methode wordt echter niet door alle browser-en-schermlezercombinaties volledig ondersteund. Er werd een aantal moeilijkheden gemeld voor Firefox + NVDA.

```
<p>
  <label>First Name: <input type="text" name="fname1"></label>
</p>
```

```
<p>
  <label>Last Name: <input type="text" name="lname1"></label>
</p>
```

Aria-labelledby als input text

Het tekstveld bevat de toegankelijke naam door de koppeling van zijn id-verwijzing aan het aria-labelledby-attribuut dat zich in het invoerveld bevindt.

Deze techniek is conform, maar heeft een nadeel: klikken in het tekstveld geeft geen focus aan het invoerveld, in tegenstelling tot de methode met het expliciete veld. Daarom mag deze methode slechts worden gebruikt als het <label> om een of andere reden niet werkt.

```
<p>
  <span id="Nickname">Nickname:</span> <input type="text" aria-labelledby="Nickname">
</p>
```

Aria-label als input text

Dit voorbeeld toont een zoekveld dat gebruik maakt van het aria-label-attribuut om een label voor schermlezers te voorzien als er geen zichtbaar label bestaat. Het zoekveld is zichtbaar gelabeld dankzij de zoekknop die vlakbij staat. Dit is een van de zeldzame gevallen waarin het toelaatbaar is om geen zichtbaar label aan het invoerveld te koppelen.

```
<p><input type="text" aria-label="search"> <input type="submit" value="Search"></p>
```

Title-attribuut als input text

In dit voorbeeld wordt het title-attribuut gebruikt als een toegankelijk label op het tekstveld. De titel is onzichtbaar voor ziende mensen, die de inhoud pas te zien krijgen als de muis er overheen zweeft. Gebruikers die enkel een toetsenbord gebruiken hebben geen toegang tot de informatie, die echter wel zichtbaar is in de aangrenzende knop.

```
<p><input type="text" title="search"> <input type="submit" value="Search"></p>
```

Labels moeten aanwezig zijn in tekstvorm

Een label op zich volstaat niet als het geen duidelijke en relevante tekst bevat die door een hulptechnologie kan worden gelezen. Een leeg label is niet beter dan helemaal geen label. Erger nog, zo'n label kan een bron van fouten vormen door de interpretatie die de hulptechnologie of de gebruiker eraan kunnen geven. Als het label geen tekst bevat, zal de gebruiker bovendien geen enkele informatie ontvangen.

Slecht voorbeeld: Pictogrammen als labels zonder tekstalternatief

Er worden icon fonts of pictogrammen gebruikt om het documentformaat te tonen dat de gebruiker in de selectievakjes kan aanvinken. Zonder tekstalternatief kan de spraaksynthese het type formaat niet begrijpen.

```
<fieldset>
```

```
<legend>Choose a file format</legend>
```

```
<p><input type="radio" id="word" name="format9966"> <label for="word"><span class="far fa-file-word fa-2x"></span></label></p>
```

```
<p><input type="radio" id="excel" name="format9966"> <label for="excel"><span class="far fa-file-excel fa-2x"></span></label></p>
```

```
<p><input type="radio" id="pdf" name="format9966"> <label for="pdf"><span class="far fa-file-pdf fa-2x"></span></label></p>
```

```
<p><input type="radio" id="plain" name="format9966"> <label for="plain"><span class="far fa-file fa-2x"></span></label></p>
```

```
</fieldset>
```

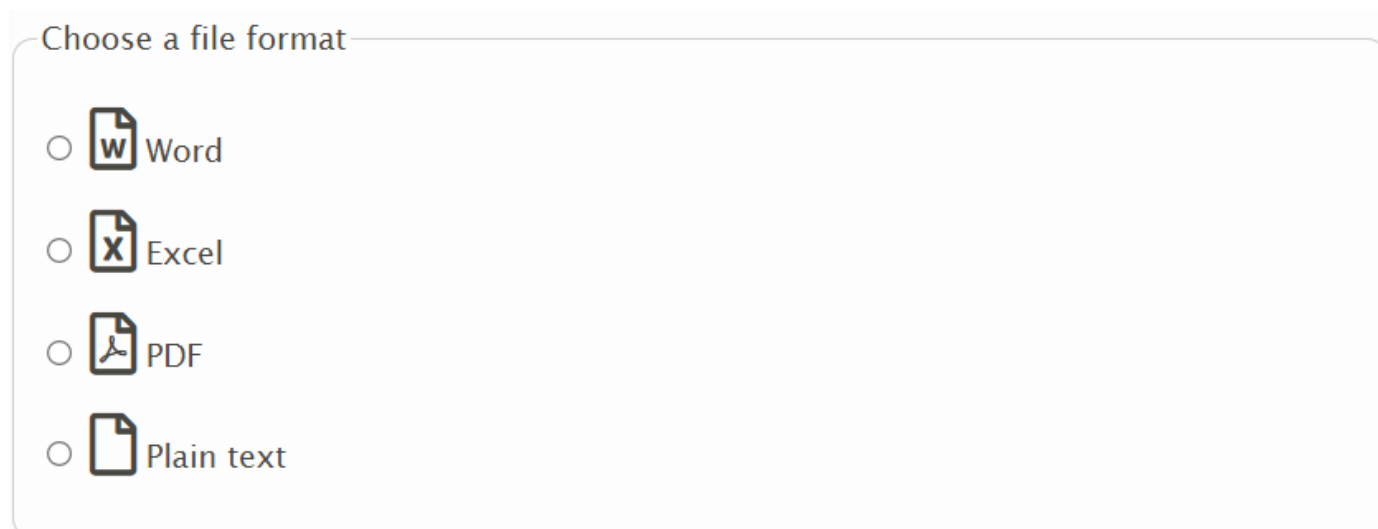
Choose a file format



Goed voorbeeld: Pictogrammen + labels met tekst

Het gebruik van een label met tekst maakt het mogelijk voor de spraaksynthese om de inhoud van elk selectievakje te begrijpen.

```
<fieldset>
<legend>Choose a file format</legend>
<p><input type="radio" id="word1" name="format9965"> <label for="word1"><span class="far fa-file-word fa-2x"></span> Word</label></p>
<p><input type="radio" id="excel1" name="format9965"> <label for="excel1"><span class="far fa-file-excel fa-2x"></span> Excel</label></p>
<p><input type="radio" id="pdf1" name="format9965"> <label for="pdf1"><span class="far fa-file-pdf fa-2x"></span> PDF</label></p>
<p><input type="radio" id="plain1" name="format9965"> <label for="plain1"><span class="far fa-file fa-2x"></span> Plain text</label></p>
</fieldset>
```



Slecht voorbeeld: Een afbeelding zonder tekstalternatief wordt als label gebruikt

In dit voorbeeld lijkt het veld een tekstlabel te bevatten door het woord 'search' dat in de afbeelding staat. Er is trouwens een labelcode aanwezig, maar die bevat geen leesbare tekst omdat de afbeelding geen tekstalternatief bevat.



```
<p>
<label for="fa-search-no-alt">
  
</label>
<input type="text" id="fa-search-no-alt">
```


</p>

[Goed voorbeeld: Een afbeelding met tekstalternatief in het label](#)

<p>

```
<label for="fa-search-alt">
```

```
  
```

```
</label>
```

```
<input type="text" id="fa-search-alt">
```

</p>

Opgelet, hoewel dit een goed voorbeeld is van een label dat een tekstalternatief bevat, is tekst in de vorm van een afbeelding te mijden voor de toegankelijkheid. De kwaliteit van de afbeelding vermindert immers als er wordt ingezoomd en de gebruikers kunnen het lettertype, de grootte en de kleur van de tekst niet aanpassen aan hun specifieke behoeften.

[b\) Betekenisvolle labels](#)

De tekst van de labels moet betekenisvol zijn

Om een formulier te kunnen invullen, hebben de gebruikers duidelijke, informatieve, nauwkeurige en betekenisvolle labels nodig. Internetgebruikers moeten precies weten welke informatie wordt gevraagd om elk veld foutloos te kunnen invullen.

[Goed voorbeeld: een beschrijvend label](#)

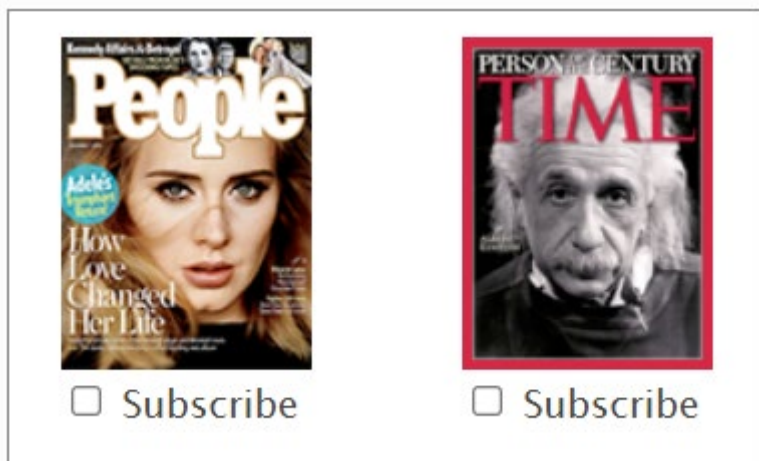
Dit label beschrijft precies de informatie die in het veld wordt verwacht. Naast het woord 'Naam' worden ook voornaam en familienaam vermeld.

Name (First and Last):

```
<label for="name">Name (First and Last):</label> <input type="text" name="name" id="name">
```

[Slecht voorbeeld: een dubbelzinnig label](#)

In dit slechte voorbeeld vermelden de twee selectievakjes slechts de tekst 'Subscribe'. Beide keuzemogelijkheden worden voorafgegaan door een afbeelding die de omslag van een tijdschrift bevat. Geen van beide afbeeldingen bevat een tekstalternatief en ze worden niet aan hun respectievelijke selectievakjes gekoppeld. Daardoor zal een gebruiker van spraaksynthese gewoon twee keer de tekst 'Subscribe' horen, zonder te weten over welk tijdschrift het gaat.



<p style="float:left;margin:0;">

<input type="checkbox" id="people9999"> <label for="people9999">Subscribe</label>

</p>

<p style="float:right;margin:0;">

<input type="checkbox" id="time9999"> <label for="time9999">Subscribe</label>

</p>

Labels mogen zich niet beperken tot zintuiglijke verwijzingen om informatie mee te delen.

Als we enkel uitgaan van zintuiglijke verwijzingen om de informatie in de labels mee te delen, beletten we sommige mensen om te begrijpen wat er wordt gevraagd. Uitgaan van het idee dat alle gebruikers vormen, kleuren of pictogrammen kunnen zien, kan bijvoorbeeld blinden, kleurenblinden of slechtzienden uitsluiten. Hou steeds de volgende punten voor ogen:

- Als er een kleur wordt gebruikt om informatie in een label over te brengen, moet er daarnaast zeker ook een andere manier worden gebruikt, zoals bijvoorbeeld tekst of een ondubbelzinnig symbool.
- Het kleurcontrast van de tekst of het pictogram van het label moet voldoen aan de minimale toegankelijkheidsverhouding.
- Alle informatie die in een label visueel wordt overgebracht, moet via een geschikte techniek begrijpelijk zijn voor hulptechnologieën.

Slecht voorbeeld: Het aria-label is niet informatief

In dit voorbeeld worden een tekstveld en zijn omschrijving gevolgd door een knop met de letter 'i'. Visueel is die knop bedoeld om informatie over het tekstveld te geven. Het aria-label bevat echter slechts de informatie 'button', die onvoldoende duidelijk is voor een gebruiker van spraaksynthese.

Favorite hue: 

<label for="hue">Favorite hue:</label> <input type="text" id="hue">

```
<button aria-label="button" id="hueButtonBad">
  <span class="fa fa-info-circle"></span>
</button>
```

Goed voorbeeld: Het aria-label is informatief

Het aria-label bevat duidelijke, informatieve informatie.

```
<label for="hue2">Favorite hue:</label> <input type="text" id="hue2">
<button aria-label="What does 'hue' mean?" id="hueButtonGood">
  <span class="fa fa-info-circle"></span>
</button>
```

Slecht voorbeeld 2: Het aria-label is niet informatief

Dezelfde logica als hierboven voor een pictogram met een vraagteken, dat de bedoeling zou moeten hebben om te informeren over de inhoud van het selectievakje. Het aria-label bevat de tekst 'Question mark'.

I have never flown in a dirigible 

```
<input type="checkbox" name="dirigible" id="dirigible1">
<label for="dirigible1">I have never flown in a dirigible</label>
<button aria-label="question mark" id="questionButtonBad">
  <span class="fa fa-question-circle"></span>
</button>
```

Goed voorbeeld 2: Het aria-label is informatief

```
<input type="checkbox" name="dirigible" id="dirigible2">
<label for="dirigible2">I have never flown in a dirigible</label>
<button aria-label="What is a dirigible?" id="questionButtonGood">
  <span class="fa fa-question-circle"></span>
</button>
```

c) Omschrijvingen in de vorm van pictogrammen

Pictogrammen mogen als tekstloze omschrijvingen worden gebruikt als hun betekenis vanzelfsprekend is EN als er technisch een tekst aan het label wordt gekoppeld om de informatie voor hulptechnologieën over te brengen.

Goed voorbeeld: De betekenis van het pictogram is duidelijk en er is een tekst aan gekoppeld

Een zoekveld wordt gevolgd door een knop die een vergrootglas afbeeldt. Er is geen aangrenzende visuele tekst die het woord 'zoeken' bevat omdat de betekenis van het vergrootglas duidelijk genoeg is en op websites regelmatig met die bedoeling wordt gebruikt.

Een aria-label bevat het woord 'Search' om de gebruikers van spraaksynthese te informeren.



```
<p class="center">
<input type="text" aria-label="Search">
<button id="search-button" aria-label="Search">
  <span class="fa fa-search"></span>
</button>
</p>
```

Slecht voorbeeld: De betekenis van het pictogram is niet vanzelfsprekend, ook al is er een duidelijke tekst aan gekoppeld voor spraaksynthese.

Een pictogram beeldt een chemische kolf af zonder visuele tekst ernaast. De inhoud van het aria-label vraagt om de chemische formule van vitamine C te tonen, maar die tekst is niet zichtbaar voor mensen die geen spraaksynthese gebruiken.



```
<button id="flask-button" aria-label="Show chemical formula for Vitamin C">
  <span class="fa fa-flask fa-2x"></span>
</button>
```

d) Omschrijvingen in de vorm van een placeholder

Placeholders mogen niet worden gebruikt als enige manier om een tekstveld van een omschrijving te voorzien.

Ontwerpers vervangen de omschrijving vaak door een placeholder om plaats te winnen op de pagina. De placeholder verdwijnt zodra het veld de focus krijgt en de gebruiker tekst begint in te voeren. De gebruiker kan die informatie dan niet meer zien.

Bovendien is de placeholder tekst vaak minder gecontrasteerd en kunnen sommige gebruikers die niet zien. Ontwerpers kunnen dan geneigd zijn om dat contrast te versterken, maar dan kan de internetgebruiker denken dat hij in dat veld al een waarde heeft ingevoerd.

Als de placeholder belangrijke informatie bevat over de manier waarop het veld moet worden ingevuld, verliest de gebruiker die informatie zodra de gegevens worden ingevoerd. Bij twijfel moet hij dan zijn eigen gegevens wissen om de instructies opnieuw te zien.

Spraaksynthese ondersteunt meestal de placeholder, maar geeft die informatie niet altijd mee als er al gegevens werden ingevuld.

Als placeholders worden gebruikt, mogen ze dus niet de enige manier zijn om een veld te labelen.

Als de placeholder naast een label wordt gebruikt, leest de spraaksynthese de verstrekte informatie tweemaal: één keer voor de inhoud van het label en één voor de placeholder.

Om al die redenen raden we het gebruik van placeholders af, hoewel we de toegankelijkheid van een formulier in het kader van een AA-audit niet ongeldig kunnen verklaren als die techniek wordt gecombineerd met een andere labeltechniek.

[Slecht voorbeeld: Een placeholder gebruikt als enige omschrijving](#)

Zoals hierboven wordt uitgelegd, verdwijnt de tekst zodra gegevens in het veld worden ingevoerd en is het standaardcontrast onvoldoende.

,

```
<p>
<input type="text" placeholder="Last Name"> ,
<input type="text" placeholder="First Name">
<input type="text" placeholder="Middle Name">
</p>
```

[Goed voorbeeld: Een 'Canada Dry'-placeholder ontworpen met een echt label dat boven het veld verschijnt als het de focus krijgt en als de muis erover zweeft \(hover\).](#)

Zolang het veld geen gegevens bevat, wordt het label door het CSS in het invoerveld geplaatst. Als het de focus krijgt, als de muis er overheen zweeft en als er informatie wordt ingevoerd, wordt het label boven het veld geplaatst.

e) [Zichtbaarheid van de omschrijving](#)

De omschrijving moet zichtbaar zijn

Zichtbare omschrijvingen komen iedereen ten goede. Zichtbare en expliciete omschrijvingen voorkomen fouten en nutteloos tijdverlies. De internetgebruikers hoeven niet te raden wat er van hen wordt verwacht.

[Goed voorbeeld: Een visueel formulier met instructies voor spraaksynthese](#)

Dit formulier bevat een expliciete omschrijving en toont visueel, door zijn vorm, het aantal lettertekens dat kan worden ingevoerd. Een systeem maakt het mogelijk om niet meer tekens in te voeren dan verwacht.

De aria-label-attributen delen duidelijke instructies mee voor de spraaksynthese.

Social Security Number: - -

<p>

<label for="ssn1">Social Security Number:</label>

<input type="text" size="3" maxlength="3" id="ssn1" aria-label="Social Security Number first three digits"> -

<input type="text" size="2" maxlength="2" aria-label="middle two digits"> -

<input type="text" size="4" maxlength="4" aria-label="last four digits">

</p>

[Slecht voorbeeld: de placeholder wordt gebruikt in combinatie met aria-label zonder enige andere omschrijving](#)

We verwijzen hier naar het [punt over placeholders](#). Enkel gebruikers van spraaksynthese kunnen steeds de informatie in aria-label horen. Bovendien is die informatie redundant met de placeholder zelf.

Account Number

User ID

Password

<p><input type="text" placeholder="Account Number" aria-label="Account Number"></p>

<p><input type="text" placeholder="User ID" aria-label="User ID"></p>

<p><input type="password" placeholder="Password" aria-label="Password"></p>

Interfacecomponenten waarvan het label tekst en tekstafbeeldingen bevat, moeten beschikken over een toegankelijke naam (name) die de visueel afgebeelde tekst bevat.

Als de toegankelijke naam niet dezelfde is als de naam die de gebruiker ziet, kan hij geen spraakfunctie gebruiken om de verschillende bedieningselementen van het formulier te activeren.

Hulptechnologieën bouwen de toegankelijke naam op aan de hand van de verschillende elementen die de component bevat. Meer bepaald de zichtbare tekst, de omschrijvingen en de attributen aria-label en aria-labelledby. Elk element kan bovendien voorrang krijgen op de andere. Hou dus steeds de volgende aspecten voor ogen:

1. Vermijd het gebruik van tekstafbeeldingen, maar als die in een knop worden gebruikt moet een tekstalternatief met dezelfde tekst als de tekstafbeelding worden voorzien.
2. Aria-label en aria-labelledby overschrijven de andere informatie van de component bij de aanmaak van de toegankelijke naam. Dat betekent dat de toevoeging van aria-label of aria-labelledby geen extra informatie verstrekt, maar de bestaande informatie vervangt, zodat het belangrijk is om daar precies weer te geven wat de gebruiker te zien krijgt.

[Slecht voorbeeld: Een zichtbaar label met een andere tekst dan de toegankelijke naam.](#)

De zichtbare omschrijving toont 'Clear Form', terwijl de toegankelijke naam (name) 'reset' vermeldt. De gebruiker die spraakherkenning wil gebruiken, zegt 'Clear Form', maar de component werkt niet.

Clear Form

```
<p><input type="reset" name="reset" value="Clear Form"></p>
```

Goed voorbeeld: De tekst van het label is identiek aan de toegankelijke naam

Select one... ▾

Submit



```
<p><input type="submit" name="submit" value="Submit"></p>
```

f) Nabijheid van de omschrijving bij haar veld of bedieningselement

Het label moet visueel vlak naast het overeenkomstige element staan.

Als het label zich te ver van zijn element bevindt, lopen gebruikers van de zoomfunctie het risico het verband tussen de twee niet te begrijpen.

Als het om een tekstveld gaat, mag het label links van het veld staan, maar de positie vlak boven het veld is nog beter.

Goed voorbeeld:

First Name:

Last Name:

```
<p>
```

```
  <label for="fname_a">First Name:</label>
```

```
  <input type="text" name="fname_a" id="fname_a">
```

```
</p>
```

```
<p>
```

```
  <label for="lname_a">Last Name:</label>
```

```
  <input type="text" name="lname_a" id="lname_a">
```

```
</p>
```

Slecht voorbeeld:

Username:

```
<div>
```

```
  <div class="labelleft">Username:</div>
```

```
  <div class="center"><input type="text" name="username" id="username"></div>
```

```
</div>
```

Het label moet in het DOM vlak naast het overeenkomstige element staan.

De schermlezer leest de pagina in de volgorde waarin de elementen in het DOM verschijnen. Als het label te ver van zijn veld staat, loopt de gebruiker het risico dat hij het verband tussen de twee niet kan leggen.

Goed voorbeeld:

[Zie voorbeeld over visuele nabijheid.](#)

Slecht voorbeeld:

De elementen van het formulier staan in een tabel. Omdat het label en zijn input correct gekoppeld zijn dankzij for + id, is de gebruiker die het document via de tabtoets raadpleegt in staat om het verband tussen de twee te leggen. Bij raadpleging via de tabpijltjes hoort de gebruiker echter slechts: 'first name, middle name, last name, edit, edit, edit'.

First Name	Middle Name	Last Name
<input type="text"/>	<input type="text"/>	<input type="text"/>

```
<table role="presentation">
```

```
<tr>
```

```
<td><label for="firstname">First Name</label></td>
```

```
<td><label for="middlename">Middle Name</label></td>
```

```
<td><label for="lastname">Last Name</label></td>
```

```
</tr>
```

```
<tr>
```

```
<td><input type="text" id="firstname" name="firstn" size="20"></td>
```

```
<td><input type="text" id="middlename" name="middlen" size="20"></td>
```

```
<td><input type="text" id="lastname" name="lastn" size="20"></td>
```

```
</tr>
```

```
</table>
```

g) Meerdere labels voor één veld

Als meerdere labels worden gebruikt voor één veld, dan moet elk label technisch aan zijn overeenkomstige element worden gekoppeld.

Soms is het nodig om meer dan één label aan een formulierveld te koppelen. Het probleem is dat label en veld een één-op-één-relatie hebben, waarin een label slechts aan één veld en een veld aan één label kan worden gekoppeld.

In dat geval kan het attribuut aria-labelledby worden gebruikt om voor een toegankelijke oplossing te zorgen.

Een klassiek voorbeeld is het gebruik van meerdere labels in een formulier met een tabelstructuur, ook al is die HTML-structuur, zoals we hierboven al zeiden, niet ideaal.

Goed voorbeeld van meerdere labels voor één element

In deze tabel moet elk element van het formulier worden gekoppeld aan meer dan één tabelkop en elke tabelkop moet meer dan één formulierveld van een label voorzien.

Zo komt het eerste selectievakje (checkbox) bijvoorbeeld overeen met 'Now' en met 'Food stamps'. De gebruikers van schermlezers moeten beide labels horen als ze met de tabtoets in de cel 'Now / Food stamps' terechtkomen. Tegelijk verwijst de kop 'Now' ook naar alle elementen van die kolom en verwijst de kop 'Food stamps' naar alle elementen van die rij.

De eerste stap bestaat erin de juiste tags te gebruiken om een toegankelijke tabel te maken en daarna de juiste aria-labelledby in de overeenkomstige cellen te plaatsen.

Edit Program Information

Program	Now	Past	Date
Food stamps	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>
TCA	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>
Medical	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>

Om meerdere koppen aan één element te koppelen, moeten we daar de id van elke kop vermelden, telkens gescheiden door een spatie. Dit is bijvoorbeeld de code die moet worden gebruikt om een element uit het formulier aan twee koppen met id="now" en id="foodstamps" te koppelen:

aria-labelledby="now foodstamps"

In nog complexere tabellen is het mogelijk om drie of meer dan drie gekoppelde labels te gebruiken, maar dat is geen goede praktijk voor formulieren.

Dit is de volledige broncode voor de hierboven afgebeelde tabel:

```
<table>
<caption>Edit Program Information</caption>
<tr>
  <th scope="col"><span id="program">Program</span></th>
  <th scope="col"><span id="now">Now</span></th>
  <th scope="col"><span id="past">Past</span></th>
  <th scope="col"><span id="date">Date</span></th>
</tr>
<tr>
  <th scope="row"><span id="foodstamps">Food stamps</span></th>
  <td><input type="checkbox" name="checkbox" aria-labelledby="now foodstamps"></td>
  <td><input type="checkbox" name="checkbox2" aria-labelledby="past foodstamps"></td>
  <td><input type="text" name="textfield" aria-labelledby="date foodstamps"></td>
</tr>
<tr>
  <th scope="row"><span id="tca">TCA</span></th>
```

```

<td><input type="checkbox" name="checkbox3" aria-labelledby="now tca"></td>
<td><input type="checkbox" name="checkbox4" aria-labelledby="past tca"></td>
<td><input type="text" name="textfield2" aria-labelledby="date tca"></td>
</tr>
<tr>
<th scope="row"><span id="medical">Medical</span></th>
<td><input type="checkbox" name="checkbox5" aria-labelledby="now medical"></td>
<td><input type="checkbox" name="checkbox6" aria-labelledby="past medical"></td>
<td><input type="text" name="textfield3" aria-labelledby="date medical"></td>
</tr>
</table>

```

Opgelet!!! Koplabels moeten worden vermeld tussen -tags die de id bevatten. Als de id op de <th> staat, bestaat het risico dat sommige schermlezers het label niet correct kunnen lezen als de gebruiker met de tabtoets op het formulierelement terecht komt.

h) Eén label voor meerdere velden

Een label dat voor meerdere elementen wordt gebruikt, moet technisch aan elk van die elementen worden gekoppeld.

Soms worden formulievelden in meerdere stukken opgesplitst om een samengesteld nummer zoals een bankrekening, een rijksregisternummer, een telefoonnummer ... in te voeren.

Social Security Number: - -

Phone: () -

Zip: -

Er bestaan verschillende technieken om in dergelijke situaties toegankelijke labels op te stellen:

1. De velden in één veld combineren om een overvloed aan labels te vermijden;
2. De labels in CSS verbergen terwijl ze toch leesbaar blijven voor schermlezers. Die techniek maakt gebruik van <label> en is erg robuust, maar minder elegant dan de techniek met aria-labels;
3. Aria-label gebruiken om verborgen labels te leveren en tegelijk één label te behouden dat het voor de gebruiker mogelijk maakt om de focus in het veld te plaatsen door op het label te klikken;
4. Een <fieldset> gebruiken voor het zichtbare label en verborgen tekst voor de andere labels.

[Goed voorbeeld: Meerdere velden gecombineerd in één veld met één label](#)

Social Security Number:

```
<label for="ssn6">Social Security Number:</label> <input type="text" id="ssn6">
```

[Goed voorbeeld: Meerdere velden met één label en verborgen labels in CSS](#)

Social Security Number : - -

```
<label for="ssn1b">Social Security Number <span class="offscreen">first three digits</span>:
</label>
```

```
<input type="text" size="3" maxlength="3" id="ssn1b"> -
```

```
<label for="ssn2b" class="offscreen">Second two digits:</label>
```

```
<input type="text" size="2" maxlength="2" id="ssn2b"> -
```

```
<label for="ssn3b" class="offscreen">Last four digits:</label>
```

```
<input type="text" size="4" maxlength="4" id="ssn3b">
```

De class 'offscreen' plaatst de inhoud bijvoorbeeld op -9999px top en left.

Enkele opmerkingen over deze methode:

- Deze methode is beter achterwaarts compatibel met oudere browsers en schermlezers dan de methode met aria-labels (zie het volgende voorbeeld);
- Met deze methode herhaalt de schermlezer niet tweemaal dezelfde inhoud en wordt het voorlezen dus comfortabeler en vloeiender.

[Goed voorbeeld: Meerdere velden met één zichtbaar label en aria-labels om omschrijvingen te verbergen.](#)

Social Security Number: - -

Phone: () -

Zip: -

```
<label for="ssn1">Social Security Number:</label>
```

```
<input type="text" size="3" maxlength="3" id="ssn1" aria-label="Social Security Number first three digits"> -
```

```
<input type="text" size="2" maxlength="2" aria-label="middle two digits"> -
```

```
<input type="text" size="4" maxlength="4" aria-label="last four digits">
```

```
</p>
```

```
<p>
```

```
<label for="phone1">Phone:</label>
```

```
(<input type="text" size="3" maxlength="3" id="phone1" aria-label="Phone Number area code">)
```

```
<input type="text" size="3" maxlength="3" aria-label="second three digits"> -
```

```
<input type="text" size="4" maxlength="4" aria-label="last four digits">
```

```
</p>
```

```
<p>
```

```
<label for="zip">Zip:</label>
```

```
<input type="text" size="5" maxlength="5" id="zip" aria-label="Zip code"> -
```

```
<input type="text" size="4" maxlength="4" aria-label="extended zip code">
```

```
</p>
```

Enkele aandachtspunten in verband met deze code:

- Het eerste veld is aan een zichtbaar <label> gekoppeld via de combinatie van id en for. Die techniek maakt het voor alle gebruikers mogelijk om de focus in het veld te plaatsen door op het label te klikken (motorische beperkingen, bevingen, slechtziendheid ...).
- Het eerste veld bevat naast het <label> een aria-label-attribuut om aan gebruikers van een schermlezer te melden dat het veld de drie eerste cijfers van het nummer 'Social Security Number' verwacht. Zonder deze techniek zal de internetgebruiker alle cijfers invoeren, wat een fout oplevert (maxlength="3").
- Op het moment van schrijven lezen JAWS en NVDA enkel de waarde van aria-label, terwijl VoiceOver zowel de waarden van aria-label als van het label leest (wat niet ideaal is).

[Goed voorbeeld: Meerdere velden met één zichtbaar label en gebruik van <fieldset>](#)



```
<fieldset>
```

```
<legend>Social Security Number</legend>
```

```
<label for="ssn10" class="offscreen">First three digits</label>
```

```
<input type="text" size="3" maxlength="3" id="ssn10"> -
```

```
<label for="ssn20" class="offscreen">Second two digits:</label>
```

```
<input type="text" size="2" maxlength="2" id="ssn20"> -
```

```
<label for="ssn30" class="offscreen">Last four digits:</label>
```

```
<input type="text" size="4" maxlength="4" id="ssn30">
```

```
</fieldset>
```

De class 'offscreen' plaatst de inhoud bijvoorbeeld op -9999px top en left.

Opmerkingen bij deze methode:

- Kan ook worden gebruikt met aria-label, zoals eerder werd besproken;
- Deze oplossing werkt heel goed met alle schermlezers en is achterwaarts compatibel;
- Een nadeel is dat het moeilijker kan zijn om de stijl van <fieldset> en <legend> te bepalen ten opzichte van normale tekst. Recente browsers maken het mogelijk om de border van de <fieldset> uit te schakelen en <legend> links van het veld te plaatsen, ook al vergt dat een beetje meer werk van de ontwerper. Oudere browsers zijn op dit punt minder compatibel;
- Een ander nadeel: sommige schermlezers (JAWS) herhalen de tekst voor <legend> in alle velden die de tag bevat. In bovenstaand geval hoort de gebruiker 'Social Security Number' in elk van de drie velden. Dat werkt echter niet blokkerend, behalve als de tekst echt lang is en dat ongemak kan gemakkelijk worden vermeden door meteen de tabtoets in te drukken na de belangrijke informatie gehoord te hebben of door de ctrl-toets te gebruiken om het voorlezen te stoppen.

Labels voor groepen formulervelden

De vereisten voor formulervelden gelden ook voor de veldgroepen van een formulier om de gebruikers in staat te stellen om die groepen vlot te identificeren en geen fouten te maken. Dat is bijvoorbeeld het geval voor een groep keuzerondjes.

De concepten die we besproken voor de labels gelden dus ook voor veldgroepen (semantiek, expliciete tekst, nabijheid en zichtbaarheid).

a) Groepslabelsemantiek

Labels moeten technisch aan hun groep worden gekoppeld als ze op zichzelf niet volstaan.

Als een veldgroep hetzelfde label deelt moeten de groep en zijn label semantisch expliciet zijn, zodat de gebruiker van een schermlezer het verband tussen de velden van het formulier, hun gekoppelde label en de groep waartoe zij behoren kan begrijpen.

De groepering van de componenten is erg belangrijk in het geval van keuzerondjes en selectievakjes (radio buttons en checkboxes). Zelfs als het veld correct aan zijn label is gekoppeld, kan de gebruiker bijvoorbeeld niet het verband leggen tussen een 'ja' of een 'nee' als hij niet weet op welke vraag dat antwoord betrekking heeft.

Dezelfde logica geldt voor velden die deel uitmaken van dezelfde familie, zoals in het geval van verzend- en factuuradressen.

Formulervelden kunnen op twee manieren worden gegroepeerd:

- De basisoplossing in HTML met gebruik van de tag <fieldset> om de elementen te groeperen en de tag <legend> om die semantisch aan een label te koppelen;
- De ARIA-oplossing waarbij de velden worden gekoppeld in een regio met een role="group" en de naam door een aria-labelledby attribuut wordt gegeven.

Goed voorbeeld: Elementen onderling gekoppeld via <fieldset> en <legend>

Contact Information

Name:

Phone:

Email:

Address:

City:

State:

Zip:

<form>

<fieldset>

<legend>Contact Information</legend>

<p><label for="name6044">Name: </label> <input type="text" id="name6044"></p>

<p><label for="phone6044">Phone: </label> <input type="text" id="phone6044"></p>

<p><label for="email6044">Email: </label> <input type="text" id="email6044"></p>

<p><label for="address6044">Address: </label> <input type="text" id="address6044"></p>

<p><label for="city6044">City: </label> <input type="text" id="city6044"></p>

<p><label for="state6044">State: </label> <input type="text" id="state6044"></p>

<p><label for="zip6044">Zip: </label> <input type="text" id="zip6044"></p>

</fieldset>

</form>

De tag <fieldset> groepeert de elementen van dezelfde familie visueel met elkaar dankzij een kader en ook semantisch, waardoor schermlezers de groepering voor de gebruikers kunnen benoemen.

Sommige schermlezers, zoals JAWS en VoiceOver, lezen de tekst binnen <legend> naast de naam van elk veld van de betreffende groep. De meeste lezen eerst de <legend> terwijl VoiceOver eerst het veldlabel leest, gevolgd door het bijschrift van de fieldset.

Goed voorbeeld: Elementen onderling gekoppeld via ARIA

De velden van dezelfde groepering worden in een container geplaatst, zoals bijvoorbeeld een <div>, met een ARIA-rol 'group' en een aria-labelledby-attribuut gekoppeld aan de id van een element dat de tekst van het bijschrift bevat.

Opmerkingen:

- Deze techniek tekent geen kader rond de container. Een visuele groepering van de elementen van dezelfde familie wordt echter wel aanbevolen. We raden aan om die in CSS toe te voegen;
- Op het moment van schrijven wordt role="group" niet door alle versies van VoiceOver ondersteund. Het eerste voorbeeld wordt dus aanbevolen.

Contact Information

Name:

Phone:

Email:

Address:

City:

State:

Zip:

```
<p id="grouplabel9961">Contact Information</p>
```

```
<div role="group" aria-labelledby="grouplabel9961">
```

```
  <p><label for="name">Name: </label> <input type="text" id="name"></p>
```

```
  <p><label for="phone">Phone: </label> <input type="text" id="phone"></p>
```

```
  <p><label for="email">Email: </label> <input type="text" id="email"></p>
```

```
  <p><label for="address">Address: </label> <input type="text" id="address"></p>
```

```
  <p><label for="city">City: </label> <input type="text" id="city"></p>
```

```
  <p><label for="state">State: </label> <input type="text" id="state"></p>
```

```
  <p><label for="zip">Zip: </label> <input type="text" id="zip"></p>
```

```
</div>
```

[Goed voorbeeld: Groepering van formulierelementen van dezelfde familie via geneste <fieldset>-tags](#)

Geneste <fieldset>-tags zijn toegelaten om groepen in een groep te maken.

Opmerking: Oude versies van browsers proberen voor elk formulierelement eerst het bijschrift van de omvattende <fieldset> te lezen en daarna dat van de interne <fieldset>, zodat het voorlezen redundant en minder comfortabel wordt. Meer recente browsers hebben hun werking verbeterd door alleen het bijschrift van de interne <fieldset> te lezen.

Contact Information

Name:

Phone:

Email:

Address:

City:

State:

Zip:

Preferred mode of communication:

Phone

Email

<fieldset>

<legend>Contact Information</legend>

<p><label for="name6045">Name: </label><input type="text" id="name6045"></p>

<p><label for="phone6045">Phone: </label><input type="text" id="phone6045"></p>

<p><label for="email6045">Email: </label><input type="text" id="email6045"></p>

<p><label for="address6045">Address: </label><input type="text" id="address6045"></p>

<p><label for="city6045">City: </label><input type="text" id="city6045"></p>

<p><label for="state6045">State: </label><input type="text" id="state6045"></p>

<p><label for="zip6045">Zip: </label><input type="text" id="zip6045"></p>

<div class="mode-radio">

<fieldset>

<legend>Preferred mode of communication:</legend>

<p><input type="radio" name="mode" id="mode-phone6045">

<label for="mode-phone6045">Phone</label>

<input type="radio" name="mode" id="mode-email6045">

<label for="mode-email6045">Email</label></p>

</fieldset>

</div>

</fieldset>

Slecht voorbeeld: Valse groepering

De elementen van dit formulier worden visueel gegroepeerd in CSS zonder semantisch gekoppeld te zijn. De gebruiker van een schermlezer begrijpt dus het verband tussen die elementen niet.

What is your preferred method of contact?

Email

Phone

Mail

HTML-code:

```
<div class="group1">
  <span><strong>What is your preferred method of contact?</strong></span>
  <div>
    <input type="radio" name="contact" value="email" id="email9017">
      <label for="email9017">Email</label><br>
    <input type="radio" name="contact" value="phone" id="phone9017">
      <label for="phone9017">Phone</label><br>
    <input type="radio" name="contact" value="mail" id="mail9017">
      <label for="mail9017">Mail</label>
  </div>
</div>
```

CSS-code:

```
.group1 {
border: 1px solid grey;
border-radius: 10px;
padding: 10px;
}
```

De labels van de groeperingen moeten technisch bepaalbaar zijn

De WCAG- en HTML-specificaties vereisen niet dat er systematisch aan elke <fieldset> een <legend>-tag wordt toegevoegd. Als een <fieldset> elementen groepeert, moet hij echter een bijschrift krijgen dat door schermlezers kan worden gebruikt.

De twee technieken die het mogelijk maken om een bijschrift aan een groepering te koppelen zijn de <legend>-tag of het aria-labelledby-attribuut bij gebruik van ARIA.

Slecht voorbeeld: Groepering zonder label

Mailing Address

Address:

City:

State:

Zip:

Billing Address

Address:

City:

State:

Zip:

Hoewel de groepering gebeurt via een <fieldset> met <legend>-bijschrift, bevat de afbeelding die in het bijschrift wordt gebruikt geen tekstalternatief dat de gebruikte afbeelding beschrijft. Bovendien lopen ook slechtziende personen die op hun scherm moeten inzoomen, het lettertype moeten wijzigen of de kleur en het contrast van de letters moeten veranderen het risico om door deze methode te worden gehandicapt.

```
<div>
```

```
  <fieldset>
```

```
    <legend></legend>
```

```
    <p><label for="mailaddress9014">Address: </label>
```

```
      <input type="text" id="mailaddress9014"></p>
```

```
    <p><label for="mailcity9014">City: </label>
```

```
      <input type="text" id="mailcity9014"></p>
```

```
    <p><label for="mailstate9014">State: </label>
```

```
      <input type="text" id="mailstate9014"></p>
```

```
    <p><label for="mailzip9014">Zip: </label>
```

```
      <input type="text" id="mailzip9014"></p>
```

```
  </fieldset>
```

```
</div>
```

```
<div>
```

```
  <fieldset>
```

```

<legend></legend>
<p><label for="billaddress9014">Address: </label>
  <input type="text" id="billaddress9014"></p>
<p><label for="billcity9014">City: </label>
  <input type="text" id="billcity9014"></p>
<p><label for="billstate9014">State: </label>
  <input type="text" id="billstate9014"></p>
<p><label for="billzip9014">Zip: </label>
  <input type="text" id="billzip9014"></p>
</fieldset>
</div>

```

b) Expliciete groepslabels

Groepslabels moeten expliciet zijn

De gebruikers moeten precies weten hoe ze de gegevens in het formulier moeten invullen.

Goed voorbeeld: Expliciete tekst voor het bijschrift

De onderstaande velden zijn onderverdeeld in vier <fieldset>-elementen, gekoppeld aan een <legend> die het nummer van de stap en de gevraagde informatie bevat.

Step 1: Pick Your Car

Porsche 911 Ford Mustang Nissan 370Z

Step 2: Pick Your Exterior Color

Red Silver Black

Step 3: Pick Your Interior Color

Light Gray Camel Charcoal

Step 4: Pick Your Delivery Timeframe

2-4 weeks 1-2 months 3-6 months

```

<fieldset>
  <legend>Step 1: Pick Your Car</legend>
  <input type="radio" id="porsche9996" name="car"> <label for="porsche9996">Porsche
  911</label>
  <input type="radio" id="mustang9996" name="car"> <label for="mustang9996">Ford
  Mustang</label>
  <input type="radio" id="nissan9996" name="car"> <label for="nissan9996">Nissan
  370Z</label>

```

```
</fieldset>
```

```
<fieldset>
```

```
  <legend>Step 2: Pick Your Exterior Color</legend>
```

```
  <input type="radio" id="red9996" name="ecolor"> <label for="red9996">Red</label>
```

```
  <input type="radio" id="silver9996" name="ecolor"> <label for="silver9996">Silver</label>
```

```
  <input type="radio" id="black9996" name="ecolor"> <label for="black9996">Black</label>
```

```
</fieldset>
```

```
<fieldset>
```

```
  <legend>Step 3: Pick Your Interior Color</legend>
```

```
  <input type="radio" id="gray9996" name="icolor"> <label for="gray9996">Light  
  Gray</label>
```

```
  <input type="radio" id="camel9996" name="icolor"> <label for="camel9996">Camel</label>
```

```
  <input type="radio" id="char9996" name="icolor"> <label for="char9996">Charcoal</label>
```

```
</fieldset>
```

```
<fieldset>
```

```
  <legend>Step 4: Pick Your Delivery Timeframe</legend>
```

```
  <input type="radio" id="short9996" name="delivery"> <label for="short9996">2-4  
  weeks</label>
```

```
  <input type="radio" id="medium9996" name="delivery"> <label for="medium9996">1-2  
  months</label>
```

```
  <input type="radio" id="long9996" name="delivery"> <label for="long9996">3-6  
  months</label>
```

```
</fieldset>
```

Slecht voorbeeld: Bijschrift met dubbelzinnige tekst

In hetzelfde voorbeeld als hierboven wordt enkel het nummer van de stap meegedeeld. De gebruiker weet niet welke informatie wordt gevraagd.

Step 1

Porsche 911 Ford Mustang Nissan 370Z

Step 2

Red Silver Black

Step 3

Light Gray Camel Charcoal

Step 4

2-4 weeks 1-2 months 3-6 months

<fieldset>

<legend>Step 1</legend>

<input type="radio" id="porsche9997" name="car2">

<label for="porsche9997">Porsche 911</label>

<input type="radio" id="mustang9997" name="car2">

<label for="mustang9997">Ford Mustang</label>

<input type="radio" id="nissan9997" name="car2">

<label for="nissan9997">Nissan 370Z</label>

</fieldset>

<fieldset>

<legend>Step 2</legend>

<input type="radio" id="red9997" name="ecolor2">

<label for="red9997">Red</label>

<input type="radio" id="silver9997" name="ecolor2">

<label for="silver9997">Silver</label>

<input type="radio" id="black9997" name="ecolor2">

<label for="black9997">Black</label>

</fieldset>

<fieldset>

<legend>Step 3</legend>

<input type="radio" id="gray9997" name="icolor2">

<label for="gray9997">Light Gray</label>

```

<input type="radio" id="camel9997" name="icolor2">
  <label for="camel9997">Camel</label>
<input type="radio" id="char9997" name="icolor2">
  <label for="char9997">Charcoal</label>
</fieldset>
<fieldset>
  <legend>Step 4</legend>
  <input type="radio" id="short9997" name="delivery2">
    <label for="short9997">2-4 weeks</label>
  <input type="radio" id="medium9997" name="delivery2">
    <label for="medium9997">1-2 months</label>
  <input type="radio" id="long9997" name="delivery2">
    <label for="long9997">3-6 months</label>
</fieldset>

```

De labels van groeperingen mogen de informatie niet uitsluitend via kleur, oriëntatie, geluid, positie of vorm aanbieden.

Door informatie enkel via een zintuiglijke eigenschap mee te delen, ontstaat het risico dat bepaalde personen worden uitgesloten omwille van hun specifieke behoeften. Het is nodig om een aantal punten voor ogen te houden:

- Als een label kleur gebruikt om informatie te verstrekken, moet de kleur worden versterkt door een andere factor, zoals tekst of een herkenbaar symbool, en op een toegankelijke manier worden geïmplementeerd;
- Het contrast van tekst, symbolen en pictogrammen moet aan de toegankelijkheidsverhoudingen voldoen;
- Elke zichtbare informatie moet technisch bruikbaar zijn voor gebruikers van spraaksynthese.

[Slecht voorbeeld: Omschrijvingen die enkel via zintuiglijke kenmerken informatie verstrekken](#)

In dit voorbeeld gaat de ontwikkelaar ervan uit dat de gebruiker kleuren kan onderscheiden om zijn menu te kiezen. Kleurenblinden en gebruikers van spraaksynthese zullen dit formulier niet op een gepaste manier kunnen beantwoorden.

To make your meal, choose one appetizer (in green), one main course (in red), and one dessert (in blue) from the options below.

- Shrimp cocktail
- Loaded potato skins
- Flat bread pizza
- Buffalo chicken wings
- Seared tuna with mixed greens
- Pasta primevera
- Cheese plate
- Chocolate torte
- Fresh fruit and ice cream

```
<fieldset>
```

```
  <legend>To make your meal, choose one appetizer (in green), one main course (in red),
  and one dessert (in blue) from the options below.</legend>
```

```
  <input type="checkbox" name="meal" value="shrimp" id="shrimp"> <label for="shrimp"
  style="color: green">Shrimp cocktail</label><br>
```

```
  <input type="checkbox" name="meal" value="potato" id="potato"> <label for="potato"
  style="color: green">Loaded potato skins</label><br>
```

```
  <input type="checkbox" name="meal" value="pizza" id="pizza"> <label for="pizza"
  style="color: green">Flat bread pizza</label><br>
```

```
  <input type="checkbox" name="meal" value="wings" id="wings"> <label for="wings"
  style="color: red">Buffalo chicken wings</label><br>
```

```
  <input type="checkbox" name="meal" value="tuna" id="tuna"> <label for="tuna"
  style="color: red">Seared tuna with mixed greens</label><br>
```

```
  <input type="checkbox" name="meal" value="pasta" id="pasta"> <label for="pasta"
  style="color: red">Pasta primevera</label><br>
```

```
  <input type="checkbox" name="meal" value="cheese" id="cheese"> <label for="cheese"
  style="color: blue">Cheese plate</label><br>
```

```
  <input type="checkbox" name="meal" value="torte" id="torte"> <label for="torte"
  style="color: blue">Chocolate torte</label><br>
```

```
  <input type="checkbox" name="meal" value="fruit" id="fruit"> <label for="fruit" style="color:
  blue">Fresh fruit and ice cream</label><br>
```

```
</fieldset>
```

[Goed voorbeeld: Omschrijvingen die informatie verstrekken via een kleur en via een andere techniek](#)

In dit voorbeeld worden de verschillende soorten gerechten gescheiden door <fieldset>-tags die gekoppeld zijn aan hun expliciete <legend>.

To make your meal, choose one appetizer, one main course, and one dessert from the options below.

Appetizers

- Shrimp cocktail
- Loaded potato skins
- Flat bread pizza

Main Courses

- Buffalo chicken wings
- Seared tuna with mixed greens
- Pasta primavera

Desserts

- Cheese plate
- Chocolate torte
- Fresh fruit and ice cream

<fieldset>

<legend>To make your meal, choose one appetizer, one main course, and one dessert from the options below.</legend>

<fieldset>

<legend>Appetizers</legend>

<input type="checkbox" name="meal" value="shrimp" id="shrimp1"> <label for="shrimp1" style="color: green">Shrimp cocktail</label>

<input type="checkbox" name="meal" value="potato" id="potato1"> <label for="potato1" style="color: green">Loaded potato skins</label>

<input type="checkbox" name="meal" value="pizza" id="pizza1"> <label for="pizza1" style="color: green">Flat bread pizza</label>

</fieldset>

<fieldset>

<legend>Main Courses</legend>

<input type="checkbox" name="meal" value="wings" id="wings1"> <label for="wings1" style="color: red">Buffalo chicken wings</label>

<input type="checkbox" name="meal" value="tuna" id="tuna1"> <label for="tuna1" style="color: red">Seared tuna with mixed greens</label>


```
<input type="checkbox" name="meal" value="pasta" id="pasta1"> <label for="pasta1" style="color: red">Pasta primevera</label>
```

```
</fieldset>
```

```
<fieldset>
```

```
<legend>Desserts</legend>
```

```
<input type="checkbox" name="meal" value="cheese" id="cheese1"> <label for="cheese1" style="color: blue">Cheese plate</label><br>
```

```
<input type="checkbox" name="meal" value="torte" id="torte1"> <label for="torte1" style="color: blue">Chocolate torte</label><br>
```

```
<input type="checkbox" name="meal" value="fruit" id="fruit1"> <label for="fruit1" style="color: blue">Fresh fruit and ice cream</label>
```

```
</fieldset>
```

```
</fieldset>
```

c) Nabijheid van groepslabels

Groepslabels moeten visueel in de nabijheid staan van de elementen waarop ze betrekking hebben

De gebruikers van een zoomfunctie moeten visueel het verband kunnen leggen tussen de omschrijving en de gekoppelde elementen.

Slecht voorbeeld van een groepslabel ver van de betreffende velden

Een grote afbeelding verstoort visueel de band tussen het bijschrift en de selectievakjes.

Choose your top three favorite holidays:



- Memorial Day
- 4th of July
- Labor Day
- Halloween
- Thanksgiving

```
</fieldset>
```

<legend>Choose your top three favorite holidays:</legend>

<p></p>

<input type="checkbox" name="holiday" id="memorial9142" value="Memorial"> <label for="memorial9142">Memorial Day</label>

<input type="checkbox" name="holiday" id="july9142" value="July"> <label for="july9142">4th of July</label>

<input type="checkbox" name="holiday" id="labor9142" value="Labor"> <label for="labor9142">Labor Day</label>

<input type="checkbox" name="holiday" id="halloween9142" value=""> <label for="halloween9142">Halloween</label>

<input type="checkbox" name="holiday" id="thanks9142" value=""> <label for="thanks9142">Thanksgiving</label>

</fieldset>

[Goed voorbeeld: het label staat dicht bij de formulervelden](#)



Choose your top three favorite holidays:

- Memorial Day
- 4th of July
- Labor Day
- Halloween
- Thanksgiving

<p></p>

<fieldset>

<legend>Choose your top three favorite holidays:</legend>

<input type="checkbox" name="holiday" id="memorial9134" value="Memorial"> <label for="memorial9134">Memorial Day</label>


```

 <label
for="july9134">4th of July</label><br>
 <label
for="labor9134">Labor Day</label><br>
 <label
for="halloween9134">Halloween</label><br>
 <label
for="thanks9134">Thanksgiving</label><br>
</fieldset>

```

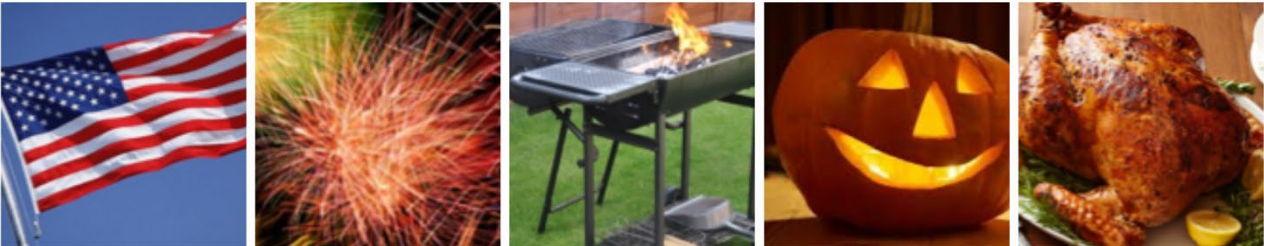
Groepslabels moeten in de nabijheid staan van de elementen waarop ze betrekking hebben in het DOM

Schermlezers lezen de inhoud van een pagina in de volgorde waarin de elementen in het DOM verschijnen. Om de gebruiker niet te misleiden, moeten de groepslabels dicht bij hun velden staan.

Slecht voorbeeld: het groepslabel staat niet dicht bij de elementen in het DOM

Dit formulier werd correct gestructureerd met een <fieldset> en een bijbehorende <legend>. Via de tabtoets kan de gebruiker van een schermlezer van een goede surfervaring genieten. Als de gebruiker echter de navigatiepijlen gebruikt, moet hij de vijf afbeeldingen doorlopen voordat hij de selectievakjes kan lezen.

Choose your top three favorite holidays:



- Memorial Day
- 4th of July
- Labor Day
- Halloween
- Thanksgiving

```

<fieldset>
<legend>Choose your top three favorite holidays:</legend>
<p>




```

```
</p>
  <input type="checkbox" name="holiday" id="memorial9129" value="Memorial">
    <label for="memorial9129">Memorial Day</label><br>
  <input type="checkbox" name="holiday" id="july9129" value="July">
    <label for="july9129">4th of July</label><br>
  <input type="checkbox" name="holiday" id="labor9129" value="Labor">
    <label for="labor9129">Labor Day</label><br>
  <input type="checkbox" name="holiday" id="halloween9129" value="Halloween">
    <label for="halloween9129">Halloween</label><br>
  <input type="checkbox" name="holiday" id="thanks9129" value="Thanksgiving">
    <label for="thanks9129">Thanksgiving</label><br>
</fieldset>
```


d) Zichtbaarheid van groepslabels

Groepslabels moeten zichtbaar zijn.

De labels moeten steeds zichtbaar zijn zodat alle gebruikers kunnen weten welke informatie in de formulervelden wordt gevraagd.

Slecht voorbeeld: de tekst van een bijschrift is verborgen

Het formulier maakt correct gebruik van een <legend> om informatie te verstrekken aan de gebruikers van schermlezers, maar die <legend> wordt in CSS verborgen voor de andere gebruikers, die de betekenis van dat deel van het formulier niet meer begrijpen.



```
<form class="labelformat1">
  <fieldset class="fieldsetformat9968">
    <legend class="visually-hidden">Contact Information</legend>
  <p>
```

```
<label for="name9968">Name: </label>
<input type="text" id="name9968">
</p>
<p>
<label for="phone9968">Phone: </label>
<input type="text" id="phone9968">
</p>
<p>
<label for="email9968">Email: </label>
<input type="text" id="email9968">
</p>
<p>
<label for="address9968">Address: </label>
<input type="text" id="address9968">
</p>
<p>
<label for="city9968">City: </label>
<input type="text" id="city9968">
</p>
<p>
<label for="state9968">State: </label>
<input type="text" id="state9968">
</p>
<p>
<label for="zip9968">Zip: </label>
<input type="text" id="zip9968">
</p>
</fieldset>
</form>
```

[Slecht voorbeeld: bijschrift met slecht contrast](#)

Dit formulier werd correct gestructureerd, maar de <legend> voldoet niet aan de toegankelijkheidsnormen voor contrast.

Contact Information

Name:

Phone:

Email:

Address:

City:

State:

Zip:

```

<form class="labelformat1">
  <fieldset class="fieldsetformat9081">
    <legend class="legendformat9081">Contact Information</legend>
    <p>
      <label for="id85_name3">Name: </label>
      <input type="text" id="id85_name3">
    </p>
    <p>
      <label for="id85_phone3">Phone: </label>
      <input type="text" id="id85_phone3">
    </p>
    <p>
      <label for="id85_email3">Email: </label>
      <input type="text" id="id85_email3">
    </p>
    <p>
      <label for="id85_address3">Address: </label>
      <input type="text" id="id85_address3">
    </p>
    <p>
      <label for="id85_city3">City: </label>
  
```

```

<input type="text" id="id85_city3">
</p>
<p>
<label for="id85_state3">State: </label>
<input type="text" id="id85_state3">
</p>
<p>
<label for="id85_zip3">Zip: </label>
<input type="text" id="id85_zip3">
</p>
</fieldset>
</form>

```

Instructies en andere informatiemeldingen

Een van de beste manieren om de toegankelijkheid van formulieren te verbeteren is de gebruikers te helpen om fouten te vermijden. Daarom is het belangrijk om de gebruiker zo snel mogelijk duidelijke instructies te verstrekken, voordat hij gegevens begint in te voeren. Ontwikkelaars moeten de volgende punten voor ogen houden:

- Zorgen dat de labels en instructies duidelijk en informatief zijn;
- Instructies en informatiemeldingen bruikbaar maken voor hulptechnologieën;
- De beperkingen van formulervelden duidelijk vermelden;
- De verplichte velden duidelijk aanduiden.

a) Instructies voor formulieren, groepen en secties

Instructies voor groepen of secties moeten technisch aan de groep worden gekoppeld

Vrije tekst vermelden in het midden van een formulier zonder die aan een label of een veld te koppelen, kan de gebruiker verhinderen om die tekst te horen als hij de pagina doorloopt via de tabtoets (enkel op de elementen die de focus krijgen).

Er bestaat geen systeemeigen techniek om een beschrijving aan een groep te koppelen. De toevoeging van `aria-describedby` aan de `<fieldset>` of `<legend>` werkt niet en dus moeten we een andere manier vinden.

- Optie 1: instructies toevoegen aan de `<legend>`. Sommige schermlezers lezen echter de tekst van het bijschrift in elk gekoppeld formulerveld dat ze doorlopen. Daarom moet deze optie worden voorbehouden voor korte instructies die bij elk veld consistent blijven;
- Optie 2: `aria-describedby` gebruiken om de instructies aan een van de velden van de groep (bij voorkeur het eerste) te koppelen. Op deze manier worden de groepsinstructies slechts één keer voorgelezen en worden ze door de schermlezer voorgelezen bij raadpleging met de tabtoets;
- Optie 3: de instructies vóór het begin van het formulier toevoegen als ongekoppelde vrije tekst en hopen dat de gebruiker de inhoud van de boodschap leest voordat hij het formulier betreedt. Deze oplossing omzeilt het probleem en laat de

verantwoordelijkheid aan de gebruiker, wat niet ideaal is, hoewel veel internetgebruikers de tekst op die manier lezen: de ene paragraaf na de andere.

Goed voorbeeld: Instructies voor de groep in de <legend>

Het woord 'Required' staat in de <legend> om aan te geven dat alle velden van de <fieldset> verplicht zijn. Schermlezers lezen de <legend>-tekst wanneer de gebruiker in het eerste veld terechtkomt.

Login Information (Required)

Username:

Password:

```
<fieldset>
<legend>Login Information (Required)</legend>
  <label for="username">Username: </label>
  <input type="text" id="username"></p>
  <label for="password">Password: </label>
  <input type="password" id="password"></p>
</fieldset>
```

Goed voorbeeld: De instructies voor de groep zijn gekoppeld via aria-describedby

Het attribuut aria-describedby wordt gebruikt om de instructies te koppelen aan 'password', het eerste veld van het formulier.

Create Account

Username:

Password fields must match

Password:

Re-enter password:

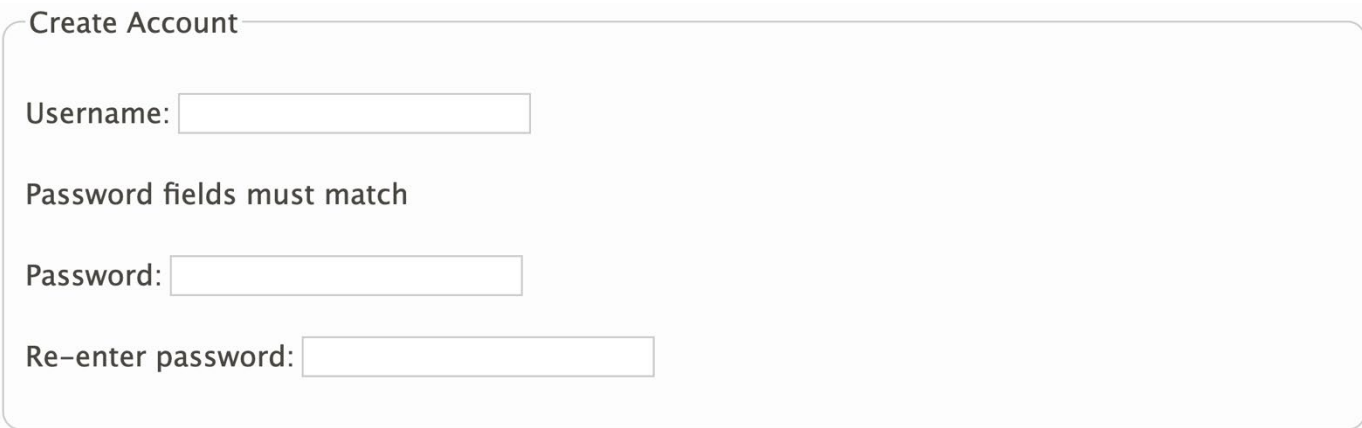
```
<fieldset>
<legend>Create Account</legend>
  <p><label for="username">Username: </label>
  <input type="text" id="username"></p>
  <p id="mustmatch">Password fields must match</p>
```



```
<p><label for="password">Password: </label>
  <input type="password" id="password" aria-describedby="mustmatch"></p>
<p><label for="password2">Re-enter password: </label>
  <input type="password" id="password2"></p>
</fieldset>
```

Slecht voorbeeld: De instructies staan in een paragraaf die niet aan de groep werd gekoppeld.

De inhoud van de paragraaf <p> krijgt geen focus. Als hij niet wordt gekoppeld, bijvoorbeeld via aria-describedby, krijgen de gebruikers van een schermlezer de tekst niet te horen.



```
<fieldset>
<legend>Create Account</legend>
<p><label for="username">Username: </label>
  <input type="text" id="username"></p>
<p>Password fields must match</p>
<p><label for="password">Password: </label>
  <input type="password" id="password"></p>
<p><label for="password2-">Re-enter password: </label>
  <input type="password" id="password2-"></p>
</fieldset>
```

Goed voorbeeld: Instructies gekoppeld aan meerdere velden van het formulier

In bepaalde gevallen is het nuttig om een boodschap aan meerdere velden, maar niet aan alle velden van het formulier te koppelen. In dit voorbeeld geldt de boodschap 'Username en password mogen geen spaties bevatten' slechts voor twee velden.

Create Account

Username and password must not contain spaces

Username:

Password:

```

<fieldset>
<legend>Create Account</legend>
  <p><strong>Username and password
  <span id="mustnot">must not contain spaces</span></strong></p>
  <p><label for="username">Username:</label>
    <input type="text" id="username" aria-describedby="mustnot"></p>
  <p><label for="password">Password:</label>
    <input type="password" id="password" aria-describedby="mustnot"></p>
</fieldset>

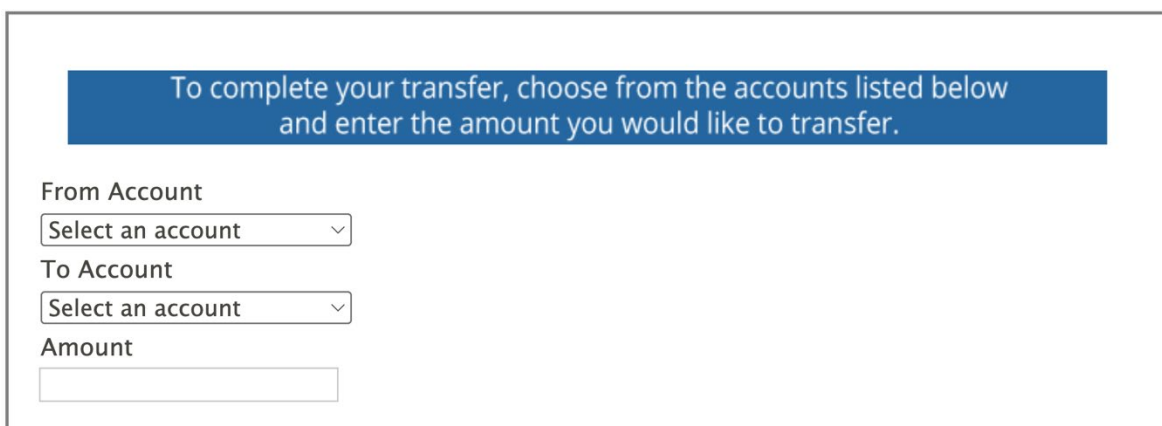
```

Instructies voor groepen of secties moeten technisch bepaalbaar zijn.

De instructies moeten in echte tekstvorm worden vermeld om voor alle gebruikers en hulptechnologieën leesbaar te zijn.

Slecht voorbeeld: Instructies in een afbeelding zonder tekstalternatief

De instructietekst wordt weergegeven in de vorm van een afbeelding zonder alt-attribuut of met een leeg alt-attribuut.



To complete your transfer, choose from the accounts listed below and enter the amount you would like to transfer.

From Account

To Account

Amount

Opmerking: Zelfs als de afbeelding een correct alternatief bevat, blijft een tekstafbeelding problematisch voor gebruikers van een zoomfunctie of voor internetgebruikers die de tekstweergave moeten aanpassen (kleur, tekstgrootte, lettertype, tekstafstand ...).

Goed voorbeeld: De instructies worden vermeld in tekstvorm en gekoppeld aan het eerste element van het formulier.

To complete your transfer, choose from the accounts listed below and enter the amount you would like to transfer.

From Account

To Account

Amount

Er wordt echte tekst vermeld en aan het eerste veld gekoppeld aan de hand van de [hierboven beschreven techniek met gebruik van aria-describedby](#)

Instructies voor groepen of secties moeten betekenisvol zijn.

De instructies moeten duidelijk, nauwkeurig en informatief zijn. De gebruikers moeten op voorhand de nodige instructies kennen om hun gegevens in te voeren.

Slecht voorbeeld: De instructie is te vaag en niet expliciet genoeg

Most fields are required.

Contact Information

Name:

Phone:

Email:

Address:

City:

State:

Zip:

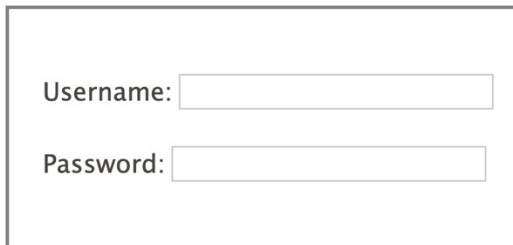
Hoewel de instructie correct aan de formulervelden werd gekoppeld, is de tekst te vaag om de gebruiker correct te informeren.

De instructies voor groepen of secties moeten zichtbaar zijn.

De instructies moeten zichtbaar zijn voor alle gebruikers. De aria-attributen zijn enkel bestemd voor schermlezers en er bestaat een risico dat onaanpasbare teksten met een te klein lettertype of in een onvoldoende contrasterende kleur voor sommige internetgebruikers niet zichtbaar zijn.

Slecht voorbeeld: Instructies in een verborgen tekst

In dit voorbeeld worden de instructies visueel verborgen in CSS. De gebruikers van een schermlezer horen de instructies wel, maar ze zijn niet zichtbaar voor ziende internetgebruikers.



A screenshot of a login form. It contains two input fields: 'Username:' followed by a text input box, and 'Password:' followed by a password input box. The form is enclosed in a simple rectangular border.

```
<div class="visually-hidden" id="desc">Must not contain spaces</div>
```

```
<p>
```

```
<label for="username">Username:</label>
```

```
<input type="text" id="username" name="username" aria-describedby="desc">
```

```
</p>
```

```
<p>
```

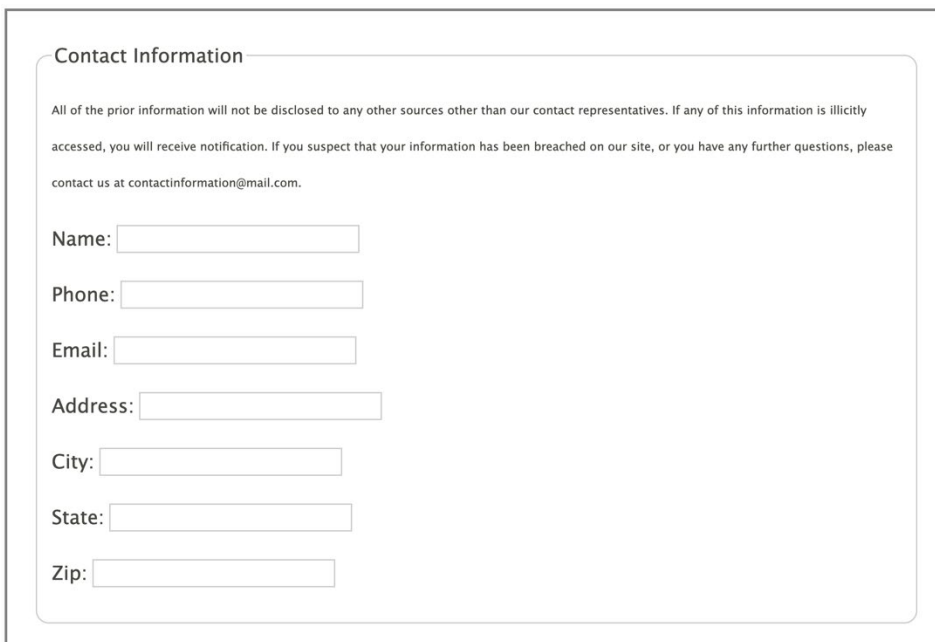
```
<label for="password">Password:</label>
```

```
<input type="password" id="password" name="password" aria-describedby="desc">
```

```
</p>
```

Slecht voorbeeld: Te kleine instructietekst

Mensen die geen zoomfunctie gebruiken lopen het risico de te kleine tekst niet te kunnen lezen.



A screenshot of a contact form titled 'Contact Information'. At the top, there is a large block of text: 'All of the prior information will not be disclosed to any other sources other than our contact representatives. If any of this information is illicitly accessed, you will receive notification. If you suspect that your information has been breached on our site, or you have any further questions, please contact us at contactinformation@mail.com.' Below this text are several input fields for 'Name:', 'Phone:', 'Email:', 'Address:', 'City:', 'State:', and 'Zip:'. The text is significantly smaller than the form fields, making it difficult to read.


Instructies voor groepen of secties moeten zich visueel dicht bij de elementen waarnaar ze verwijzen bevinden.

Slecht voorbeeld: De instructies zijn van het formulier gescheiden

In dit voorbeeld worden de instructies gescheiden door een afbeelding en een tekst. Er bestaat een risico dat de internetgebruikers vergeten of over het hoofd zien dat het paswoord en de

login geen spaties mogen bevatten omdat de boodschap zich niet meer in hun gezichtsveld bevindt. Dat geldt voor zoomgebruikers, gebruikers van schermlezers, mensen met een kort en gebrekkig geheugen ... alle gebruikers met een scherm dat niet het volledige formulier toont.

Password and Username must not contain spaces.



We want you to become a part of our network. You can become a member by just signing up!
We offer you special features for just being a part of our group and give you access to extended chances for other offers. Join now to take part in the exclusive benefits we offer to our members.


Create Account

Username:

Password:

[Goed voorbeeld: De instructies bevinden zich vlak voor de invoervelden](#)

De instructies staan visueel in de nabijheid van de betrokken velden, waaraan ze werden gekoppeld via het attribuut aria-describedby.



We want you to become a part of our network. You can become a member by just signing up! We offer you special features for just being a part of our group and give you access to extended chances for other offers. Join now to take part in the exclusive benefits we offer to our members.

Create Account

Password and Username must not contain spaces

Username:

Password:

```


<p>We want you to become a part of our network. You can become a member by
  just signing up! We offer you special features for just being a part of our
  group and give you access to extended chances for other offers. Join now to
  take part in the exclusive benefits we offer to our members.</p>
<fieldset>
<legend>Create Account</legend>
<p><strong>Username and password
  <span id="mustnot">must not contain spaces</span></strong></p>
<p><label for="username">Username:</label>
  <input type="text" id="username" aria-describedby="mustnot"></p>
<p><label for="password">Password:</label>
  <input type="password" id="password" aria-describedby="mustnot"></p>
</fieldset>
    
```

Instructies voor groepen of secties moeten zich in het DOM dicht bij de elementen waarnaar ze verwijzen bevinden.

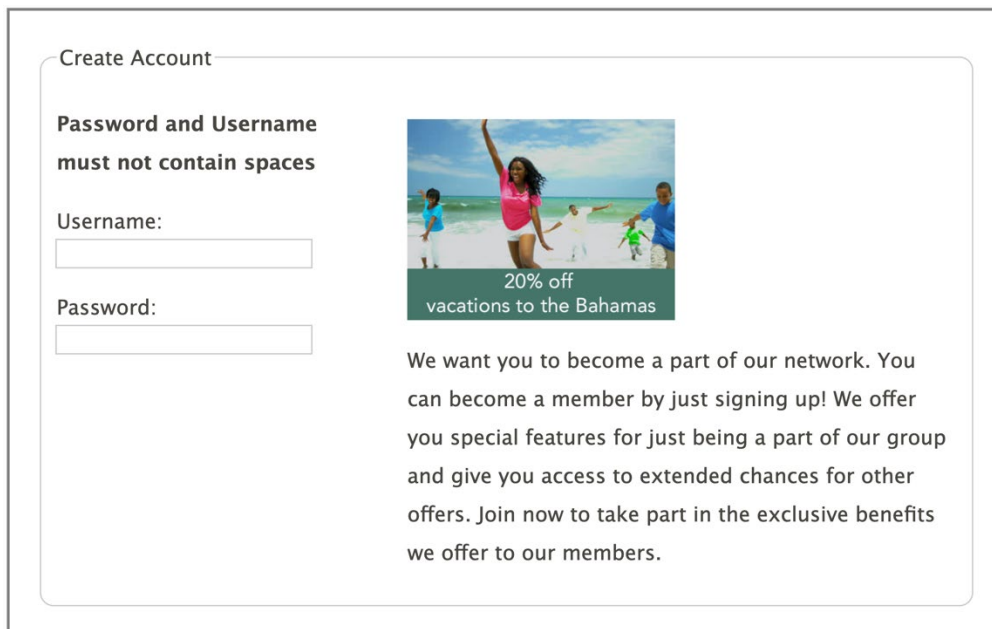
Schermlezers lezen meestal de inhoud van de pagina in de volgorde waarin die in het DOM verschijnt. Instructies moeten zich dus in de nabijheid bevinden van de elementen waarop ze betrekking hebben.

In de meeste gevallen waarin de informatie zich visueel vlak bij haar elementen bevindt, is die nabijheid ook effectief in het DOM. Maar sommige CSS-posities kunnen een nabijheid tonen die niet met de realiteit overeenstemt.

Slecht voorbeeld: Een scheiding in het DOM veroorzaakt door een float-positie in CSS

In dit geval lijken de instructies 'Username and password must not contain spaces' visueel vlak bij hun overeenkomstige velden te staan, maar dat is niet het geval in het DOM.

In het formulier werden tussen de instructies en de velden een volledige paragraaf en een afbeelding ingelast. In CSS wordt hun positie aan de rechterkant van het formulier weergegeven, terwijl ze in het DOM vlak na de instructietekst staan.



```
<fieldset>
```

```
<legend>Create Account</legend>
```

```
<p><strong>Username and password<br>
must not contain spaces</strong></p>
```

```
<div id="float">
```

```

```

```
<p>We want you to become a part of our network. You can become a member by just signing
up! We offer you special features for just being a part of our group and give you access to
extended chances for other offers.
```

```
Join now to take part in the exclusive benefits we offer to our members.
```

```

</p>
</div>
<p><label for="username">Username:</label><br>
  <input type="text" id="username"></p>
<p><label for="password">Password:</label><br>
  <input type="password" id="password"></p>
</fieldset>

```

b) Instructies voor formulervelden

De instructies voor een element moeten technisch aan dat element worden gekoppeld

De specifieke instructies om gegevens in te voeren moeten technisch aan de velden, knoppen en bedieningselementen gekoppeld zijn. De koppeling kan semantisch via het gekoppelde <label> gebeuren, maar ook via het attribuut aria-describedby.

Goed voorbeeld: De instructies zijn beschikbaar via aria-describedby

Email: (Must be a valid email address)

```

<p>
  <label for="email">Email:</label>
  <input type="email" name="email" id="email" aria-describedby="input-instructions">
  <span id="input-instructions">(Must be a valid email address)</span>
</p>

```

Slecht voorbeeld: De instructies zijn niet gekoppeld aan de <input>

Email: (Must be a valid email address)

```

<p>
  <label for="email">Email:</label>
  <input type="email" name="email" id="email">
  <span>(Must be a valid email address)</span>
</p>


```

De instructies voor een element moeten technisch aanwezig zijn in de vorm van expliciete tekst

Zonder expliciete tekst die aan het veld werd gekoppeld, kan een schermlezer de gebruiker geen informatie verstrekken. De mogelijke methodes zijn het inlassen van de tekst in het correct gekoppelde <label> of het gebruik van aria-attributen.

Slecht voorbeeld: Een icon font zonder gekoppelde tekst

Het bestandsformaat (Microsoft Word) wordt enkel gepreciseerd via een pictogram dat het Word-logo afbeeldt. Zonder tekstalternatief kan een schermlezer het gewenste formaat niet vermelden.

Upload file : Aucun fichier choisi

<p>

```
<label for="upload">Upload file</label>
```

```
<span class="far fa-file-word fa-lg" id="desc"></span>:
```

```
<input type="file" id="upload" aria-describedby="desc">
```

</p>

Goed voorbeeld: een icon font met een gekoppeld tekstalternatief

In hetzelfde voorbeeld als hierboven wordt na het pictogram de tekst 'in MS Word format' toegevoegd om alle gebruikers de mogelijkheid te bieden het gewenste bestandsformaat te begrijpen. Deze tekst wordt aan het veld gekoppeld via het attribuut aria-describedby om gebruikers van schermlezers de mogelijkheid te bieden om het verband te leggen tussen het veld en de boodschap.

Upload file  in MS Word format: Aucun fichier choisi

<p>

```
<label for="upload">Upload file</label>
```

```
<span id="desc">in MS Word format</span> <span class="far fa-file-word fa-lg"></span>:
```

```
<input type="file" id="upload" aria-describedby="desc">
```

</p>

De instructies voor een element moeten expliciet en ondubbelzinnig zijn

Als een veld een specifiek formaat vereist, zoals bijvoorbeeld een datumformaat, moet die informatie duidelijk worden beschreven.

Zo is de tekst 'Voer een geldig datumformaat in' zinloos. Wat moet worden vermeld is bijvoorbeeld 'MM/YYYY'.

Een bijzondere opmerking over datumformaten: MM en YYYY zijn in het Engels en moeten in alle talen vertaald worden. Er wordt overigens aanbevolen om een concreet voorbeeld te geven, zoals '11/2022'.

De instructies voor een element moeten zichtbaar en groot genoeg zijn, visueel en in het DOM dicht bij hun overeenkomstige elementen staan, geen aanwijzingen geven via zintuiglijke elementen (vorm of kleur) ...

Deze criteria werden al eerder besproken en geïllustreerd.

c) Verplichte velden

Duidelijk geïdentificeerde verplichte velden stellen de gebruiker in staat om tijd te winnen bij het invullen van het formulier en heen-en-weer gaan en foutjes te vermijden.

Meestal wordt het verplichte karakter aangegeven in het <input>-veld via het attribuut aria-required="true", gecombineerd met een visuele identificatie voor mensen die geen schermlezer gebruiken.

Verplichte velden moeten technisch als dusdanig worden geprogrammeerd

Aria-required="true"

Het gebruik van dit attribuut op een veld is een heel goede manier om het verplichte karakter ervan aan te geven. De schermlezer zegt dan 'verplicht' of 'vereist' na het label van het veld te hebben voorgelezen. Deze informatie is echter onzichtbaar en moet dus door een visuele aanwijzing worden versterkt.

Het HTML-attribuut 'required'

Voor de gebruikers van schermlezers is het gedrag identiek aan `aria-required="true"`, maar er wordt een extra visueel gedrag toegevoegd dat de gebruiker verplicht om het veld in te vullen als het leeg werd gelaten.

Dit attribuut wordt echter niet door alle browsers ondersteund en het verplichte karakter moet, net zoals bij `aria-required`, op voorhand worden vermeld voor ziende gebruikers.

`aria-required` lijkt dus meer aangewezen, temeer omdat wordt gevraagd om `required` en `aria-required="true"` niet voor hetzelfde veld te gebruiken om informatieredundantie te vermijden voor schermlezergebruikers.

Goed voorbeeld: De verplichte velden gebruiken `aria-required="true"` en een visueel sterretje

Fields with asterisks (*) are required.

First Name:*

Last Name:*

Age:

```
<p>Fields with asterisks (*) are required.</p>
```

```
<p>
```

```
<label for="firstname">First Name:*</label>
```

```
<input id="firstname" name="firstname" type="text" aria-required="true">
```

```
</p>
```

```
<p>
```

```
<label for="lastname">Last Name:*</label>
```

```
<input id="lastname" name="lastname" type="text" aria-required="true">
```

```
</p>
```

```
<p>
```

```
<label for="age">Age:</label>
```

```
<input id="age" name="age" type="text">
```

```
</p>
```

Opmerkingen:

- Het feit dat de verplichte velden door een sterretje worden aangeduid moet worden vermeld voordat het formulier wordt ingevuld.
- De schermlezer leest het sterretje in het <label> en het woord ‘verplicht’ of ‘vereist’ gekoppeld aan `aria-required="true"`. Dat gedrag werkt niet blokkerend, maar de redundantie van de informatie kan worden voorkomen met de in het volgende voorbeeld beschreven techniek.
- Het sterretje (*) mag worden vervangen door de voluit geschreven vermelding van het woord ‘(verplicht)’ in het label van de betreffende velden.

[Goed voorbeeld: De verplichte velden gebruiken `aria-required="true"` en een visuele aanduiding die onzichtbaar is voor schermlezers.](#)

Deze techniek is in grote lijnen dezelfde als in het vorige voorbeeld, maar in dit geval krijgt de gebruiker het sterretje dat in de labels staat niet te horen, omdat het wordt weergegeven door een afbeelding in een CSS-klasse.

De gebruiker kan echter verrast worden doordat hij in het begin van het formulier hoort dat velden met een sterretje verplicht zijn en daarna niets meer over die sterretjes hoort. De <p> die ‘Fields with asterisks (*) are required’ bevat, kan echter voor schermlezers worden verborgen dankzij het attribuut `aria-hidden="true"` om de oplossing compleet te maken.

Fields with asterisks (*) are required.

First Name: *

Last Name: *

Age:

<p>Fields with asterisks (*) are required.</p>

<p>

<label for="firstname">First Name: </label>

<input id="firstname" name="firstname" type="text" aria-required="true">

</p>

<p>

<label for="lastname">Last Name: </label>

<input id="lastname" name="lastname" type="text" aria-required="true">

</p>

<p>

<label for="age">Age:</label>

<input id="age" name="age" type="text">

</p>

Het validatieproces van het formulier moet een foutmelding bevatten die niet-ingevoerde verplichte velden na de verzendpoging visueel en technisch identificeert.

Een duidelijke boodschap die zowel visueel als voor schermlezers begrijpelijk is, moet na verzending van het formulier de verplichte velden identificeren die niet werden ingevuld.

Opmerkingen:

- De gebruiker moet feedback krijgen over het succes van de bewerking: geslaagd of fouten in het formulier (identificatie in de paginatitel, focus op de eerste foutmelding ...).
- De gebruiker moet velden met fouten gemakkelijk kunnen identificeren zonder alle informatie opnieuw in te voeren of systematisch alle velden opnieuw te doorlopen.
- Bij een niet-gevalideerd formaat, moeten de foutmeldingen een concreet voorbeeld geven van de informatie zoals ze moet worden ingevoerd.
- De foutmelding moet voor alle gebruikers zichtbaar zijn (tekst, voldoende contrast, grootte en positie van de melding, bruikbaar voor hulptechnologieën ...).

Goed voorbeeld: De foutmeldingen worden visueel en technisch geïdentificeerd

In dit voorbeeld zijn de foutmeldingen technisch gekoppeld aan de overeenkomstige velden via het attribuut `aria-describedby`.

Opmerkingen:

- In de foutmelding zou een concreet datumvoorbeeld moeten worden vermeld.
- Het attribuut `value="User name"` is niet verplicht en kan bij sommige gebruikers zelfs tot misverstanden leiden. Het is aanbevolen om dit veld leeg te laten.



The screenshot shows a form with two input fields. The first field is labeled 'Username:' and contains the text 'User name'. The second field is labeled 'Expiration:' and is empty. To the right of the 'Expiration:' field, there is a red error message: 'Error: Date is required.'.

```
<p>
```

```
<label for="username">Username:</label>
```

```
<input type="text" name="username" id="username" value="User name">
```

```
</p>
```

```
<p class="error">
```

```
<label for="expire">Expiration:</label>
```

```
<input type="text" name="expire" id="expire"
```

```
  aria-required="true"
```

```
  aria-invalid="true"
```

```
  aria-describedby="expDesc">
```

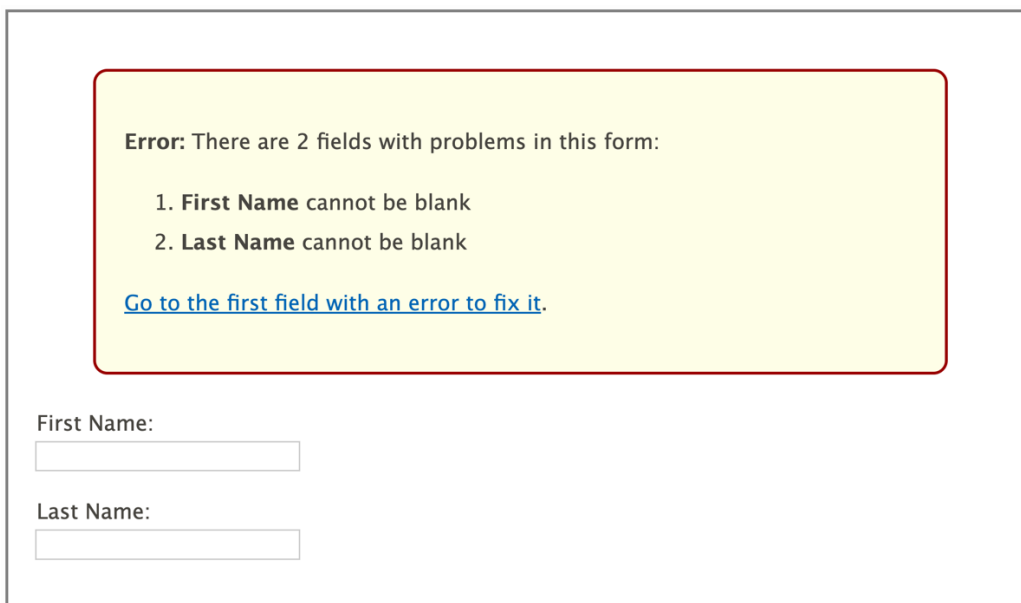
```
<span id="expDesc">Error: Date is required.</span>
```

```
</p>
```

Goed voorbeeld: Kort overzicht van de fouten

Dit voorbeeld toont een overzicht van de fouten na de poging om het formulier te verzenden. Het heeft de volgende kenmerken:

- Er is informatie aanwezig over het aantal vastgestelde fouten;
- Duidelijke uitleg over de velden met fouten;
- Duidelijke uitleg over het fouttype;
- Expliciete link naar het eerste veld dat een fout bevat;
- Gebruik van aria-required om verplichte velden te identificeren;
- Gebruik van aria-invalid om aan te geven dat een veld een fout bevat (leeg of niet-conform formaat). Gecombineerd met aria-required mag de parameter niet op 'true' staan zolang het formulier niet werd verzonden. De parameters 'true', 'false', 'grammar' of 'spelling' worden gewijzigd via JavaScript.



```

<div class="alert">
<p><strong>Error:</strong>
  There are 2 fields with problems in this form:</p>
<ol>
  <li><strong>First Name</strong> cannot be blank</li>
  <li><strong>Last Name</strong> cannot be blank</li>
</ol>
<a href="#fname" id="errorSummaryLink">Go to the first field with an error to fix it</a>.
</div>
<label for="fname">First Name:</label><br>
<input type="text" id="fname" aria-required="true" aria-invalid="true">
<label for="lname">Last Name:</label><br>

```

```
<input type="text" id="lname" aria-required="true" aria-invalid="true">
```

De bedoeling van de gebruikelijke velden voor het verzamelen van persoonlijke gegevens moet technisch worden bepaald op basis van de officiële W3C-bronnen voor interfacecomponenten.

De bedoeling van dit criterium is om de gebruiker te helpen om de formulervelden sneller en foutloos in te vullen door automatisch zijn correct geïdentificeerde persoonlijke gegevens in de juiste velden in te vullen (autocomplete).

Dat criterium wordt geïmplementeerd door het attribuut autocomplete="parameter" in de overeenkomstige <input> in te lassen.

[De lijst met bruikbare parameters kan op de W3C-website worden geraadpleegd](#)

Voorbeeld van het gebruik van autocomplete:

```
<input class="loginPassword" type="password" id="loginPassword" name="password" aria-
required="true" autocomplete="current-password" required="required"
oninvalid="this.setCustomValidity('Enter password to login')"
oninput="this.setCustomValidity('')"/>
```

Dynamische formulieren

a) Progressieve formulieren

In bepaalde gevallen wordt aan de gebruikers gevraagd de informatie in opeenvolgende stappen in te voeren en beïnvloeden de antwoorden op elke stap de andere stappen van het formulier.

Om die procedure toegankelijk te maken, moeten de volgende goede praktijken in overweging worden genomen:

Wijzig de opties die nog moeten komen, maar niet de opties die al voorbij zijn.

Gebruikers van schermlezers lezen de inhoud op een meer lineaire manier. Als ze een plaats in het formulier hebben bereikt, zullen ze niet op hun stappen terugkeren om te controleren of er een wijziging plaatsvond.

We moeten er dus voor zorgen dat de optiewijzigingen geen invloed hebben op wat in het DOM staat vanaf de positie van de gebruiker.

Bied de gebruiker de mogelijkheid om keuzes uit vorige stappen te wijzigen.

In bepaalde gevallen zal het voor de gebruiker nodig zijn om de antwoorden die hij in een vorige stap gaf te kunnen wijzigen. Het formulier moet hem in staat stellen om op zijn stappen terug te keren en zijn antwoorden te corrigeren.

Beperk het aantal stappen per scherm.

Een overdaad aan stappen of vragen op één pagina kan verwarrend zijn en navigatieproblemen of misverstanden opleveren.

De vragen moeten in samenhangende stappen worden gegroepeerd en de navigatie tussen de verschillende stappen om de vragen/antwoorden te herzien moet gemakkelijk zijn. Een andere praktische oplossing bestaat erin een overzicht van de vragen/antwoorden te tonen (in de vorm van een navigatiesysteem) en de gebruiker de mogelijkheid te bieden om naar elke vraag terug te keren door rechtstreeks op de overeenkomstige link te klikken.

Vermeld het aantal stappen.

De gebruikers stellen het op prijs om op elk ogenblik het nummer van de stap waarin ze zich bevinden en het totale aantal stappen te kennen. Dat systeem moet natuurlijk ook toegankelijk zijn via schermlezers.

Maak toetsenbordnavigatie mogelijk.

De focus moet bij elke stap van de procedure correct worden beheerd. In functie van de manier waarop het formulier is opgebouwd, moet de focus op het meest logisch mogelijke element worden geplaatst om de navigatie soepel en toegankelijk te maken. Als het element niet van nature de focus kan krijgen, bijvoorbeeld een kopniveau (header) of een container, dan kan de focus worden geïmplementeerd via `tabindex="-1"`.

De knoppen die het mogelijk maken om van de ene stap naar de andere te navigeren moeten ook de focus kunnen krijgen.

Zorg er ook voor dat de focus zichtbaar is (contrastregels, kadertjes ...). De CSS-eigenschappen voor zweven over de focuspositie (hover) moeten ook worden geïmplementeerd.

Overweeg een beknopte samenvatting van de antwoorden

Het kan vervelend zijn om alle stappen van een formulier te moeten overlopen om zich te herinneren welke informatie werd ingevoerd alvorens het formulier te verzenden. Een beknopte samenvatting maakt dat proces gemakkelijker en stelt de gebruiker gerust, terwijl het ook minder tijdverlies veroorzaakt.

Het kan ook interessant zijn om de samenvatting van de antwoorden technisch aan de verzendknop te koppelen, zoals in het onderstaande codevoorbeeld:

```
<div id="summary">Summary goes here ... </div>
```

```
<button aria-describedby="summary">Submit</button>
```

Good Example: Progressive Form

Buy a T-Shirt

Gender: Female

Style: Ladies V-Neck

Color: Red

Step 3 of 4 — Choose a color:

White

Black

Red

Pink

Yellow

Back
Continue

Good Example: Progressive Form

Buy a T-Shirt

Gender: Female

Style: Ladies V-Neck

Color: Red

Step 4 of 4 — Buy it!

Back
Purchase

b) Contextwijzigingen

Gebruikers kunnen verward raken door onverwachte wijzigingen van de context. Een contextwijziging kan worden gedefinieerd als een wijziging van:

- de user agent: bijvoorbeeld als de klik op een link of een knop de browser afsluit en een andere toepassing opent;
- de viewport: bijvoorbeeld als het scherm omhoog of omlaag scrolt, als een element visueel een ander element vervangt of van positie verandert of als er een nieuw venster wordt geopend;
- de focus: bijvoorbeeld als de focus zich naar een ander element verplaatst;
- de inhoud: bijvoorbeeld als de betekenis van een pagina aanzienlijk verandert of elementen van de pagina worden herschikt.

Al die wijzigingen zijn toegestaan, maar voor de digitale toegankelijkheid mogen ze niet automatisch gebeuren zonder dat de gebruiker zich ervan bewust is dat er iets gaat veranderen en zonder dat hij de aard van de verandering en de impact van zijn actie kent.

De focus op een element leggen of over een element zweven, mag niet automatisch een contextwijziging veroorzaken, behalve als de gebruiker op voorhand over de aard van de wijziging werd gewaarschuwd.

Onverwacht gedrag koppelen aan de focus op een element kan alle internetgebruikers en vooral de gebruikers van schermlezers in verwarring brengen.

Slechte voorbeelden:

- De focus op een formulerveld opent automatisch een venster dat informatie verstrekt over de manier waarop het veld moet worden ingevuld.
- Met de muis over een zone van de pagina zweven (hover) opent automatisch een venster.

De wijziging van de waarde van een element mag niet automatisch een contextwijziging veroorzaken, behalve als de gebruiker op voorhand over de aard van de wijziging werd gewaarschuwd.

Opmerking: De contextwijziging is toegestaan bij het klikken op de verzendknop van het formulier (of een gelijkaardige actie) omdat de gebruiker zijn actie heeft voltooid en die wijziging verwacht.

Slechte voorbeelden:

- Een <select>-lijst wordt gebruikt als middel om door de website te navigeren. De inhoud van de pagina verandert zodra de gebruiker een van de opties selecteert;
- Een formulier bevat drie velden om één nummer in te voeren. Zodra het maximum aantal cijfers in één veld wordt bereikt, verspringt de focus automatisch naar het volgende veld zonder de gebruiker te waarschuwen.

c) Custom formulervelden

[De ARIA-documentatie](#) bespreekt in detail de technieken om de verschillende widgets in JavaScript te implementeren. We sommen hier de voornaamste aandachtspunten op om die componenten te implementeren.

Voor zover mogelijk moeten bij voorkeur systeemeigen HTML-elementen worden gebruikt.

HTML-formulierelementen zijn inherent toegankelijk zonder JavaScript-aanvullingen om ze bruikbaar te maken via het toetsenbord of hulptechnologieën.

Bovendien zijn de gebruikers gewend aan hun gebruik en kennen ze hun standaardgedrag, zodat er geen nood is aan specifieke instructies of leercurves.

Custom formulierelementen moeten zoveel mogelijk functioneren en reageren als systeemeigen HTML-elementen.

Zo moet bijvoorbeeld een selectievakje (checkbox) aan- en uitgevinkt kunnen worden via de spatiebalk en een boodschap moet zich dicht bij de door schermlezers aangekondigde boodschap bevinden. Let er vooral op om elke component toetsenbordvriendelijk te maken.

Het voor elke component verwachte gedrag is beschikbaar op de [W3C-WAI-website](#)

Custom formulierelementen moeten een toegankelijke naam, een rol en een waarde hebben.

Al die informatie is online beschikbaar via de documentatie over de [ARIA-specificaties](#)

Naam (Name):

Het label van het element, dat zijn toegankelijke naam vermeldt, zoals bijvoorbeeld 'date picker', 'Submit' ... wordt vaak opgenomen in een aria-label of een aria-labelledby.

Rol (Role):

De bedoeling of de functie van een element, zoals bijvoorbeeld 'dialog', 'menu' of 'tabpanel' ... moet worden gedefinieerd. De lijst met rollen is beschikbaar op de [W3C-ARIA-website](#)

Waarde (Value):

De attributen of de huidige toestand van het element, zoals bijvoorbeeld `aria-selected="true"` of `aria-expanded="false"`. De waarden moeten worden geüpdatet indien nodig (true wordt false ...)

Updates en statuswijzigingen die niet via HTML- of ARIA-methodes kunnen worden meegedeeld moeten via aria-live-meldingen worden meegedeeld.

Schermlezers moeten de wijzigingen die aan de pagina werden aangebracht herkennen. Bijvoorbeeld: 'tabel gesorteerd op naam, oplopend' of 'resultaten gesorteerd op populariteit' ... Die wijzigingen kunnen dan aan de gebruikers worden gemeld.

De verschillende waarden en attributen voor aria-live zijn beschikbaar [op de W3C-WAI-website](#)

Validatie van de formulieren

Zelfs met de best mogelijke instructies kunnen we fouten in formulieren niet vermijden. De verwerking van fouten is een belangrijk aspect van de toegankelijkheid. De WCAG 2.1-richtlijnen dringen sterk aan op de manier waarop gebruikers worden gewaarschuwd om hen te helpen bij het corrigeren van hun fouten. Een aantal punten verdienen bijzondere aandacht:

- Geef alle gebruikers positieve of negatieve feedback;
- Bewaar de correcte informatie zoveel mogelijk;
- Toon foutmeldingen in hun context om een snelle correctie mogelijk te maken;
- Geef de gebruikers voldoende informatie om hun fouten te corrigeren. De meldingen moeten:
 - duidelijk zijn,
 - de aandacht trekken,
 - zich door hun stijl van invoervelden onderscheiden,
 - volledig toegankelijk zijn.

a) Fouten voorkomen: Goede praktijken

Het is de bedoeling om formulieren samen te stellen die voldoende expliciet zijn om fouten van de gebruikers te voorkomen. Deze handleiding heeft al heel wat preventietechnieken besproken. Een korte samenvatting:

- De tekst van labels is nauwkeurig en beschikbaar voor alle gebruikers;
- Als de labeltekst niet voldoende instructies geeft over de bedoeling van een veld, moeten bijkomende instructies worden verstrekt;
- Als dat nodig en gepast is, moet het in het veld verwachte gegevenstype worden beperkt: bijvoorbeeld teksttype voor een postcode, selectielijsten, keuzerondjes of selectievakjes;
- Weergaven van het formulier worden best stap na stap aangeboden om de gebruiker niet te overstelpen met een overdaad aan gelijktijdige gegevens;
- Gebruik vertrouwde design patterns die de gebruiker al kent en zonder teveel leertijd kan oppikken. Gebruik zoveel mogelijk systeemeigen HTML-elementen;
- De gebruiker helpen om zijn aandacht toe te spitsen door labels en instructies dicht bij hun respectieve velden te plaatsen, door afleiding te beperken en door elementen te gebruiken om hem bij zijn procedure te begeleiden. Interactieve boodschappen visueel vanzelfsprekend maken;
- Een samenvatting opmaken van de informatie die wordt meegedeeld voordat de verzendknop wordt gebruikt en de gebruiker de mogelijkheid bieden om zijn gegevens te controleren, vooral als de actie onomkeerbaar is;

- Na de verzending wordt een samenvatting van het resultaat getoond.
- Stel de gebruiker zo mogelijk in staat om de gegevens in hun oorspronkelijke staat te herstellen door een technische ‘undo’-oplossing, door contact op te nemen met een dienstverlener of door een procedure die een correctie van zijn gegevens mogelijk maakt;
- Bezorg de gebruiker als dat past een e-mailbevestiging als extra controlemiddel en om hem gerust te stellen over de rest van de verwerking.

b) Fouten identificeren

De foutfeedback moet meteen na de verzending van het formulier beschikbaar zijn.

De gebruiker moet zijn eventuele fouten meteen na de verzending kennen. Zonder feedback over die informatie, kan de gebruiker de indruk krijgen dat het formulier niet werkt en de actie staken. Dat is met name het geval voor gebruikers van schermlezers.

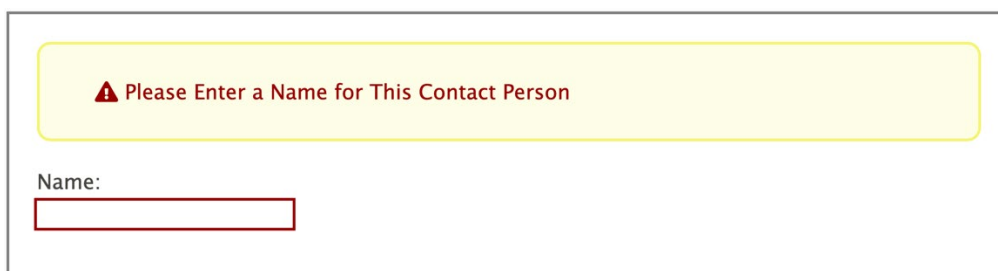
De fouten kunnen worden meegedeeld na de volledige verzending, na elke stap of na het invullen van elk veld.

De foutfeedback moet technisch aan het betreffende element worden gekoppeld.

Als de foutmeldingen aan hun overeenkomstige elementen gekoppeld zijn, kan de gebruiker ze gemakkelijker terugvinden om ze snel te corrigeren.

Goed voorbeeld: De foutmelding is technisch aan haar element gekoppeld

In dit voorbeeld koppelt het attribuut aria-describedby de foutmelding aan het betreffende veld.



```
<div class="formborder">
<div class="error">
  <span class="fa fa-exclamation-triangle" role="img" aria-label="Error: "></span>
  <span id="errorMsg">Please Enter a Name for This Contact Person</span>
</div>
<p>
  <label for="name">Name: </label><br>
  <input type="text" name="name" id="name" aria-invalid="true" aria-describedby="errorMsg">
</p>
</div>
```

De foutfeedback moet technisch bepaalbaar zijn.

De tekst van de technisch gekoppelde melding moet voldoende duidelijk en expliciet zijn. Naargelang het geval kunnen de meldingen worden vermeld in de attributen alt, aria-label, aria-labelledby en aria-describedby.

Anders kunnen gebruikers van schermlezers hun fouten niet corrigeren.

Slecht voorbeeld: foutmelding zonder technisch bepaalbare boodschap

In dit voorbeeld is het voor de gebruiker van een schermlezer onmogelijk om de foutmelding te begrijpen, ondanks de driehoek met uitroepteken die visueel de aandacht op de fout vestigt.



```
<span class="fa fa-exclamation-triangle"></span><label for="name"> Name: </label><br>
```

```
<input type="text" name="name" id="name" aria-invalid="true">
```

Goed voorbeeld: Foutmelding met technisch bepaalbare boodschap

In dit voorbeeld wordt een pictogram met driehoek en uitroepteken gebruikt om een fout visueel weer te geven en wordt aria-describedby gebruikt om informatie over de aard van de fout te verstrekken voor alle gebruikers.



```
<span class="fa fa-exclamation-triangle"></span>
```

```
<label for="name">Name:</label><br>
```

```
<input type="text" name="name" id="name"
```

```
  aria-invalid="true" aria-describedby="errorMsg"><br>
```

```
<span id="errorMsg" class="errorMsg">Error: Please enter a name"</span>
```

De foutfeedback moet expliciet en betekenisvol zijn.

De foutmeldingen moeten duidelijk en gepast zijn en moeten voldoende informatie meedelen over de actie die moet worden uitgevoerd om de informatie te corrigeren.

Voorbeeld van een goede boodschap: 'Het e-mailadres mag niet leeg zijn'

Voorbeeld van een slechte boodschap: 'Een gebruiker aanmaken is onmogelijk ten gevolge van een fout'

De foutfeedback moet zichtbaar zijn.

De meldingen moeten technisch aan hun veld worden gekoppeld en bruikbaar zijn voor schermlezers, maar ze moeten ook voor alle gebruikers zichtbaar zijn.

Slecht voorbeeld: De foutmelding is enkel zichtbaar voor schermlezers

Dit voorbeeld toont een rood omkaderd veld om een fout aan de duiden. Naast het feit dat niet alle gebruikers kleuren kunnen onderscheiden, is de aard van de fout slechts zichtbaar voor schermlezers via het aria-label attribuut.



```
<label for="date">Date: </label>
```

```
<input type="text" name="date" id="date" aria-invalid="true"
  aria-label="Error: Please enter the date as MM/YYYY">
```

c) Bevestiging van de geslaagde verzending van het formulier

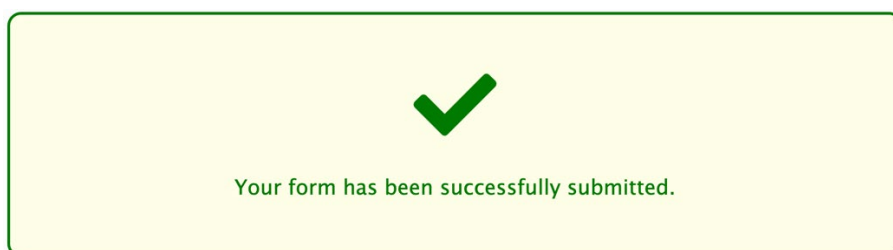
Bevestigingsberichten voor de verzending van een formulier moeten technisch bepaalbaar zijn.

Zoals voor de andere meldingen in de andere hoofdstukken moet het bevestigingsbericht technisch bepaalbaar zijn om door schermlezers te kunnen worden begrepen.

Goed voorbeeld: Technisch bepaalbaar bevestigingsbericht voor de verzending

Het voorbeeld toont een groen validatiepictogram, gekoppeld aan een aria-label dat de status van de verzending vermeldt.

Het pictogram wordt versterkt door een tekst in een paragraaf die voor iedereen zichtbaar is.



```
<div class="success">
```

```
<p><span class="fa fa-check fa-4x" aria-label="Success"></span></p>
```

```
<p>Your form has been successfully submitted.</p>
```

```
</div>
```

Bevestigingsberichten voor de verzending van een formulier moeten expliciet en betekenisvol zijn.

De bevestigingsberichten voor de verzending moeten duidelijk, gepast en informatief zijn om aan de gebruiker te bevestigen dat zijn verzoek goed wordt opgevolgd.

De volgende informatie moet bijvoorbeeld worden meegedeeld:

- Werd het formulier in goede orde verzonden?
- Werden de gegevens gewist?
- Werden de voorkeuren opgeslagen? ...

Bevestigingsberichten voor de verzending van een formulier moeten zichtbaar zijn.

Alle gebruikers, zowel zienden als slechtzienden, moeten toegang krijgen tot de berichten die de verzending bevestigen.

Slecht voorbeeld: Bevestigingsbericht via aria-live in een verborgen container

Een aria-live-regio kan worden gebruikt om in realtime aan schermlezers aan te kondigen dat de voorkeuren werden opgeslagen. De gebruiker krijgt die informatie te horen.

In dit voorbeeld ligt de aria-live-regio helaas vervat in een verborgen container via een klasse 'visually-hidden'. Daardoor krijgen zienden geen toegang tot het bevestigingsbericht.

<div id="aria-live" aria-live="assertive" class="visually-hidden">

Your preferences have been saved.</div>

P. Semantiek en structuur

<Title> van de pagina

De paginatitel (title) is bijzonder belangrijk en kritiek. Hij bevat de eerste informatie die door de schermlezer wordt gelezen, hij maakt het mogelijk om de pagina visueel te identificeren als er in de browser meerdere tabbladen open staan en hij maakt het mogelijk om gemakkelijk een opgeslagen of geraadpleegde pagina terug te vinden in de bladwijzers en de historie van de browser.

Naast de specifieke behoeften van blinden en mensen met een gebrekkig geheugen of aandachtsproblemen, is een correct gebruikte title voor iedereen voordelig en laat hij toe de SEO van de pagina te verbeteren.

a) Een <title> voor alle pagina's

De <title> van de pagina moet bestaan en tekst bevatten

Goed voorbeeld: een geldige pagina-<title>

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Products and Services | My website</title>
</head>
```

Slechte voorbeelden: ontbrekende of lege title

```
<!DOCTYPE html>
<html lang="en">
<head>
</head>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
<title></title>
</head>
```

De <title> van de pagina moet na elke adreswijziging worden bijgewerkt.

De adreswijziging van de pagina ten gevolge van het laden van een nieuwe pagina, een AJAX-oproep of een JavaScript-event moet leiden tot het bijwerken van de <title> van de pagina om die consistent te maken ten opzichte van de getoonde inhoud.

b) Een betekenisvolle pagina-<title>

De <title> van de pagina moet nauwkeurig en informatief zijn

De title moet duidelijk de bedoeling en inhoud van de pagina identificeren om te vermijden dat de gebruikers zich van pagina vergissen of naar informatie zoeken die zich niet in de inhoud van de pagina bevindt.

Inhoud van het type 'Untitled Document' of 'Document Content' is dus niet toegelaten.

Als de pagina het resultaat is van een handeling van de gebruiker of van een geprogrammeerde contextwijziging, moet de inhoud van de <title> het resultaat van de contextwijziging aan de gebruiker meedelen.

Goed voorbeeld: Een titel die het zoekwoord in een zoekmachine bevat

<title>"Warranty" - Search Results</title>

Slecht voorbeeld: Een titel die het zoekwoord in een zoekmachine niet bevat

<title>Search Results</title>

De <title> moet beknopt zijn

De volledige inhoud van de titel wordt door de schermlezer voorgelezen en weergegeven in het tabblad van de browser, de bladwijzers en de historiek. De tag bevat geen maximumlengte maar het wordt aanbevolen om zo kort mogelijk te blijven zonder aan betekenis in te boeten. 80 lettertekens lijkt een aanvaardbaar maximum te zijn.

Goed voorbeeld: Een beknopte titel

<title>Accessibility Services | Accessia</title>

Slecht voorbeeld: Misbruik van sleutelwoorden i.p.v. een goede titel

<title>Mortgages Inc. - Mortgage services, best mortgages in New York, low rates, low API, easy financing, variable rates, fixed rate mortgages</title>

De <title> moet uniek zijn, beginnen met de voornaamste informatie van de pagina die uniek is op de website

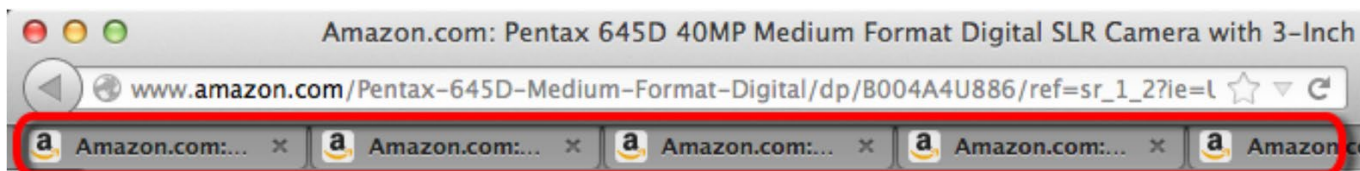
Om verwarring te vermijden, het voorlezen van de pagina te versnellen, de pagina vlotter terug te vinden in de browsers en de SEO te verbeteren, moet de paginatitel uniek zijn (geen twee identieke titles op dezelfde website) en beginnen met de belangrijkste informatie op de pagina die uniek is op de website.

Een goede praktijk bestaat erin te beginnen met de hoofdtitel van de pagina <h1> (of een sterk vergelijkbare inhoud) in de <main> van de pagina en af te sluiten met de naam van de site. Om de zichtbaarheid te verbeteren kunnen die twee delen door een scheidingsteken worden gescheiden.

We geven dus de voorkeur aan een title van het volgende type: <title>Accessibility Services | Accessia</title> in plaats van <title>Accessia | Accessibility Services</title>

Slecht voorbeeld: De paginatitels op de website Amazon.com beginnen met de naam van de website

Op de website Amazon.com begint elke paginatitel met Amazon.com en is het daardoor niet mogelijk om de naam van de open pagina's in de browser te zien. De schermlezer zal eerst 'Amazon.com ...' lezen en pas daarna de naam van de pagina.



De taal van de pagina

De meeste schermlezers kunnen meerdere talen lezen. Tijdens de installatie kiest de gebruiker een standaardtaal die door de schermlezer wordt gebruikt als de taal van de pagina niet werd gepreciseerd.

Een foute taalkeuze kan leiden tot een slecht begrip van de pagina door een ongeschikt accent of tot misverstanden over bepaalde woorden die in verschillende talen worden gebruikt, maar niet dezelfde betekenis hebben.

Een correcte taalkeuze helpt bovendien vertalers om meteen de juiste taal te identificeren en draagt bij tot een betere SEO.

a) De hoofdtaal van de pagina

De hoofdtaal van de pagina moet worden geïdentificeerd door een geldige waarde in de <html>-tag.

Goed voorbeeld van de implementering van de taal van de pagina

```
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>I Have a Dream, Excerpt</title>
</head>
<body>
<p>English text...</p>
</body>
</html>
```

b) De taal van een deel van de pagina

De tekst op een pagina die in een andere taal dan de hoofdtaal is geschreven, moet door een geldig lang-attribuut worden geïdentificeerd.

Het lang-attribuut kan van toepassing zijn op een element van het type block (<div>, <h1>, <p>, <table> ...) of inline (, <a>, ...).

Goed voorbeeld van de implementering van een taalwijziging op de pagina

```
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
```

```
<title>I Have a Spanish Dream</title>
```

```
</head>
```

```
<body>
```

```
<p>English text...</p>
```

```
<p>While in Spain, my friend tried to speak Spanish, but she wasn't very good. Everyone kept saying
```

```
<span lang="es">No comprendo nada de lo que dices.</span></p>
```

```
</body>
```

```
</html>
```

c) De taalcodes moeten geldig zijn

Taalcodes zijn gestandaardiseerd en moeten correct worden gebruikt. De landcode (country code) mag worden toegevoegd om de realiteit van een land, zoals specifieke accenten, getallen ('nonante' of 'quatre-vingt-dix') beter te benaderen. Schermlezers passen zich vandaag echter niet allemaal aan de landcode aan.

Goed voorbeeld van de implementering van de taal 'Frans uit België'

```
<html lang="fr-BE">
```

Regions (landmarks)

HTML5 biedt de mogelijkheid om de grote regio's (regions) van een webpagina, zoals <header>, <nav>, <main> en <footer>, te identificeren. ARIA maakt het ook mogelijk om regio's te identificeren, via de volgende respectievelijke attributen: role="banner", "navigation", "main" en "contentinfo".

Schermlezers bieden de mogelijkheid om van de ene regio naar de andere te navigeren. Jammer genoeg bieden browsers vandaag nog geen optie aan die alle (ziende) gebruikers in staat stelt om op die manier te navigeren. Skip-links om inhoud over te slaan blijven dus nog steeds bijzonder nuttig voor mensen die de muis niet kunnen gebruiken.

a) Landmarks en/of Aria-rollen moeten worden gebruikt om de grote secties van een webpagina aan te duiden

Het gebruik van regio's laat een vlottere navigatie toe via schermlezers, vergemakkelijkt het onderhoud van de opmaak via gestandaardiseerde code (CSS) en draagt bij tot een betere SEO.

In het geval van een codecorrectie kan het eenvoudiger zijn om ARIA-rollen toe te voegen aan bestaande div's in plaats van de systeemeigen landmarks te gebruiken.

Het toevoegen van een ARIA-rol aan een systeemeigen landmark maakt de code robuuster en toegankelijker. Bijvoorbeeld: <header role="banner">

b) Overeenkomst tussen HTML5 en ARIA-equivalenten

HTML5	ARIA-rollen role="..."	Opgelijst door schermlezers	Verplicht voor toegankelijkheid
<header>	banner	Ja	Ja
<nav>	navigation	Ja	Ja
n.v.t.	search	Ja	Ja indien beschikbaar
<main>	main	Ja	Ja
<footer>	contentinfo	Ja	Ja
<aside>	complementary	Ja	Nee
<section>	region	Opgelijst in de meeste gevallen als er een naam werd toegevoegd via aria-label of aria-labeledby. Als de code goed gestructureerd is aan de hand van headings (kopniveau <h1> tot <h6>) kunnen de gebruikers van de ene kop naar de andere navigeren. Het gebruik van talrijke landmarks op één pagina wordt niet aanbevolen omdat het voorlezen	Nee

		daardoor wordt afgeremd.	
<article>	article	Ja voor JAWS	Nee
<form>	form	Voor de meeste schermlezers enkel opgelijst met toevoeging van role="form"	Ja indien beschikbaar

c) Goede praktijken voor landmarks

Alle teksten moeten in een regio vallen en die regio moet geschikt zijn voor de inhoud
 Goed voorbeeld: <header><div>This is the header.</div></header>

Slecht voorbeeld: <div>This is the header.</div>

De instanties van hetzelfde type landmark moeten technisch worden onderscheiden door een label (aria-label of aria-labeledby)

Het is de bedoeling om de gebruikers van schermlezers zo goed mogelijk te informeren over de inhoud van de regio waarin ze zich bevinden.

Goed voorbeeld: Meerdere navigatiemogelijkheden met correcte labels

```
<nav aria-label="Main Menu">
  <ul>
    <li><a href="index.html">Home</a></li>
    <li><a href="products.html">Products</a></li>
    <li><a href="services.html">Services</a></li>
  </ul>
</nav>
<main>
  <div>[...other content...]</div>
  <nav aria-label="Products list">
    <ul>
      <li><a href="1.html">Product 1</a></li>
      <li><a href="2.html">Product 2</a></li>
      <li><a href="3.html">Product 3</a></li>
      <li><a href="4.html">Product 4</a></li>
    </ul>
  </nav>
</div>
<div>[...other content...]</div>
```

```

</main>
<footer>
  <nav aria-label="Corporate and legal info">
    <ul>
      <li><a href="about.html">About us</a></li>
      <li><a href="copyright.html">Copyright notice</a></li>
      <li><a href="terms.html">Terms of use</a></li>
      <li><a href="contact.html">Contact us</a></li>
    </ul>
  </nav>
</footer>

```

Een pagina mag slechts één instantie van het type banner, main en contentinfo bevatten

De ARIA-specificaties preciseren dat de rollen role="banner", "main" en "contentinfo" slechts één maal per pagina mogen worden gebruikt. Hetzelfde geldt voor de HTML-tags <header>, <main> en <footer>.

Het aantal landmarks per pagina moet tot de essentie beperkt blijven om het navigeren via schermlezers te vergemakkelijken.

Er is geen maximumgrens voor het aantal landmarks, maar hun vermenigvuldiging belemmert vlot voorlezen. Er wordt aanbevolen om zich te beperken tot de essentiële voor hun inhoud geschikte landmarks.

Kopniveaus (headings)

Kopniveaus (h1 tot h6) spelen een essentiële rol in de toegankelijkheid van een website of een document (Word, pdf ...). Ze delen aan elke gebruiker een duidelijke paginastuur mee. Het skelet dat zo wordt aangemaakt stelt de gebruikers van schermlezers in staat om een kort overzicht van de inhoud van de pagina te zien en naar de gewenste kop te navigeren.

Een goede koppenstructuur verbetert bovendien de SEO in zoekmachines en vergemakkelijkt het onderhoud en de aanpassing van de opmaak van een website. In word-documenten laten ze ook toe om automatisch een inhoudstafel aan te maken.

a) Echte kopniveaus

Teksten die dienst doen als visuele of structurele koppen moeten de juiste kop-tags gebruiken.

Koppen worden gebruikt om de gebruiker te helpen bij zijn navigatie op de pagina. De bedoeling is semantisch en niet visueel. Een schermlezer kan geen CSS lezen en kan niet afleiden of een grotere of vettere tekst als een kop moet worden beschouwd.

Koppen worden dus afgebakend door de tags <h1> tot <h6> en de niveaus van <hx> worden niet gekozen in functie van de layout, maar volgens de hiërarchie van de koppen. CSS kan vervolgens de lay-out van elk kopniveau aanpassen.

Goed voorbeeld: Echte koppen met correcte tags

```
<h1>Types of Web Accessibility Laws</h1>
```

<h2>Civil Rights Laws</h2>

<h2>Procurement Laws</h2>

<h2>Industry-Specific Laws</h2>

Goed voorbeeld: Echte koppen met ARIA-tags

In bepaalde gevallen van 'toegankelijkheidsherstel' van een website, kan ARIA worden gebruikt om van een component een kopniveau te maken. Daarvoor werd de role="heading" in combinatie met aria-level ontworpen.

Opgelet, het gebruik van de systeemeigen tags <h1> tot <h6> blijft de voorkeuroplossing en ARIA is een back-upoplossing (ook al is die robuust).

```
<div role="heading" aria-level="1">
```

Screen readers will recognize this as a heading level 1

```
</div>
```

Slecht voorbeeld: Valse koppen met gebruik van het stijlblad

De onderstaande paragrafen <p style ...> lijken visueel op koppen, maar vervullen die functie niet.

We merken bovendien op dat gebruikers die een andere lettergrootte, kleur, lettertype ... nodig hebben in functie van hun profiel die niet vlot kunnen aanpassen als de stijl, zoals in dit voorbeeld, in de HTML-code werd geïntegreerd.

```
<div id="noHeadingExample">
```

```
<p style="font-weight: bold; font-size: 200%;">
```

Setting the Exposure Manually on a Camera

```
</p>
```

```
<p>Put text here...</p>
```

```
<p style="font-weight: bold; font-size: 150%;">
```

Set the ISO

```
</p>
```

```
<p>Put text here...</p>
```

```
<p style="font-weight: bold; font-size: 150%;">
```

Choose an aperture

```
</p>
```

```
<p>Put text here...</p>
```

```
<p style="font-weight: bold; font-size: 150%;">
```

Choose a shutter speed

```
</p>
```

```
<p>Put text here...</p>
```

</div>

Teksten die niet bedoeld zijn om als kop te worden gebruikt, mogen niet door de koppen <hx> worden afgebakend.

Teksten die een speciale opmaak vereisen zonder een kopfunctie te vervullen, moeten in CSS worden ontworpen zonder gebruik te maken van koptags. Die praktijk maakt het voorlezen verwarrend voor gebruikers van schermlezers en kan fouten of misverstanden veroorzaken.

Slecht voorbeeld: Een promotionele blikvanger afgebakend zoals een kop

Special rates good until July 31!

<h1 class="promotional-item">Special rates good until July 31!</h1>

b) *Expliciete, betekenisvolle koppen*

Koppen moeten gepast en informatief zijn

Koppen moeten voldoende duidelijk zijn om de gebruiker in staat te stellen om de betekenis van de tekst die ze inleiden te begrijpen. De schermlezer leest de gebruiker het kopniveau, gevolgd door de inhoud van <hx> voor.

Het aantal tekens wordt niet beperkt tot een bepaalde limiet, maar er wordt aanbevolen om zo kort en expliciet mogelijk te blijven. Het is de bedoeling om de inhoud die op de kop volgt kort in te leiden.

Goed voorbeeld: Expliciete koppen

<h1>Emergency Preparedness Guide</h1>

<h2>Know the Risks</h2>

<h2>Make a Plan</h2>

<h2>Get an Emergency Kit</h2>

<h3>Emergency Kit Basic Items</h3>

<h2>Resources</h2>

Slecht voorbeeld: Algemene koppen

<h1>Emergency Preparedness Guide</h1>

<h2>Section 1</h2>

<h2>Section 2</h2>

<h2>Section 3</h2>

<h3>(a)</h3>

<h2>Section 4</h2>

c) *Hiërarchie en structuur van de inhoud*

De kopniveaus moeten een samenhangende structuur van de organisatie van de paginasecties presenteren

Denk altijd in de eerste plaats aan de paginastructuur en zorg pas daarna voor de layout. De inhoud krijgt voorrang en CSS wordt pas later verzorgd. Die aanpak is trouwens ok voordelig

voor de SEO en maakt het mogelijk om de layout op smartphones gemakkelijker te maken (inhoud verbergen of schikken in functie van de behoeften).

Als een paginastructuur enkel koppen met weinig tekst bevat, moet misschien worden overwogen om de structuur door een lijst met opsommingstekens (bulletlijst) te vervangen. Als de inhoud echter één of meer verklarende paragrafen bevat, moet een structuur en een hiërarchie worden overwogen om de gebruiker naar de inhoud die hem interesseert te leiden en hem snel de ‘inhoudstafel’ van de pagina te tonen.

[Goed voorbeeld: Een logische kopstructuur om de belichting van een fotoapparaat aan te passen](#)

Een lijst stappen zonder inhoud kan in de vorm van een bulletlijst worden opgesteld.

```
<ul>
  <li> Setting the Exposure Manually on a Camera
    <ul>
      <li>Set the ISO</li>
      <li>Choose an aperture</li>
      <li>Set a shutter speed</li>
    </ul>
  </li>
  <li>...</li>
</ul>
```

Maar als elke stap meer uitleg vereist, is een koppenstructuur aangewezen.

```
<h1>Setting the Exposure Manually on a Camera</h1>
  <p>Put text here...</p>
  <h2>Set the ISO</h2>
  <p>Put text here...</p>
  <h2>Choose an aperture</h2>
  <p>Put text here...</p>
  <h2>Choose a shutter speed</h2>
  <p>Put text here...</p>
```

Begin de inhoud van elke pagina met een <h1> en beperk uzelf tot één <h1> per pagina

Schermlezers bevatten snelkoppelingen waarmee de gebruiker rechtstreeks naar de gewenste kop kan navigeren. Als de pagina niet met een <h1> begint, kan de gebruiker verward raken. Hetzelfde geldt voor een pagina die meerdere <h1>-tags zou bevatten.

Daarom moet elke pagina-inhoud met een <h1> beginnen. Afgezien van uitzonderingen (interactieve modale vensters, lijst met artikels op de pagina ...) mag er op de pagina slechts

één <h1>-tag worden gebruikt. Idealiter moet die <h1> tekst bevatten met een betekenis die nauw aansluit bij de <title> van de pagina.

Die praktijk maakt ook een betere positionering in zoekmachines mogelijk en versnelt eventuele esthetische aanpassingen.

d) De hiërarchie van de koppen mag niet worden verbroken

De hiërarchie van de koppen moet worden nageleefd. Zo mogen we bijvoorbeeld niet van een <h3>-niveau naar een <h5> springen zonder tussen die twee een <h4> te voorzien. Die fout wordt vaak gemaakt als een kopniveau wordt gekozen omwille van de layout en niet omwille van de semantiek. Zoals hierboven reeds werd besproken, komt de inhoud op de eerste plaats en wordt de layout pas daarna via de CSS bepaald.

De hiërarchie van de inhoud kunnen we met een stamboom vergelijken:

Helemaal bovenaan, de overgrootmoeder <h1>

... die kinderen heeft (grootouders) <h2>

... die kinderen hebben (ouders) <h3>

... die kinderen hebben (kinderen) <h4>

... die kinderen hebben (kleinkinderen) <h5>

... die kinderen hebben (achterkleinkinderen) <h6>.

In die boom kan een ouder (h3) niet op hetzelfde niveau staan als een kleinkind (h5) en het vergeten van een niveau <hx> heeft hetzelfde gevolg als het overstappen van het niveau van de overgrootmoeder (h1) op dat van de ouders (h3), wat onmogelijk is omdat de grootouders de ouders van de ouders zijn ...

Goed voorbeeld: Gedetailleerde stappen van de belichtingsinstellingen van een fototoestel

```
<div class="exampleHeadings">
```

```
<h1>Setting the Exposure Manually on a Camera</h1>
```

```
<h2>Setting the ISO</h2>
```

```
<h3>The effect of ISO on image quality</h3>
```

```
<p>The higher the ISO, the greater your ability to take photos in low light [...]</p>
```

```
<h3>High ISO cameras</h3>
```

```
<p>Cameras with larger sensors are capable of higher ISO values [...]</p>
```

```
<h2>Choose an aperture</h2>
```

```
<h3>The effect of aperture on depth of focus</h3>
```

```
<p>The larger the aperture, the narrower the plane of focus [...]</p>
```

```
<h3>Vignetting</h3>
```

```
<p>Vignetting occurs when a lens lets in less light around the edges than in the center [...]</p>
```

```
<h3>Diffraction</h3>
```

<p>A small aperture can cause the light to diffract, which reduces the sharpness of the image, even though more of the image is in focus [...]</p>

<h2>Choose a shutter speed</h2>

<h3>Shutter speed limitations for hand-holding a camera</h3>

<p>Hand-holding a camera at a slow shutter speed increases the chance of vibration and blur in the final image [...]</p>

<h3>Long exposures</h3>

<p>Use a tripod when taking long exposures [...]</p>

</div>

Slecht voorbeeld: Een plan zonder structuur

Het onderstaande voorbeeld heeft twee problemen:

1. De pagina begint met twee <h3>-koppen i.p.v. één <h1> en geen van die koppen kan worden beschouwd als de hoofdtitel van de pagina.
2. Verderop bevat de inhoud een <h1> die meteen naar een <h3> overspringt.

<h3>Quick Links</h3>

<h3>Photography Tutorials</h3>

<h1>Setting the Exposure Manually on a Camera</h1>

<h3>Set the ISO</h3>

<h4>The effect of ISO on image quality</h4>

<h4>High ISO cameras</h4>

<h3>Choose an aperture</h3>

<h4>The effect of aperture on depth of focus</h4>

<h4>Vignetting</h4>

<h4>Diffraction</h4>

<h3>Choose a shutter speed</h3>

<h4>Shutter speed limitations for hand-holding a camera</h4>

<h4>Long exposures</h4>

Hyperlinks

Schermlezers herkennen hyperlinks en kondigen die als dusdanig aan. Het is dus niet nodig om daar bijvoorbeeld de woorden 'link naar ...' aan toe te voegen.

a) Hyperlinks correct aanduiden

Hyperlinks moeten semantisch als hyperlinks worden aangeduid

Hyperlinks moeten zich binnen <a href...>-tags bevinden of via ARIA als hyperlinks worden aangeduid.

Goed voorbeeld: Hyperlink in een <a>-tag die een geldige waarde bevat

```
<a href="https://eqla.be">Eqla</a>
```

Goed voorbeeld: Hyperlink als dusdanig aangeduid met ARIA en die een geldige waarde bevat

Het gebruik van de tag <a> verdient de voorkeur voor het aanmaken van hyperlinks, maar in sommige gevallen (herstelling van websites, correctie van toegankelijkheidsgebreken ...) is het mogelijk om het element dat de hyperlink bevat via ARIA te veranderen in een element van het type hyperlink.

In dat geval gelden de drie volgende voorwaarden:

- Role="link" toevoegen aan het element;
- Het attribuut tabindex="0" toevoegen aan het element zodat het de focus krijgt bij toetsenbordnavigatie;
- Ervoor zorgen dat het gekoppelde script in JavaScript het mogelijk maakt om de link zowel via de muis als via het toetsenbord te activeren.

```
<span role="link" tabindex="0" data-href="https://eqla.be">Eqla</span>
```

Slecht voorbeeld: Een valse link in een spambericht in JavaScript

De onderstaande code toont een hyperlink naar de website van Eqla op de pagina. De tekst lijkt op een hyperlink en werkt met de muis, maar wordt niet beschouwd als een link door hulptechnologieën en krijgt geen focus.

```
<script>
```

```
function openInNewTab(url) {
  var win = window.open(url, '_blank');
  win.focus();
}
```

```
</script>
```

```
<span class="fakeLink" onclick="openInNewTab('https://www.eqla.be');">Eqla</span>
```

Links en knoppen moeten semantisch worden gebruikt voor de functie waarvoor ze werden ontworpen

Schermlezers preciseren voor hun gebruikers op welk type van element ze zich bevinden. Ze beginnen met te zeggen 'link' of 'knop' zodra ze de focus geven.

Een link gebruiken voor een knop of omgekeerd kan misverstanden of verwerkingsfouten veroorzaken voor de betrokken gebruikers. Als de internetgebruiker bijvoorbeeld specifiek de

lijst van de hyperlinks op de pagina opvraagt, zullen de links die als knoppen worden beschouwd daar niet in worden opgenomen.

De HTML-documentatie vermeldt duidelijk de gevallen van gebruik, die we hier in herinnering willen brengen:

1. Hyperlinks verwijzen naar een andere pagina of naar een specifieke locatie op dezelfde pagina.
2. Knoppen activeren een functie, in het algemeen op dezelfde pagina (een dialoogvenster openen, een accordion-inhoud uitrollen ...), of verzenden gegevens op het einde van een formulier.

b) Tekst in hyperlinks

Een hyperlink moet technisch een tekst bevatten die als toegankelijke naam (accessible name) wordt beschouwd

Een hyperlink moet worden gekoppeld aan een tekst die dienstdoet als toegankelijke naam berekend door de accessibility tree bij de algoritmische berekening. Als hij op de hyperlink terechtkomt, leest de schermlezer 'link', gevolgd door de toegankelijke naam. Als die naam niet bestaat of niet correct werd geïmplementeerd, leest de schermlezer de inhoud van de href-tag om te proberen om de gebruiker informatie te verschaffen. Die niet-expliciete inhoud zal vaak te lang of onbegrijpelijk blijken en kan misleidend zijn.

De toegankelijke naam wordt door schermlezers in de volgende voorkeurvolygde berekend:

1. Aria-labelledby,
2. Aria-label,
3. Tekst tussen de openingstag <a> en de afsluiter met inbegrip van het alt-attribuut van de afbeeldingen,
4. Het title-attribuut. Dat geval mag pas in allerlaatste instantie worden overwogen omdat het tot andere toegankelijkheidsproblemen kan leiden.

Ook al verschijnt de tekst tussen <a>-tags pas in derde positie, hij zou als eerste methode moeten worden gebruikt omdat ze gemakkelijk uitvoerbaar is, systeemeigen in HTML werd voorzien, geen fouten veroorzaakt bij updates van de website door een webmaster die niet in deze techniek werd getraind en die de berekeningstechniek niet kent zonder te beseffen dat zijn linktekst niet correct wordt begrepen.

Goed voorbeeld: Link in de vorm van tekst tussen <a>-tags

```
<p><a href="https://eqla.be/contact/">Contact Us</a></p>
```

```
<p><a href="https://eqla.be/accueil/notre-equipe/">Our team</a></p>
```

Goed voorbeeld: Link in de vorm van tekst in het alt-attribuut van een hyperlinkafbeelding

```
<p>
```

```
<a href="https://eqla.be/">
```

```

```

```
</a>
```

```
</p>
```

Goed voorbeeld: aria-label om de oorspronkelijke tekst van de link te overschrijven

Dit voorbeeld toont hoe aria-label een algemene hyperlinktekst zoals 'Read more ...', gebruikt als 'teaser' in een pagina met een lijst van publicaties, expliciet kan maken.

Aria-label vervangt de oorspronkelijke tekst 'Read more' door 'Read more + titel van de betreffende publicatie'.

<p>

The National Museum of African American History and Culture was established in 2003 by an Act of Congress, making it the 19th Smithsonian Institution museum.

```
<a href="https://www.si.edu/Museums/african-american-history-and-culture-museum"
  aria-label="Read more about the National Museum of African American History and Culture">
```

Read more...

```
</a>
```

</p>

<p>

The National Air and Space Museum, Steven F. Udvar-Hazy Center displays hundreds of aviation and space artifacts that are too large to exhibit in the museum on the National Mall in Washington, D.C.

```
<a href="https://www.si.edu/Museums/air-and-space-museum-udvar-hazy-center"
  aria-label="Read more about the Steven F. Udvar-Hazy Center">
```

Read more...

```
</a>
```

</p>

Goed voorbeeld: Aanvullende tekst in een link die verborgen is op het scherm, maar leesbaar is voor schermlezers

In sommige gevallen kan het nuttig zijn om enkel aan de gebruikers van schermlezers bijkomende informatie te verstrekken via in CSS verborgen tekst aan de hand van de clip-methode.

```
<head>
```

```
<title>Museum Information</title>
```

```
<style>
```

```
.visually-hidden {
```

```
  position: absolute;
```

```
  clip: rect(0 0 0 0);
```

```

border: 0;
height: 1px; margin: -1px;
overflow: hidden;
padding: 0
width: 1px;
white-space: nowrap;
}
</style>
</head>
<body>
<p>
The National Museum of American History is devoted to the scientific,
cultural, social, technological, and political development of the United States.
<a href="https://www.si.edu/Museums/american-history-museum">
Read more
<span class="visually-hidden">
about the National Museum of American History
</span>...
</a>
</p>
</body>

```

[Goed voorbeeld: Aria-label gebruiken om aan achtergrondafbeeldingen een hyperlinktekst te geven](#)

Achtergrondafbeeldingen gekoppeld aan een hyperlink worden vaak gebruikt om een link naar sociale netwerken te leggen. Die techniek werkt voor ziende mensen, maar laat niet toe om een toegankelijke naam voor de hypelink te voorzien.

In dit voorbeeld levert aria-label de link een tekst die schermlezers aan gebruikers kunnen meedelen.

```

<a href="https://www.facebook.com/Eqla.asbl/" class="facebook"
aria-label="Facebookpagina van Eqla"></a>

```

De bedoeling van elke link moet meteen expliciet zijn dankzij de tekst van de link

De context van de hyperlink bestaat uit de onmiddellijke omgeving van de link: een kop, een paragraaf ...

Gebruikers van schermlezers kunnen de inhoud van een pagina ontdekken door de hele inhoud van boven naar beneden te lezen of door van de ene titel of de ene link naar de andere te springen ... Om van de ene link naar de andere te navigeren vereist een link zoals 'lees

meer', zelfs als de context van de link expliciet is, meer handelingen van de gebruiker om de bedoeling van de link te begrijpen.

De eenvoudigste en meest robuuste techniek bestaat er dan ook in de link los van zijn context expliciet te maken door de bedoeling rechtstreeks in de linktekst te vermelden.

We hebben het hier meer bepaald over algemene, korte hyperlinks zoals 'Klik hier', 'Hier', 'Lees meer', 'Meer' ...

Expliciete links hebben bovendien nog andere voordelen:

- Een betekenisvolle link draagt bij tot een betere positie in zoekresultaten (SEO);
- Een groter klikoppervlak is gemakkelijker bereikbaar voor slechtzienden of mensen met een beperkte capaciteit om kleine doelen te raken;
- Een groter klikoppervlak nodigt de gebruiker uit om 'tot actie over te gaan' en op de website meer pagina's te raadplegen.

Goed voorbeeld: Links die ook los van hun context iets betekenen

<p>Learn more about our products.</p>

<p>Read a fascinating article about the resident microbes in the human body.</p>

<p>Our second quarter earnings have surpassed investor expectations.</p>

Slecht voorbeeld: Links met een tekst die niet beschrijvend is

<p>Learn more about our products here.</p>

<p>To read a fascinating article about the resident microbes in the human body click here. </p>

<p>Our second quarter earnings have surpassed investor expectations. More...</p>

Functies zoals labels, namen en tekstalternatieven met dezelfde functie moeten op dezelfde manier worden geïdentificeerd

Het is de bedoeling om dubbelzinnigheid te vermijden en de samenhang van de hele website te verzekeren.

Slecht voorbeeld: Links naar dezelfde pagina die niet door dezelfde tekst worden beschreven

<p>Our Company</p>

<p>Contact Us</p>

Slecht voorbeeld: Links naar verschillende pagina's die door dezelfde tekst worden beschreven

<p>Contact Us</p>

<p>Contact Us</p>

c) Links naar andere websites, nieuwe vensters en bestanden

Een link die een nieuw venster of tabblad opent, moet die opening melden

Alle gebruikers moeten erop worden gewezen dat de link in een nieuw venster wordt geopend, ofwel door een tekst zoals 'nieuw venster' of door een visuele aanwijzing die door een

tekstalternatief wordt versterkt (alt). Die aanwijzing moet zowel voor zienden als voor blinden zichtbaar en begrijpelijk zijn.

Slecht voorbeeld: Het openen in een nieuw venster wordt niet gemeld

```
<p>
  <a href="https://eqla.be" target="_blank">Eqla</a>
</p>
```

Goed voorbeeld: Een afbeelding met tekstalternatief om het openen in een nieuw venster te melden

Ziende gebruikers zien de afbeelding en blinden horen het tekstalternatief.

```
<p>
  <a href="https://eqla.be" target="_blank">Eqla
    
  </a>
</p>
```

Goed voorbeeld: Gebruik van CSS en ARIA om het openen van een nieuw venster te melden

Ook hier zien ziende gebruikers de afbeelding en horen blinden het tekstalternatief.

```
<p>
  <a
    aria-describedby="a11y-message--new-window"
    class="icon--new-window"
    href="https://eqla.be" target="_blank">Eqla</a>
</p>
```

```
.
.
```

```
<span aria-hidden="true" class="visually-hidden" id="a11y-message--new-window"> (opens new window)</span>
```

De klasse 'icon--new-window' wordt gebruikt om het pictogram via CSS (glyph icon) na de tekst van de link in te voegen.

De die de id="a11y-message--new-window" bevat levert de tekst 'opens a new window' die één maal in de pagina wordt opgenomen en die op het scherm wordt verborgen via CSS met de [clip-methode](#) en voor schermlezers wordt verborgen via aria-hidden="true".

Goed om te weten: aria-describedby en aria-labelledby hebben toegang tot de inhoud van containers die worden verborgen door aria-hidden="true". In ons voorbeeld houdt dat in dat de boodschap 'opens a new window' wordt gelezen als de link de focus krijgt via het toetsenbord, maar de gebruikers van schermlezers het bericht niet te horen krijgen als ze de inhoud van de pagina in leesmodus (zonder tabtoets) lezen.

Het voordeel van deze techniek, ten opzichte van het vorige voorbeeld (afbeelding + alt), is dat de afbeelding slechts één keer in de pagina wordt geladen. Het nadeel is dat NVDA en JAWS in leesmodus zonder tabtoets met focus op de link het bericht 'opens a new window' niet voorlezen.

Een link naar een bestand, een toepassing of een ander formaat dan het web moet het type, het gewicht en de taal van het bestand of de toepassing vermelden.

Alle gebruikers moeten op voorhand weten welk type bestand of toepassing wordt geopend door op de link te klikken. Die informatie kan worden gemeld via tekst (bijv. pdf, xlsx, docx ...) of via een afbeelding met een tekstalternatief zoals hierboven wordt beschreven [bij het openen van een nieuw venster](#).

Om onaangename verrassingen te voorkomen moet de link het gewicht van het document (bijv. 2 Mo) bevatten en ook de taal van het document als dat in een andere taal dan die van de pagina werd opgesteld (bijv. EN voor een document in het Engels op een Franstalige website).

Die goede praktijken stellen alle gebruikers in staat om tijd te winnen door enkel de documenten te downloaden die op hun apparaat kunnen worden gebruikt (geen .xlsx-lezer op mijn smartphone), zonder hun data-abonnement te overschrijden (een bestand van 2 Go ...) en in een taal die ze kennen.

Goed voorbeeld: Download ons activiteitenverslag

`Download het activiteitenverslag 2022 van Eqla (PDF, 2MO, EN)`

d) Hyperlinks visualiseren

De links moeten visueel van hun omringende context kunnen worden onderscheiden

In standaard HTML verschillen links door hun kleur en onderstreping duidelijk van de omringende tekst, maar dat standaardgedrag wordt logischerwijze aangepast om met de huisstijl van de website overeen te stemmen.

Het gebruik van kleur alleen volstaat niet om een link van de rest van de tekst te onderscheiden, tenzij de contrastverhouding voldoet aan de toegankelijkheidsnormen (minstens 3:1) en een ander onderscheid wordt toegepast als de cursor over de link zweeft (hover) of als de link de focus krijgt.

Er bestaan verschillende toegankelijke manieren om hyperlinks van de omringende tekst te onderscheiden:

- Een andere kleur + onderstrepen (standaardgedrag);
- Een andere kleur + onderstrepen bij hover EN focus. Dat gedrag wordt vaak uitgeschakeld door sommige ontwerpers die deze stijl niet op prijs stellen;
- Een andere achtergrondkleur bij hover EN focus + onderstreping bij hover EN focus;
- Een outline of omkadring bij hover EN focus + onderstrepen EN/OF een andere achtergrondkleur bij hover EN focus;
- Navigatiemenu: in navigatiemenu's verwachten de gebruikers enkel links naar andere pagina's of binnen dezelfde pagina en is onderstrepen niet noodzakelijk.

e) Visuele focusindicatie

Alle elementen die de focus krijgen moeten visueel worden aangeduid zodra ze de focus krijgen

Standaard brengen browsers een gekleurde ononderbroken of stipellijnkader aan rond componenten die de focus krijgen. Een ononderbroken lijn is beter zichtbaar dan een stippellijn, die echter wel aan de minimale toegankelijkheidsnormen voldoet.

Jammer genoeg beschadigen sommige website-ontwerpen die visuele informatie, hetzij vrijwillig of door een slecht ontworpen CSS-reset.

Zo kunnen ziende gebruikers die andere navigatiemiddelen dan de muis gebruiken hun positie op de website niet terugvinden.

Slecht voorbeeld: De focusindicatie werd verwijderd

```
<style>
a:focus {
  outline: none;
}
</style>
```

Elementen die de focus krijgen moeten genieten van een verbeterde visuele indicatie

Om alle gebruikers in staat te stellen om vlotter door een website te navigeren, moet de standaard visuele focusindicatie worden versterkt.

Er bestaan verschillende middelen om de focus op links, knoppen, formulierelementen en andere componenten te versterken. Die versterking kan worden toegepast op de background, de letterkleur of de kader:

- Links – a:focus {...}
- Knoppen (buttons) – button:focus {...}
- Formulievelden (inputs) – input:focus {...} voor alle inputs, maar het is mogelijk om de inputtypes als volgt een stijl te geven:
 - text inputs — input[type=text]:focus {...}
 - image inputs — input[type=image]:focus {...}
 - submit buttons — input[type=submit]:focus {...}
 - radio buttons — input[type=radio]:focus {...}
- Selectievakjes (checkboxes) — input[type=checkbox]:focus {...}
- Keuzelijsten (drop-down) selection inputs — select:focus {...}
- Commentaarvelden (textarea fields) — textarea:focus {...}
- ARIA controls:
 - ARIA links — [role=link]:focus {...}
 - ARIA buttons — [role=button]:focus {...}
 - ARIA inputs — [role=radio], [role=checkbox], etc.
 - ARIA tabs — [role=tab]:focus {...}

Goed voorbeeld: De focusindicatie werd versterkt

```
<style>
a:focus {
  outline: 2px solid #8cc63f;
  background-color: #fdf6e7;
}
</style>
```

Navigatie tussen verschillende pagina's

Navigatiemenu's moeten worden afgebakend door de tag <nav> of door een role="navigation". Het aantal en de volgorde van de links in het menu moeten coherent zijn en op alle pagina's identiek blijven.

a) Navigatielijsten

Navigatielijsten moeten worden aangeduid door de tag <nav> of door role="navigation".

De hoofdnavigatie moet worden afgebakend door de tag <nav> of door role="navigation". Het is niet wenselijk om alle secundaire navigaties op die manier af te bakenen en om de gebruiker niet te verwarren. De verschillende navigatielijsten moeten een expliciet label krijgen (zie hoofdstuk over de regio's).

Goed voorbeeld: gebruik van de <nav>-tag

```
<nav>
  <ul>
    <li><a href="home/">Home</a></li>
    <li><a href="home/products">Products</a></li>
    <li><a href="home/services">Services</a></li>
  </ul>
</nav>
```

Goed voorbeeld: gebruik van role="navigation"

Let erop de rol toe te wijzen aan de div en niet aan de om de semantiek van de lijst met opsommingstekens niet te overschrijven en de schermlezer de mogelijkheid te bieden om te preciseren dat het om een bullelijst met x items gaat.

```
<div role="navigation">
  <ul>
    <li><a href="home/">Home</a></li>
    <li><a href="home/products">Products</a></li>
    <li><a href="home/services">Services</a></li>
  </ul>
</div>
```

Een navigatielijst moet gebruikers visueel tonen waar de huidige pagina in het menu staat

Het is de bedoeling om de actieve pagina in het menu duidelijk te tonen. Dezelfde regel geldt voor een reeks pagina's die tot dezelfde verzameling behoren, zoals bijvoorbeeld de verschillende pagina's van de resultaten van een zoekmachine. De huidige pagina moet visueel worden geïdentificeerd in de paginalijst (1, 2, 3 ...).

Goed voorbeeld: Nadruk op de huidige pagina

De huidige pagina wordt geïdentificeerd via een klasse 'current-page'

```
<head>
  <title>Groceries</title>
  <style>
    li.current-page {
```

```

outline: red solid 1px;
}
</style>
</head>
<body>
  <nav>
    <ul>
      <li><a href="#">Fruits</a></li>
      <li><a href="#">Vegetables</a></li>
      <li><a href="#">Meats</a></li>
      <li class="current-page">Dairy</li>
      <li><a href="#">Breads, Pasta, & Cereals</a></li>
      <li><a href="#">Soups & Canned Goods</a></li>
      <li><a href="#">Frozen Foods</a></li>
      <li><a href="#">Desserts</a></li>
      <li><a href="#">Snack Foods</a></li>
    </ul>
  </nav>
</body>

```

Een navigatielijst moet gebruikers van schermlezers in staat stellen om de huidige pagina te identificeren

Het attribuut `aria-current` geeft voor schermlezers het huidige element, zoals bijvoorbeeld de actieve pagina, aan. Dat attribuut wordt door de meeste browser-en-schermlezercombinaties ondersteund.

Dezelfde regel geldt voor een reeks pagina's die tot dezelfde verzameling behoren, zoals bijvoorbeeld de verschillende pagina's van de resultaten van een zoekmachine. De huidige pagina moet door de hulptechnologieën worden geïdentificeerd in de paginalijst (1, 2, 3 ...).

[Goed voorbeeld: identificatie van de huidige pagina via `aria-current`](#)

```

<ul>
  <li><a href="#">Fruits</a></li>
  <li><a href="#">Vegetables</a></li>
  <li><a href="#">Meats</a></li>
  <li><a href="#" aria-current="page">Dairy</a></li>
  <li><a href="#">Breads, Pasta, & Cereals</a></li>
  <li><a href="#">Soups & Canned Goods</a></li>

```

```
<li><a href="#">Frozen Foods</a></li>
```

```
<li><a href="#">Desserts</a></li>
```

```
<li><a href="#">Snack Foods</a></li>
```

```
</ul>
```

Coherente navigatie op alle pagina's van de website

De onderdelen van het menu en hun volgorde moeten op alle pagina's van de website identiek zijn

Gebruikers kunnen in verwarring raken wanneer het menu van de ene pagina op de andere verandert en verliezen tijd omdat ze de navigatiewijzigingen telkens opnieuw moeten begrijpen.

Navigatie binnen de pagina

De gebruikers van schermlezers begrijpen de structuur van de pagina dankzij semantische tags. Ze kunnen vlot en op verschillende manieren navigeren tussen elementen van hetzelfde type ... maar sommige functies zijn niet systeemeigen, zoals bijvoorbeeld 'u bevindt zich hier' in zoekresultaten.

Bovendien hebben ziende gebruikers zonder schermlezer die enkel het toetsenbord gebruiken geen toegang tot die functies en is het nodig om hen middelen te bezorgen om vlotter en sneller te navigeren.

a) Skip-links

Een skip-linkfunctie moet de gebruikers in staat stellen om rechtstreeks in de hoofdinhoud van elke pagina te navigeren

Voordat hij in de <main>-hoofdinhoud van de pagina terechtkomt, moet de gebruiker een aantal stappen doorlopen, zoals het logo van de hoofding, de zoekmachine, de loginwidget, de hoofdnavigatie (die soms bijzonder lang kan zijn) ... In een aantal gevallen komt daar nog een secundaire navigatie bij vóór hij de hoofdinhoud bereikt. Gebruikers moeten al die inhoud kunnen overslaan via een snelkoppeling in de vorm van een skip-link, die zich op de eerste positie van de pagina bevindt en via het toetsenbord toegankelijk is.

Let wel op enkele gebruikelijke voorzorgen:

- De skip-link is het eerste element dat op de pagina de focus kan krijgen;
- Hij moet helemaal in het begin van het document worden geplaatst, vlak na het openen van de body;
- De link verwijst naar de id die het begin van de hoofdtekst identificeert. Normaal zou dat <main> of <h1> moeten zijn;
- De skip-link moet focus geven aan het doelelement en niet enkel het scherm scrollen tot aan de inhoud die moet worden geraadpleegd.

Goed voorbeeld van een skip-link

```
<body>
```

```
<div id="skipnav"><a href="#mainContent">Skip navigation</a></div>
```

```
<!-- the header, navigation, etc. go here -->
```

```
<main id="mainContent" tabindex="-1">
```

```
<!-- the main content goes here -->
```

</main>

...

</body>

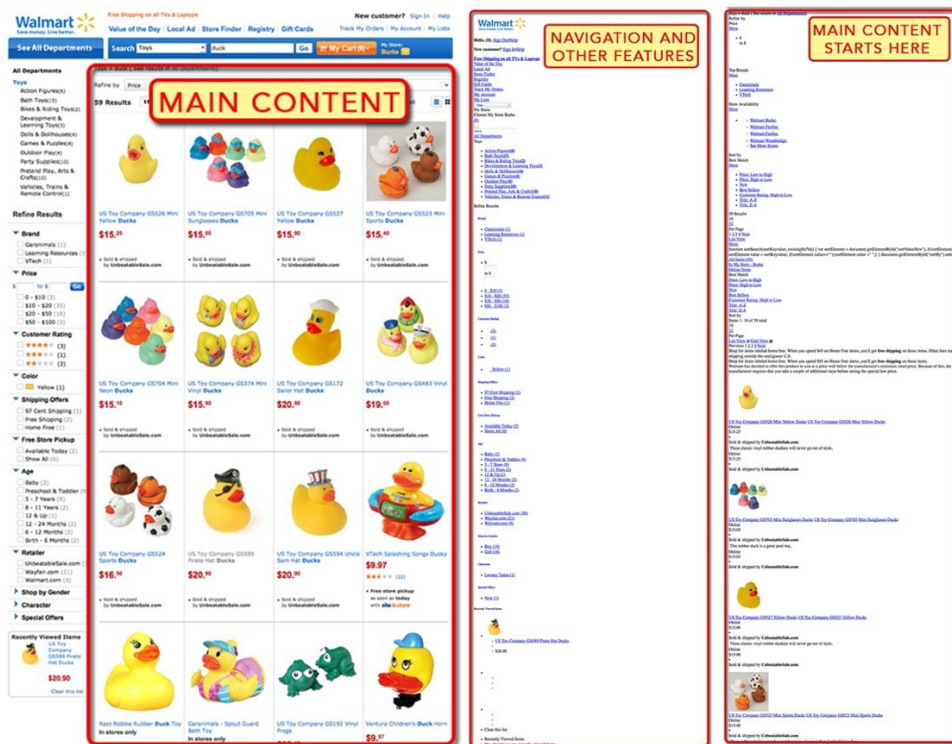
Sommige browsers, zoals Safari, vereisen een bestemming die een systeemeigen focus kan krijgen (link, knop ...) of een element dat een tabindexwaarde bevat. Zoniet scrolt de viewport naar de gewenste positie, maar plaatst de volgende tab de focus op het element dat meteen na de skip-link volgt.

Door aan de bestemming een tabindex="-1" toe te voegen, wordt de focus op dat element geplaatst zonder de eventueel op de pagina aangebrachte tabvolgorde te verstoren.

Slecht voorbeeld: Geen skip-link op een verkoopsite met talrijke stappen vóór de main

De onderstaande beelden tonen de Walmart-website, met ongeveer 76 links en 5 formulervelden voordat de hoofdinhoud wordt bereikt. Ook al worden enkele links verborgen via JavaScript, het aantal onvermijdelijke tabs bij elke geopende pagina maakt de raadpleging moeizaam en erg lang.

De screenshot waarop CSS werd uitgeschakeld toont de lineaire reeks zoals ze door een schermlezer of via de tabtoets van het toetsenbord wordt doorlopen.



Een skip-link moet permanent of bij gebruik van de tabfunctie via het toetsenbord zichtbaar zijn.

De meest toegankelijke optie voor iedereen is om de skip-link voor iedereen zichtbaar te houden, maar omwille van de huisstijl of van ontwerpvoorkeuren wordt de link soms visueel verborgen, terwijl hij toch leesbaar blijft voor schermlezers en kan worden geactiveerd via de tabfuncties van het toetsenbord.

De skip-link mag niet onzichtbaar worden gemaakt op het scherm via `display:none` omdat die parameter de link onbruikbaar zou maken voor het toetsenbord en voor schermlezers. Het gebruik van de [clip-methode in CSS](#), die in dit document al enkele keren werd besproken, wordt aanbevolen.

[Goed voorbeeld van een verborgen skip-link die zichtbaar wordt bij de focus](#)

```
<head>
```

```
<title>Museum Information</title>
```

```
<style>
```

```
#skipnav a {
```

```
  position: absolute;
```

```
  clip: rect(0 0 0 0);
```

```
  border: 0;
```

```
  height: 1px; margin: -1px;
```

```
  overflow: hidden;
```

```
  padding: 0
```

```
  width: 1px;
```

```
  white-space: nowrap;
```

```
}
```

```
#skipnav a:focus {
```

```
  clip:auto;
```

```
  left:0;
```

```
  top:0;
```

```
  width:100%;
```

```
  height:auto;
```

```
  margin:0;
```

```
  padding:10px 0;
```

```
  background:#fdf6e7;
```

```
  border:2px solid #990000;
```

```
  border-left:none;
```

```
  border-right:none;
```

```
  text-align:center;
```

```
  font-weight:bold;
```

```
  color:#990000;
```

```
}
```

```

</style>
</head>
<body>
  <div id="skipnav"><a href="#mainContent">Skip navigation</a></div>
  <!-- document banner, navigation, etc. -->
  <main id="mainContent" tabindex="-1">
    <h1>Link will take users to this location.</h1>
    <!-- other content in the main content -->
  </main>
  <!-- other content on the web page -->
</body>

```

b) Andere overwegingen in verband met de interne navigatie

Inhoudsopgave

Een inhoudsopgave aan het begin van de pagina maakt de pagina-inhoud begrijpelijker en vergemakkelijkt de navigatie, vooral voor ziende personen die de muis niet kunnen gebruiken. De inhoudsopgave geeft een beknopt overzicht van de onderwerpen die op de pagina worden besproken en het anker maakt het mogelijk om rechtstreeks naar het gewenste onderdeel te navigeren.

Als er een inhoudsopgave wordt voorzien, moet die de koppenstructuur van de pagina weerspiegelen. Naargelang het geval kiezen we om alle kopniveaus in de lijst op te nemen of enkel <h2>, <h2> + <h3> ...

Tabvolgorde en raadpleging

De schermlezer leest de pagina in de volgorde van de inhoud in het DOM, dus stemt wat op het scherm zichtbaar is in principe overeen met de leesvolgorde voor slechtzienden, maar er moet worden opgelet voor CSS-posities of voor dynamische invoegingen, bijvoorbeeld via AJAX, die de volgorde waarin elementen verschijnen kunnen beïnvloeden.

Het voorlezen van de inhoud moet logisch en intuïtief zijn. Het uitschakelen van CSS is een goede techniek om de volgorde waarin de inhoud verschijnt te tonen.

De toetsenbordnavigatie via de tabtoets moet ook logisch zijn. De navigatie moet bovenaan de pagina beginnen en dan naar beneden gaan zonder stappen in de logische leesrichting over te slaan.

Tabindexen met positieve waarde mogen niet worden gebruikt. Eerst en vooral omdat die de logische tabvolgorde kunnen wijzigen en verwarring kunnen veroorzaken. Maar ook omdat ze het element uit de natuurlijke tabvolgorde verwijderen en het naar de eerste plaats in de tabvolgorde verplaatsen. Een tabindex met een waarde groter dan of gelijk aan 1 krijgt voorrang op alle andere elementen. Het gedrag van de tabtoets bij toetsenbordnavigatie kan daardoor sterk worden verstoord en de gebruiker totaal in verwarring brengen.

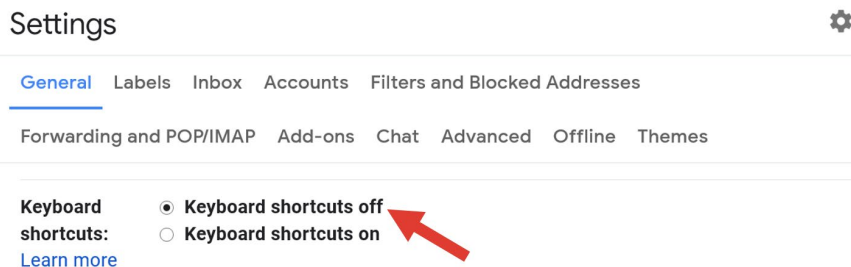
Sneltoetsen met één toets

De toepassingen en eigen parameters van de gebruiker, al dan niet in verband met een hulptechnologie, maken misschien gebruik van sneltoetsen.

Sommige internetgebruikers maken door bepaalde motorische beperkingen veel tikfouten en kunnen de toetsen soms onbedoeld activeren.

Als de website een sneltoets met één enkele toets bevat, moet de gebruiker die dus kunnen uitschakelen of wijzigen.

Zo gebruikt Gmail bijvoorbeeld veel sneltoetsen met één toets ('e' om een gesprek te archiveren, 'a' om te antwoorden aan iedereen, 'f' om te forwarden ...), maar krijgt de gebruiker wel de mogelijkheid om die opties in te stellen.



Lijsten

Als ze in een lijst terechtkomen, melden schermlezers de lijst, het begin en het einde van de lijst en het aantal elementen in die lijst. Dat gedrag is bijzonder nuttig voor blinden die daardoor, net zoals ziende gebruikers, een overzicht van de elementen krijgen.

a) Lijsten moet worden opgesteld volgens de correcte HTML-semantiek

Naargelang hun doel zijn de volgende lijsten mogelijk:

1. Lijsten zonder volgorde die elementen tussen -tags groeperen. Wordt gebruikt om een reeks items op te sommen die in eender welke volgorde kunnen worden geplaatst.
2. Lijsten met volgorde die elementen tussen -tags groeperen. Wordt gebruikt om een reeks elementen in een bepaalde volgorde op te sommen.
3. Definitielijsten <dl> die <dt>-termen gekoppeld aan hun <dd>-definitie groeperen.

Schermlezers begrijpen deze drie types en lezen de inhoud op een samenhangende manier voor.

Het gebruik van
 en/of lettertekens (streepjes, sterretjes ...) via het toetsenbord om valse lijsten op te stellen, laat schermlezers niet toe om de logica tussen die elementen te begrijpen. Het voorlezen wordt daardoor moeizamer en soms zelfs onbegrijpelijk.
 kan worden voorgelezen als 'nieuwe regel', het streepje als 'streepje' ...

Een correcte afbakening van lijsten heeft een positieve invloed op de SEO en maakt het gemakkelijker om coherente pagina's op de website te ontwerpen.

b) De tags voor lijsten mogen niet worden gebruikt voor andere inhoud dan lijsten.

Omwille van de layout, uit gebrek aan kennis, voor het 'gemak' ... worden soms bullets gebruikt om op een visuele manier de aandacht van de gebruiker te trekken (een bullet gebruikt als pijltje om een richting aan te geven ...).

Een dergelijk gebruik maakt het voorlezen moeizaam en kan tot misverstanden leiden.

Onthoud:

- Lijsten mogen enkel worden gebruikt om semantische redenen en niet voor de layout;
- Lijsten groeperen elementen die tot dezelfde familie behoren;
- Gebruik een lijst vanaf meer dan twee elementen uit dezelfde familie.

c) Vermijd het gebruik van geneste lijsten op meer dan 2 niveaus

Het is mogelijk om lijsten in andere lijsten te nesten. Vanaf twee nestniveaus kan dat moeilijk leesbaar en ingewikkeld worden.

In dat geval is het beter om de paginastructuur te herzien en extra kopniveaus te gebruiken.

Frames (Iframes)

Frames maken het mogelijk om externe inhoud in een kader op een webpagina te integreren. Die inhoud kan bestaan uit reclame, video's, publicaties op sociale netwerken, widgets, formulieren ... Soms bevatten die frames JavaScript-inhoud die voor de gebruikers verborgen blijft.

Met het oog op de toegankelijkheid moeten Iframes een titel bevatten om de gebruikers in staat te stellen om er doorheen te navigeren, om ze terug te vinden en om ze te identificeren. Het is ook belangrijk om te weten dat de elementen van het frame (kopniveaus, links ...) door schermlezers worden begrepen en behandeld als inhoud die deel uitmaakt van de pagina.

a) Kadertitels

Iframes die inhoud aanbieden aan gebruikers moeten een titel (title) hebben die informatief, expliciet en niet leeg is. Bovendien moet hij uniek zijn in de context van de pagina (niet tweemaal dezelfde titel op één pagina).

De titel mag niet algemeen zijn en mag geen tekst van het type 'video' of 'reclame' bevatten.

Goed om te weten: JAWS leest eerst de titel van het document in het Iframe. Als die niet bestaat, leest JAWS de titel van het Iframe. Om misverstanden te vermijden, vermelden we dus best eerst een Iframe-title die lijkt op de titel van de broninhoud en zorgen we ervoor dat die een correcte titel bevat (in de veronderstelling dat we over die content enige controle hebben).

Goed voorbeeld van het gebruik van 'title' in Iframes

```
<iframe
```

```
title="Specify the Language" width="560" height="315"
src="https://www.youtube.com/embed/qyjDrUV_el8" frameborder="0" allowfullscreen>
```

```
</iframe>
```

```
<iframe
```

```
title="Video of touch screen for the blind in New York City taxis"
style="width:640px;height:480px;margin:auto;text-align:center;"
src="//www.youtube.com/embed/hM0x0k2Bv3Y" allowfullscreen>
```

```
</iframe>
```

b) De semantische structuur van frames

De inhoud van de frames maakt integraal deel uit van de pagina en wordt dus als dusdanig beschouwd door de schermlezer. Als de pagina vijf kopniveaus bevat en het frame drie, zal de schermlezer acht kopniveaus vermelden.

Voor zover het mogelijk is om de inhoud van het Iframe te wijzigen, zorgen we er dus voor dat zijn structuur past in de structuur waarin het zal worden weergegeven. Zo moeten we bijvoorbeeld vermijden om meer dan één <h1> op de pagina te vinden.

c) Verberg frames die geen betekenisvolle inhoud aanbieden

Soms brengt een inhoud de gebruiker niets bij, wordt hij beschouwd als decoratieve inhoud of kan hij zelfs het voorlezen door schermlezers moeilijker maken (bijv. Google Maps).

In dat geval is het beter om het Iframe te verbergen met het attribuut `aria-hidden="true"`.

Voorbeeld van het gebruik van aria om inhoud voor schermlezers te verbergen

```
<iframe
```

```
  title="Intentionally blank"  aria-hidden="true"  src="http://bit.ly/2cfBoyE"  width="120"
  height="50">
```

```
</iframe>
```

Q. Tabellen

De gebruikers van schermlezers kunnen in tabellen van cel naar cel navigeren. Als de tabellen correct gestructureerd zijn, hoort de internetgebruiker het verband tussen de koppen en de overeenkomstige cellen. Er moeten ook middelen worden voorzien om iedereen in staat te stellen om de tabellen goed te begrijpen.

[Een goede semantiek voor datatabellen](#)

Datatabellen moeten gebruik maken van <table>, de HTML-tags bestemd voor tabellen ... Soms worden valse tabellen aangemaakt in CSS om een betere leesbaarheid mogelijk te maken, met name op smartphones.

Zonder de correcte semantiek kunnen de gebruikers van schermlezers de verbanden tussen cellen en koppen niet begrijpen en kunnen ze niet navigeren zoals in een tabel.

[Bijschriften bij tabellen](#)

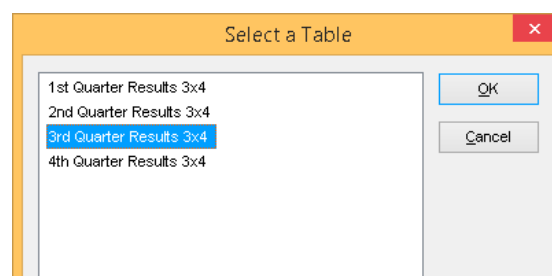
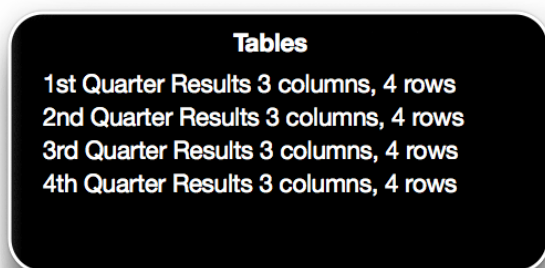
a) *Datatabellen moeten technisch aan een bijschrift of een naam worden gekoppeld*

Schermlezers lezen het bijschrift of de naam die aan de tabel werden gekoppeld wanneer de gebruiker in de tabel navigeert.

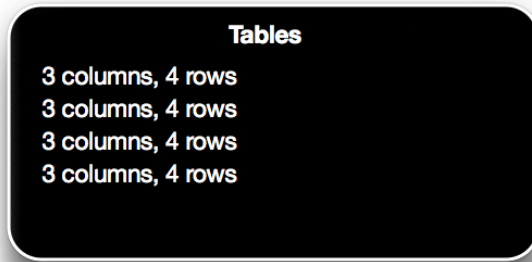
Het <caption>-element is de meest directe techniek daarvoor, maar het is ook mogelijk om aria-label of aria-labelledby te gebruiken.

De internetgebruiker kan zijn schermlezer gebruiken om een lijst van de op de pagina aanwezige tabellen te horen. Als er een bijschrift bij de tabel staat, zal dat ook worden voorgelezen. Anders bevat de lijst slechts een overzicht van de tabellen (aantal rijen en kolommen) zonder enige andere informatie.

De twee onderstaande afbeeldingen tonen de lijst van vier correct van een bijschrift voorziene tabellen (VoiceOver en Jaws)



De afbeelding hieronder toont dezelfde tabellen zonder bijschrift in VoiceOver.



Goed voorbeeld: een tabel met bijschrift via het element <caption>

```

<table>
  <caption>1st Quarter Results (EUR)</caption>
  <thead>
    <tr>
      <td>&nbsp;</td>
      <th scope="col">Goal</th>
      <th scope="col">Actual </th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">January</th>
      <td>200</td>
      <td>150</td>
    </tr>
    <tr>
      <th scope="row">February</th>
      <td>200</td>
      <td>240</td>
    </tr>
    <tr>
      <th scope="row">March</th>
      <td>200</td>
      <td>370</td>
  </tbody>
</table>

```

```
</tr>
```

```
</tbody>
```

```
</table>
```

Goed voorbeeld: Een bijschrift met aria-label

```
<table aria-label="Second quarter Results">
```

Opgelet, de tekst in aria-label is niet zichtbaar voor gebruikers van schermlezers.

Goed voorbeeld: Een in CSS verborgen bijschrift

```
<table>
```

```
  <caption><span class="visually-hidden">1st Quarter Results</span></caption>
```

Er wordt een CSS-klasse gebruikt om het bijschrift te verbergen, maar het wel leesbaar te maken voor schermlezers. Opgelet, de klasse mag niet rechtstreeks op de <caption> worden geplaatst, want NVDA leest dan niet het correcte aantal rijen.

Goed voorbeeld: Een bijschrift in aria-labeledby

```
<h3 id="tableCaption">Third Quarter Results</h3>
```

```
<table aria-labelledby="tableCaption">
```

Een variant op deze methode bestaat erin de naam in een <figcaption> te plaatsen en die aan de tabel te koppelen via aria-labelledby.

```
<figure>
```

```
<table aria-labelledby="figCaption">
```

```
<!--data table-->
```

```
</table>
```

```
<figcaption id="figCaption">Fourth Quarter Results</figcaption>
```

```
</figure>
```

Slecht voorbeeld: Een tabel met vals bijschrift

Deze tabel bevat geen bijschrift, maar begint met een koprij die drie cellen samenvoegt. Niet alleen wordt dat bijschrift niet correct verwerkt, maar de navigatie in de tabel zelf wordt verwarrend en moeilijk.

```
<table>
```

```
  <tr>
```

```
    <th colspan="3">Revenue Goals (EUR)</th>
```

```
  </tr>
```

```
  <tr>
```

```
    <td>&nbsp;</td>
```

```
    <th scope="col">Goal</th>
```

```
    <th scope="col">Actual</th>
```

```
  </tr>
```

```

<tr>
  <th scope="row">January</th>
  <td>200</td>
  <td>150</td>
</tr>
<tr>
  <th scope="row">February</th>
  <td>200</td>
  <td>240</td>
</tr>
<tr>
  <th scope="row">March</th>
  <td>200</td>
  <td>370</td>
</tr>
</table>

```

Slecht voorbeeld: Een tabelkop die niet aan de tabel is gekoppeld

```
<h3>Revenue Goals</h3>
```

```
<table>
```

```
<!--data table -->
```

De kop `<h3>` lijkt op de tabelkop, maar zonder technische koppeling kan de schermlezer het verband niet leggen.

b) Het bijschrift van de tabel moet het onderwerp van de tabel uniek, expliciet, betekenisvol en beknopt beschrijven.

De bedoeling van deze regel is om het mogelijk te maken de juiste tabel snel en foutloos te identificeren.

In dezelfde lijn moet elke tabel, op een pagina die meerdere tabellen bevat, telkens over een unieke kop beschikken.

Koppen van datatabellen

a) De koppen van datatabellen moeten worden afgebakend met <th>

De `<th>`-tag wordt gebruikt om de gebruiker van een schermlezer te laten weten dat hij zich in een kopcel bevindt.

Het is aanbevolen om de scope van de kop expliciet te maken door een attribuut `scope="col"` of `"row"` of `"colgroup"` of `"rowgroup"` aan de `<th>`-tag toe te voegen.

b) De koppen van datatabellen moeten specifiek de categorie van de inhoud van de cellen beschrijven

De koptekst mag niet vaag zijn en moet de gebruiker duidelijk informeren over de inhoud van de cellen. Schrijf bijvoorbeeld niet 'kolom 2' en beschrijf expliciet het alternatief voor een knop of een link.

Blijf in regel kort en duidelijk.

Koppelingen tussen koppen en cellen van datatabellen

a) De datacellen moeten aan hun overeenkomstige kopcel worden gekoppeld

Het scope-attribuut scheidt een expliciete koppeling tussen de kopcel en de overeenkomstige datacel. De opties zijn:

- Scope="col" voor kolomkoppen,
- Scope="row" voor rij-koppen.

Goed voorbeeld: Een eenvoudige tabel met rij- en kolomkoppen

In de onderstaande tabel bevat de eerste rij de koppen voor de cellen in elke kolom en de eerste kolom bevat de koppen voor de cellen in elke rij.

**Greensprings Running Club
Personal Bests**

Name	1 mile	5 km	10 km
Mary	8:32	28:04	1:01:16
Betsy	7:43	26:47	55:38
Matt	7:55	27:29	57:04
Todd	7:01	24:21	50:35

```
<table class="data">
```

```
<caption><strong>Greensprings Running Club Personal Bests</strong></caption>
```

```
<thead>
```

```
<tr>
```

```
<th scope="col">Name</th>
```

```
<th scope="col">1 mile</th>
```

```
<th scope="col">5 km</th>
```

```
<th scope="col">10 km</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
<tr>
```



```

<th scope="row">Mary</th>
<td>8:32</td>
<td>28:04</td>
<td>1:01:16</td>
</tr>
<tr>
<th scope="row">Betsy</th>
<td>7:43</td>
<td>26:47</td>
<td>55:38</td>
</tr>
<tr>
<th scope="row">Matt</th>
<td>7:55</td>
<td>27:29</td>
<td>57:04</td>
</tr>
<tr>
<th scope="row">Todd</th>
<td>7:01</td>
<td>24:21</td>
<td>50:35</td>
</tr>
</tbody>
</table>

```

b) In de koppen van gegroepeerde tabellen moeten datacellen aan hun overeenkomstige kopcellen worden gekoppeld

De gebruikers van schermlezers en alle gebruikers in het algemeen kunnen moeite hebben om dergelijke tabellen met gecombineerde koppen te begrijpen en erin te navigeren. In de mate van het mogelijke wordt aanbevolen om data te structureren in eenvoudige tabellen (één koppenrij of -kolom zonder samengevoegde cellen).

De beste manier om een groepering van kopcellen voor schermlezers aan te duiden is het gebruik van `scope="rowgroup"` of `scope="colgroup"`.

Goed voorbeeld: toepassing van scope="colgroup" op samengevoegde cellen van kolomkoppen

	Females			Males		
	Mary	Betsy	Joanne	Matt	Todd	Jake
1 mile	8:32	7:43	9:51	7:55	7:01	7:51
5 km	28:04	26:47	38:15	27:27	24:21	24:31
10 km	1:01:16	55:38	1:56:01	57:04	50:35	50:45

```

<table class="data complex">
  <caption>
    Table with colgroup
  </caption>
  <thead>
    <tr>
      <td rowspan="2">&nbsp;</td>
      <th colspan="3" scope="colgroup">Females</th>
      <th colspan="3" scope="colgroup">Males</th>
    </tr>
    <tr>
      <th scope="col">Mary</th>
      <th scope="col">Betsy</th>
      <th scope="col">Joanne</th>
      <th scope="col">Matt</th>
      <th scope="col">Todd</th>
      <th scope="col">Jake</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">1 mile</th>
      <td>8:32</td>
      <td>7:43</td>
  
```

```

    <td>9:51</td>
    <td>7:55</td>
    <td>7:01</td>
    <td>7:51</td>
  </tr>
  <tr>
    <th scope="row">5 km</th>
    <td>28:04</td>
    <td>26:47</td>
    <td>38:15</td>
    <td>27:27</td>
    <td>24:21</td>
    <td>24:31</td>
  </tr>
  <tr>
    <th scope="row">10 km</th>
    <td>1:01:16</td>
    <td>55:38</td>
    <td>1:56:01</td>
    <td>57:04</td>
    <td>50:35</td>
    <td>50:45</td>
  </tr>
</tbody>
</table>

```

Goed voorbeeld: scope="rowgroup" toegepast op samengevoegde cellen van rijkoppen

Dit voorbeeld toont een andere weergave van dezelfde tabel.

Schermlezers hebben meer moeite met de scope="rowgroup". In de mate van het mogelijke wordt dus aanbevolen om complexe tabellen te structureren voor het gebruik van scope="colgroup".

Table with rowgroup				
		1 mile	5 km	10 km
Females	Mary	8:32	28:04	1:01:16
	Betsy	7:43	26:47	55:38
	Joanne	9:51	38:15	1:56:01
Males	Matt	7:55	27:29	57:04
	Todd	7:01	24:21	50:35
	Jake	7:51	24:31	50:45

```

<table class="data complex">
  <caption>
    Table with rowgroup
  </caption>
  <thead>
    <tr>
      <td colspan="2">&nbsp;</td>
      <th scope="col">1 mile</th>
      <th scope="col">5 km</th>
      <th scope="col">10 km</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th rowspan="3" scope="rowgroup">Females</th>
      <th scope="row">Mary</th>
      <td>8:32</td>
      <td>28:04</td>
      <td>1:01:16</td>
    </tr>
    <tr>
      <th scope="row">Betsy</th>
      <td>7:43</td>
  
```

```

<td>26:47</td>
<td>55:38</td>
</tr>
<tr>
  <th scope="row">Joanne</th>
  <td>9:51</td>
  <td>38:15</td>
  <td>1:56:01</td>
</tr>
<tr>
  <th rowspan="3" scope="rowgroup">Males</th>
  <th scope="row">Matt</th>
  <td>7:55</td>
  <td>27:29</td>
  <td>57:04</td>
</tr>
<tr>
  <th scope="row">Todd</th>
  <td>7:01</td>
  <td>24:21</td>
  <td>50:35</td>
</tr>
<tr>
  <th scope="row">Jake</th>
  <td>7:51</td>
  <td>24:31</td>
  <td>50:45</td>
</tr>
</tbody>
</table>

```

c) In complexe tabellen moeten datacellen die niet via <th> en scope aan hun overeenkomstige kopcellen kunnen worden gekoppeld, worden aangeduid met headers + id

Complexe datatabellen met samengevoegde datacellen of twee kopniveaus vereisen het gebruik van een andere techniek dan het gebruik van <th> en scope om het verband met de gekoppelde cellen te leggen.

De methode bestaat erin de cellen te koppelen via headers en id.

Opgelet, dergelijke tabellen kunnen moeilijk te begrijpen zijn en de navigatie met een schermlezer zal altijd moeilijker zijn, ondanks de gebruikte toegankelijkheidstechnieken. Het blijft dus steeds aanbevolen om de structuur van tabellen zo eenvoudig mogelijk te houden.

Sommige schermlezers, vooral op mobiele apparaten, kunnen complexe tags niet goed verwerken. De oude versies van VoiceOver op OS X tot en met Mac OS X 10.10.2 geven leesfouten voor complexe tabellen die gebruik maken van koppelingen via headers + id.

Goed voorbeeld: Een complexe tabel met headers + id

Het onderstaande voorbeeld beschrijft een complexe tabel die aan de toegankelijkheidsnormen voldoet, maar desondanks moeilijk te begrijpen blijft. De navigatie via een schermlezer is moeizamer of zelfs onmogelijk met sommige oudere versies van schermlezers.

Belangrijke opmerking: ID's mogen niet beginnen met cijfers, maar uitsluitend met letters.

New Employee Orientation Schedule

Date	Schedule		Location	Topics
	Start	End		
Monday, June 1	9:00 a.m.	10:30 a.m.	RH 001	Introduction to Company: Vision and Mission
	10:30 a.m.	12:00 p.m.	RH 001	HR Policies Review
	<i>Lunch from 12:00 p.m. to 1:00 p.m.</i>			
	1:00 p.m.	2:30 p.m.	RH 001	Overview of Benefits
	3:00 p.m.	4:30 p.m.	RH 005	Health and Safety Procedures

```
<table class="complexexample">
```

```
<caption>New Employee Orientation Schedule</caption>
```

```
<tbody>
```

```
<tr>
```

```
<th rowspan="2" id="date">Date</th>
```

```
<th colspan="2" id="schedule">Schedule</th>
```

```
<th rowspan="2" id="location">Location</th>
```

```
<th colspan="2" rowspan="2" id="topics1">Topics</th>
```

```

</tr>
<tr>
  <th id="start">Start</th>
  <th id="end">End</th>
</tr>
<tr>
  <th id="monday" rowspan="5">Monday, June 1</th>
  <td headers="schedule start monday">9:00 a.m.</td>
  <td headers="schedule end monday">10:30 a.m.</td>
  <td headers="location monday">RH 001</td>
  <td headers="topics1 monday">
    Introduction to Company: Vision and Mission</td>
</tr>
<tr>
  <td headers="schedule start monday">10:30 a.m.</td>
  <td headers="schedule end monday">12:00 p.m.</td>
  <td headers="location monday">RH 001</td>
  <td headers="topics1 monday">HR Policies Review</td>
</tr>
<tr>
  <td headers="schedule monday" colspan="5">
    <strong><em>
      Lunch from 12:00 p.m. to 1:00 p.m.
    </em></strong>
  </td>
</tr>
<tr>
  <td headers="schedule start monday">1:00 p.m.</td>
  <td headers="schedule end monday">2:30 p.m.</td>
  <td headers="location monday">RH 001</td>
  <td headers="topics1 monday">Overview of Benefits</td>
</tr>

```

```

<tr>
  <td headers="schedule start monday">3:00 p.m.</td>
  <td headers="schedule end monday">4:30 p.m.</td>
  <td headers="location monday">RH 005</td>
  <td headers="topics1 monday">
    Health and Safety Procedures
  </td>
</tr>
</tbody>
</table>

```

Hoewel het moeilijk is om die structuur ‘met de hand’ in HTML te schrijven, is ze technisch eenvoudig te programmeren aan de serverzijde dankzij programmeertalen zoals PHP, .net, JSP, Python ... om tabellen te maken die database-exports weergeven.

d) Geneste of gespleten tabellen

In tabellen die zich voordoen als één tabel, maar die verschillende onderdelen van geneste tabellen bevatten, kunnen de hierboven besproken koppeltechnieken tussen koppen en cellen (scope en headers + id) niet worden gebruikt omdat het onmogelijk is om cellen uit verschillende tabellen te koppelen.

De enige oplossing zou het gebruik van aria-labelledby zijn, maar dat levert bijzonder veel werk op om een slecht ontworpen tabel te compenseren.

Slecht voorbeeld: Geneste tabellen

Greensprings Running Club Personal Bests

		1 mile	5km	10km
Females	Mary	8:32	28:04	1:01:16
	Betsy	7:43	26:47	55:38
Males	Matt	7:55	27:29	57:04
	Todd	7:01	24:21	50:35

```

<table class="data">
<caption>
Greensprings Running Club Personal Bests
</caption>
<tr>
<td>&nbsp;</td>
<td style="padding:0">

```



```

<table style="margin:0;padding:0">
<tr>
<td style="width:60px;">&nbsp;</td>
<th style="width:60px">1 mile</th>
<th style="width:60px">5km</th>
<th style="width:60px">10km</th>
</tr>
</table></td>
</tr>
<tr>
<th>Females</th>
<td style="padding:0">
<table style="margin:0;padding:0">
<tr>
<th style="width:60px;">Mary</th>
<td style="width:60px">8:32</td>
<td style="width:60px">28:04</td>
<td style="width:60px">1:01:16</td>
</tr>
<tr>
<th style="width:60px;">Betsy</th>
<td>7:43</td>
<td>26:47</td>
<td>55:38</td>
</tr>
</table>
</td>
</tr>
<tr>
<th>Males</th>
<td style="padding:0">
<table style="margin:0;padding:0">

```

```

<tr>
<th style="width:60px;">Matt</th>
<td style="width:60px">7:55</td>
<td style="width:60px">27:29</td>
<td style="width:60px">57:04</td>
</tr>
<tr>
<th style="width:60px;">Todd</th>
<td>7:01</td>
<td>24:21</td>
<td>50:35</td>
</tr>
</table></td>
</tr>
</table>

```

[Tabelsamenvattingen \(summary\)](#)

a) Voor datatabellen moet een tabelsamenvatting worden voorzien

Een samenvatting is niet verplicht. De tabelstructuur en de koptitels zouden moeten volstaan om de gegevens in de tabel te begrijpen.

Als de tabel extra informatie vereist om begrijpelijker te zijn, kan echter een samenvatting worden gegeven.

De samenvatting heeft tot doel om de gebruiker te helpen de tabel beter te begrijpen. Ze mag in geen geval worden gebruikt om sleutelwoorden voor de SEO uit te werken.

De inhoud moet beknopt en expliciet zijn. Ze is maximaal twee zinnen lang, maar mag soms iets langer zijn om complexe tabellen te beschrijven. Er zijn vier manieren om een samenvatting aan te bieden:

1. De samenvatting wordt geschreven in een paragraaf die voorafgaat aan of volgt op de tabel. De paragraaf wordt aan de tabel gekoppeld via `aria-describedby`;
2. Als ze heel kort is, kan de samenvatting in de `<caption>` van de tabel worden geschreven;
3. De tabel staat tussen `<figure>`-tags en haar kop en samenvatting worden in de `<figcaption>` geschreven;
4. Enkel voor HTML ouder dan HTML5 mag de samenvatting in het `summary`-attribuut staan. Dit raakt in onbruik en wordt dus niet aanbevolen.

Goed voorbeeld: een samenvatting in een gekoppelde paragraaf met aria-describedby (methode 1)

Geschreven tekst die zich in een paragraaf bevindt, is voor iedereen leesbaar. De expliciete koppeling via aria-describedby maakt het mogelijk voor schermlezers om het verband tussen de paragraaf en de tabel te leggen.

```
<p id="table-description">This table lists the members of the  
Greensprings Running Club and their personal best times in various race distances.  
The first column lists the runners and the first row lists the race distances.</p>
```

```
<table aria-describedby="table-description">  
  <caption>Greensprings Running Club Personal Bests</caption>  
  <tr>  
    <th scope="col">Name</th>  
    <th scope="col">1 mile</th>  
    <th scope="col">5 km</th>  
    <th scope="col">10 km</th>  
  </tr>  
  <tr>  
    <th scope="row">Mary</th>  
    <td>8:32</td>  
    <td>28:04</td>  
    <td>1:01:16</td>  
  </tr>  
  <tr>  
    .....  
</table>
```

Het is mogelijk om de samenvatting visueel te verbergen door in CSS een klasse toe te passen die gebruik maakt van de in de andere hoofdstukken besproken clip-methode. Die methode is perfect toegankelijk voor de gebruikers van schermlezers, maar de inhoud die zo werd verborgen zou ook nuttig kunnen zijn voor andere internetgebruikers die moeite hebben om de tabel te begrijpen.

```
<p id="table-description" class="visually-hidden">This table lists the members of the  
Greensprings Running Club and their personal best times in various race distances.  
The first column lists the runners and the first row lists the race distances.</p>
```

Goed voorbeeld: een korte samenvatting in de <caption> van de tabel (methode 2)

```
<table>  
  <caption>
```

Greensprings Running Club Personal Bests

(The first column lists the runners and the first row lists the race distances)

</caption>

<tr>

<th scope="col">Name</th>

<th scope="col">1 mile</th>

<th scope="col">5 km</th>

<th scope="col">10 km</th>

</tr>

.....

</table>

Goed voorbeeld: Een tabel tussen <figure>-tags met een samenvatting in de <figcaption>-tag (methode 3)

Goed om te weten: De inhoud van <figure>-tags voorlezen met een schermlezer duurt langer omdat ze meer interacties vereist en het woord <figure> algemener is, zodat de gebruiker niet meteen begrijpt over welk type element het gaat.

De techniek is echter wel perfect toegankelijk.

<figure>

<figcaption id="table_figcaption">

Greensprings Running Club Personal Bests

(The first column lists the runners and the first row lists the race distances)

</figcaption>

<table aria-labelledby="table_figcaption">

<tr>

<th scope="col">Name</th>

<th scope="col">1 mile</th>

<th scope="col">5 km</th>

<th scope="col">10 km</th>

</tr>

<tr>

<th scope="row">Mary</th>

<td>8:32</td>

<td>28:04</td>

```

<td>1:01:16</td>
</tr>
<tr>
  <th scope="row">Betsy</th>
  <td>7:43</td>
  <td>26:47</td>
  <td>55:38</td>
</tr>
<tr>
  <th scope="row">Matt</th>
  <td>7:55</td>
  <td>27:29</td>
  <td>57:04</td>
</tr>
<tr>
  <th scope="row">Todd</th>
  <td>7:01</td>
  <td>24:21</td>
  <td>50:35</td>
</tr>
</table>

```

```
<figure>
```

Goed voorbeeld: Een tabel tussen <figure>-tags met minstens de samenvatting in dezelfde <figure> (methode 3)

```

<figure>
  <figcaption>
    Greensprings Running Club Personal Bests
  </figcaption>
  <p>(The first column lists the runners and the first row lists the race distances) </p>
<table>
  ...
</table>
</figure>

```

Lay-out-tabellen

a) Tabellen mogen niet worden gebruikt voor ontwerpdoeleinden (layout)

Tabellen mogen enkel worden gebruikt om data weer te geven. Semantisch werden ze met die bedoeling ontworpen. CSS is beter geschikt voor layout en is trouwens robuuster en beter aangepast aan het moderne web.

Navigeren in een tabel via een schermlezer duurt langer en de gebruiker hoort dat hij zich in een tabel bevindt, zodat hij verwacht gegevens te vinden die worden weergegeven in cellen met koppen ...

Als er echt een tabel moet worden gebruikt, dan moet daarin de rol `role="presentation"` worden gespecificeerd om de gebruikers van schermlezers toe te laten de inhoud als lineaire tekst te horen.

b) Lay-out-tabellen mogen geen semantische elementen in verband met tabellen bevatten

Als de tabel semantische tags bevat die specifiek voor tabellen bestemd zijn, zal de schermlezer die als dusdanig voorlezen en aan de gebruiker suggereren dat hij een datatabel leest.

De tabel mag met andere woorden geen gebruik maken van de volgende elementen:

- `<caption>`
- `<summary>`
- `<th>`
- `scope`
- `headers + id`

R. Dynamische componenten

Het is mogelijk om toegankelijke dynamische componenten aan te maken via JavaScript. De documentatie over elke component is online beschikbaar op de website van het [W3C-WAI](#)

Dit document bevat dus niet alle regelmatig bijgewerkte hulpmiddelen die op de officiële website beschikbaar zijn, maar geeft aanvullende informatie en enkele voorbeelden.

ARIA-concepten

ARIA (Accessible Rich Internet Applications) werd specifiek ontworpen voor de toegankelijkheid van het web en meer bepaald voor hulptechnologieën. Het betreft elementen om de beperkingen van HTML5 te compenseren en/of om websites te corrigeren waar het om allerlei redenen moeilijk is om de structuur te herzien.

Het is belangrijk om de volgende punten voor ogen te houden alvorens ARIA te gebruiken:

- Gebruik ARIA nooit als het niet nodig is;
- Gebruik ARIA altijd als het nodig is;
- Zonder de nodige kennis wordt ARIA gewoonlijk slecht gebruikt.

ARIA is met andere woorden steeds een laatste redmiddel en moet met kennis van zaken worden gebruikt.

Dynamische componenten in JavaScript gebruiken vaak ARIA om de hierboven aangehaalde redenen. Elke component (widget) moet aan de toegankelijkheidsnormen voldoen om te beantwoorden aan de behoeften van de internetgebruikers die al dan niet gebruik maken van hulptechnologieën (navigatie enkel via de muis, enkel via het toetsenbord, via hulptechnologieën ...).

a) Wat doet ARIA?

ARIA biedt de mogelijkheid om aan schermlezers de volgende informatie mee te delen:

- Labels (label): de naam van de elementen. Voorbeeld: `aria-label`;
- Rollen (role): de bedoeling van de elementen. Voorbeeld: `role="navigation"`;
- De status (state) van dynamische elementen: hun toestand. Voorbeeld: `aria-selected="true"`;
- Eigenschappen (properties). Voorbeeld: `aria-haspopup="true"`;
- De relaties (relationships) tussen elementen. Voorbeeld: `aria-controls`;
- Aankondigingen in realtime (live announcements). Live regions die een wijziging aan de pagina aankondigen.

b) Toegankelijke naam (Name)

De toegankelijke naam is de naam van het element zoals hij wordt geproduceerd door de accessibility tree, ingeschreven in het DOM, meegedeeld aan de schermlezer en voorgelezen aan de gebruiker.

In de onderstaande illustratie toont het 'Accessibility'-tabblad van de code-inspecteur de toegankelijke naam 'Learn more about APG patterns examples' van de hyperlink 'Learn more'. De toegankelijke naam werd gegenereerd op basis van het `aria-label`-attribuut van de link.

De illustratie toont ook de rol (link), de status en de waarde van het element (`Focusable:true`, `Focused:true`).

Building blocks that help you make the web accessible

Design Patterns and Examples

Learn how to make accessible web components and widgets with ARIA roles, states and properties and by implementing keyboard support. One or more ways of implementing each pattern is demonstrated with a functional example.

#text 86.52 × 16

[Learn More](#)

Use ARIA Landmarks

Learn how to use HTML sectioning elements and ARIA landmark roles to make it easy for assistive technology users to understand the meaning of the layout of a page.

[Learn More](#)

```

<div>
  <link rel="stylesheet" href="/WA1/content-assets/wai-aria-practices/homepage.css">
  <div class="off-white-section">...</div>
  <div class="white-section">
    <div class="centered">...</div>
    <div class="contained margin-fix">
      ::before
      <div class="resource-item">
        <div class="resource-item-content">
          <h3>Design Patterns and Examples</h3>
          <p>...</p>
          <a href="/patterns/" aria-label="Learn more about APG patterns and examples" class="button-link">Learn More
        </a>
      </div>
      <div class="resource-item-img">...</div>
    </div>
    <div class="resource-item">...</div>
    <div class="resource-item">...</div>
    <div class="resource-item">...</div>
    ::after
  </div>
  <div class="work-grid-pattern grid-pattern"></div>
</div>
</div>
<div class="off-white-section" style="display: none; /* tempo rarily hidden because the linked resources are not yet polished */>...</div>
<div class="white-section">...</div>
<div class="bottom-off-white-section off-white-section">
</div>
<script src="/WA1/ARIA/app/example-index/js/skipto.js">
</script>
</main>
</div>
    
```

Accessibility Tree

- RootNodeArea "ARIA Authoring Practices Guide | APG | WA1"
 - generic
 - main
 - generic
 - Link "Learn more about APG patterns and examples"
 - StaticText "Learn More"

ARIA Attributes

- aria-label: Learn more about APG patterns and examples

Computed Properties

- Name: "Learn more about APG patterns and examples"
 - aria-labelledby: Not specified
 - aria-label: "Learn more about APG patterns and examples"
 - contents: "Learn More"
 - title: Not specified
 - Role: Link
 - Focusable: true
 - Focused: true

Berekening van de toegankelijke naam (algoritme)

Eenvoudig gezegd is dit de volgorde waarin de toegankelijke naam wordt berekend:

1. Aria-labelledby. De tekst van dit attribuut overschrijft als hij bestaat alle andere labelteksten voor dit element;
2. Aria-label. Zonder aria-labelledby overschrijft de tekst van aria-label de andere teksten. In de bovenstaande illustratie was de inhoud van de link tussen de tags `<a>` en `` 'Learn more'. Zonder de tekst van het attribuut `aria-label="Learn more about APG patterns and examples"` zou de toegankelijke naam 'Learn more' zijn (wat buiten de context niet expliciet is). Vergeet niet dat aria-label op het scherm onzichtbaar is en enkel beschikbaar is voor gebruikers van een schermlezer;
3. De tekst van het systeemeigen HTML-element of zijn label of zijn tekstalternatief. Zonder de twee bovenstaande attributen vormt de systeemeigen tekst de toegankelijke naam. Het kan daarbij gaan om de tekst tussen de openings- en sluitingstags van een knop (`<button>`), het `<label>` van een correct gekoppeld formulierveld, een alternatief voor een afbeelding (`alt`), de tekst tussen de hyperlinktags `<a>` en `` (`alt` en `title` van een afbeelding inbegrepen) ...;
4. Het title-attribuut. Als geen van de bovenstaande elementen tekst bevat, wordt in de laatste plaats de title gebruikt om de toegankelijke naam te berekenen. Opgelet, de title wordt niet beschouwd als een toegankelijke techniek omdat zijn inhoud enkel zichtbaar is bij muisbewegingen (hover) en dus onzichtbaar blijft voor toetsenbordgebruikers.

Hoewel de tekst van het systeemeigen element pas op de derde plaats staat, is dat de methode die in de eerste plaats de voorkeur verdient.

Aria-labelledby,

De tekst van aria-labelledby is meestal zichtbaar op de pagina voor alle ziende gebruikers (in tegenstelling tot aria-label). De bedoeling is om technisch een element aan die tekst (label) te koppelen via een id zodat hij door de schermlezer wordt voorgelezen als die voorbij het element komt. In sommige zeldzame gevallen kan die tekst op het scherm onzichtbaar worden gemaakt, zodat ziende gebruikers worden benadeeld.

Het attribuut `aria-labelledby` vermeldt het id van het element dat de tekst bevat. Het bevat zelf geen tekst (in tegenstelling tot `aria-label`).

Opgelet, `aria-labelledby` vervangt het bestaande label, maar voegt geen extra tekst toe. Hoewel de schermlezer in sommige configuraties zowel het oorspronkelijke label als `aria-labelledby` voorleest. Met dat bijzondere geval moet echter geen rekening worden gehouden.

[Voorbeeld van aria-labelledby](#)

In dit voorbeeld draagt de `div` die de inhoud van de pop-up bevat de naam van zijn `<h1>`, die wordt geïdentificeerd door de id 'h1': 'Confirm your selection'

```
<div class="modal" role="dialog" tabindex="0" aria-labelledby="h1">
```

```
<h1 id="h1">Confirm your selection</h1>
```

Een van de sterke punten van dit attribuut is dat het naar meerdere elementen kan verwijzen. Dat kan bijzonder interessant zijn voor formulieren waarin bepaalde velden meerdere verschillende inlichtingen moeten bevatten om zo de gebruiker te helpen om die in te vullen.

[Voorbeeld van aria-labelledby met meerdere ID's](#)

```
<span id="males"> ... <span id="frank"> ... <span id="ranking">
```

```
<input type="text" aria-labelledby="males frank ranking">
```

Aria-label

Het attribuut `aria-label` bevat de tekst van het label zelf en is niet zichtbaar op het scherm. Enkel de gebruiker van een schermlezer kan dat raadplegen.

[Voorbeeld van aria-label](#)

Het onderstaande navigatie-element krijgt als naam de inhoud van het attribuut `aria-label`.

```
<nav aria-label="Clothing for Girls">
```

De tekst van `aria-label` vervangt de bestaande tekst. Het is niet bedoeld om aanvullende informatie te verstrekken, omdat het enkel zichtbaar is voor schermlezers.

[Voorbeeld van aria-label ter vervanging van de systeemeigen tekst van een hyperlink](#)

In het onderstaande voorbeeld horen gebruikers van schermlezers niet dezelfde informatie als de informatie die zichtbaar is op de pagina.

```
<a href="http://w3.org" aria-label="The World Wide Web Consortium">W3C</a>
```

[Voorbeeld van aria-label op landmark-regio's](#)

Als de website slechts één hoofdnavigatie bevat, is het niet nodig om die een naam te geven. De schermlezer leest dan 'Navigatie'.

Als de pagina echter meerdere navigaties bevat, kan het nuttig zijn om die een naam te geven zodat de gebruiker vlotter kan navigeren door de verschillende navigatiemenu's van elkaar te onderscheiden.

```
<nav role="navigation" aria-label="Product Categories">
```

[Voorbeeld van aria-label op een zoekmachine](#)

Er bestaan specifieke gevallen waarin het element niet noodzakelijk een label moet hebben. Dat is het geval bij een zoekformulier, waar de nabijheid van de 'Search'-knop voor ziende gebruikers visueel de bedoeling van de knop aangeeft.

Blinde gebruikers kunnen de bedoeling van het veld jammer genoeg pas afleiden als ze verder navigeren. Voor die mensen kan het ontbreken van een label een probleem zijn.

In dat geval is aria-label een manier om dit veld voor hen expliciet te maken zonder op het scherm een extra label toe te voegen.

```
<form action="#" role="search">
  <input aria-label="Site Search" name="search" type="search">
  <input type="submit" value="Search">
</form>
```

c) Role

Elk HTML-element heeft een rol (role) met eigenschappen en methodes om de gebruiker informatie te verschaffen. Hulptechnologieën leiden uit de rol van het element af welke interactiemiddelen ze aan de gebruiker moeten aanbieden.

De tag meldt bijvoorbeeld de role 'afbeelding' aan de schermlezer, die de gebruiker meedeelt, dat hij zich op een 'afbeelding' of een 'graphic' bevindt (het precieze woord hangt af van de schermlezer) en die het tekstalternatief in het alt-attribuut voorleest. Als dat alt-attribuut ontbreekt, leest de schermlezer het pad en/of de volledige naam van de afbeelding voor. Dat werkt anders in een <p> die de rol 'paragraaf' aangeeft ...

Verschillende rollen

Er bestaan verschillende rollen, waarvan een aantal al in deze documentatie werden besproken. Op de website van [W3C-ARIA](#) worden ze allemaal omstandig besproken. We gaan ze hier dus niet uitgebreid opnemen. Die rollen worden in categorieën gegroepeerd: Landmark, Widget, Pseudo, Document, Application, Presentation, Math, Definition, Note, Directory en Abstract. We verwijzen de lezer naar de officiële documentatie voor een gedetailleerde beschrijving.

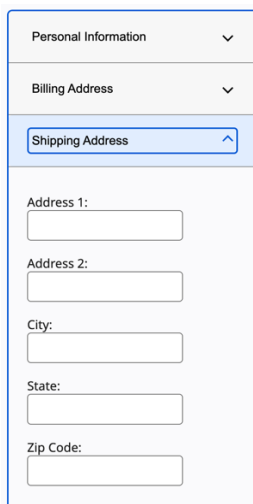
We willen wel nog preciseren dat elk element van de pagina een HTML-rol kan krijgen (pseudo html roles). Zo wordt de paragraaf <p role="heading" aria-level="1"> beschouwd als een kop van niveau 1 <h1>.

Bovendien hebben alle dynamische componenten (widgets) een specifieke rol die voldoet aan criteria waarmee rekening moet worden gehouden voor de interactie met hulptechnologieën. De 30 widget-rollen worden uitgebreid beschreven in de onlinedocumentatie van W3C-ARIA onder de naam [Design Patterns](#)

Voor elke component geeft de site de verplichte interacties met het toetsenbord en hulptechnologieën, voorbeelden van componenten, de broncode (HTML, CSS en JavaScript), de rollen (roles), toestanden (states) en eigenschappen (properties).

Voorbeeld van een dynamische component: Expand / Collapse (accordion)

Het pattern bestaat uit een wisselknop die verborgen (collapsed) inhoud blootlegt of die een op het scherm uitgerolde inhoud (expanding) verbergt. Die component wordt vaak gebruikt in formulieren of FAQ's om op de pagina niet teveel inhoud tegelijk weer te geven.



Via het toetsenbord moet de ENTER-toets of de spatiebalk het element openen of sluiten (focusable).

De schermlezer moet begrijpen dat het een knop betreft, de tekst van de knop lezen en zeggen of de inhoud ingeklapt op uitgeklaapt is (Collapse / Expand).

Als de technische richtlijnen goed worden nageleefd, kunnen alle browser-en-schermlezercombinaties met die component communiceren.

[Informatie over het design pattern accordion](#)

[Technische documentatie en voorbeeld van de accordion-component](#)

d) Waarde (value)

De eigenschappen en toestanden van de componenten, maar ook hun wijzigingen, moeten aan de hulptechnologieën worden meegedeeld.

In bepaalde gevallen verwijzen deze attributen naar de ID's van andere elementen (bijv. aria-labelledby="eenID"). In andere gevallen zijn de opties 'true' of 'false' (bijv. aria-hidden="true"). In nog andere gevallen kunnen de attributen beperkt blijven tot vooraf bepaalde opties (bijv. aria-orientation="vertical" of "horizontal").

Voor dynamische componenten moet de waarde in JavaScript worden gewijzigd op basis van gebruikersinteracties of bepaalde omstandigheden (bijv. aria-expanded="true" of "false").

Alle waarden, toestanden en eigenschappen worden beschreven [op de website van het W3C-ARIA](#)

e) Beschrijving (aria-describedby)

Het attribuut aria-describedby wordt gebruikt om aanvullende informatie bij een element mee te delen. In tegenstelling tot aria-label en aria-labelledby, wordt er geen rekening gehouden met aria-describedby voor de berekening van de toegankelijke naam. Dit attribuut kan met andere woorden niet worden gebruikt om een element een naam, een label of een titel te geven. Als een element slechts een toegankelijke naam zonder aanvullende beschrijving nodig heeft, mag het attribuut aria-describedby niet worden gebruikt.

Als het element een naam en een beschrijving heeft, lezen schermlezers eerst de naam en dan de beschrijving.

Als de tekst van aria-describedby kritiek (belangrijk) is, moet hij zichtbaar zijn voor zienden en moet hij in de context van het document voortdurend beschikbaar zijn op een manier die door schermlezers kan worden gelezen.

In het onderstaande voorbeeld wordt aanvullende informatie meegedeeld via aria-describedby.

Choose a new password: Minimum 8 characters, with both letters and numerals

```
<label for="newPassword">Choose a new password:</label>
```

```
<input type="password" id="newPassword" aria-describedby="pwdInfo">
```

```
<span id="pwdInfo">Minimum 8 characters, with both letters and numerals</span>
```

f) Live Regions

Aria-live-attributen lichten schermlezers in over wijzigingen van de pagina-inhoud, los van wat de gebruiker daar op het ogenblik van de wijziging doet. De wijzigingen kunnen worden veroorzaakt door de gebruiker, gebeuren in functie van een timer, het gevolg zijn van een serverprocedure ... Ze kunnen al dan niet zichtbaar zijn op het scherm en gaan meestal niet gepaard met een verplaatsing van de focus.

Technisch is het de bedoeling om een lege container aan te maken die een injectie in JavaScript verwacht. Zodra de inhoud in die region wordt ingelast, wordt de informatie doorgeschoven naar de schermlezer, die de wijziging meldt.

Goed voorbeeld: Melding via aria-live

Eerst wordt de container als een regio aangeduid. Hij moet leeg zijn bij het laden van de pagina of wanneer hij aan het DOM wordt toegevoegd (geen tekst tussen de openings- en sluitingstags). De meldingen gebeuren enkel bij wijzigingen in de live region. Als er al tekst aanwezig is vóór de wijziging, dan wordt die niet door de schermlezer voorgelezen.

```
<div aria-live="polite"></div>
```

Als de trigger in werking treedt, wordt de inhoud in de div ingelast en zegt de schermlezer 'Hello, screen reader user !'

```
<div aria-live="polite">Hello, screen reader user!</div>
```

Assertive / polite

De aria-live-melding kan op twee manieren gebeuren: assertief (assertive) of beleefd (polite).

De gebruiker krijgt niet de mogelijkheid om de meldingen opnieuw te beluisteren of te pauzeren. Ze moeten dus zo kort mogelijk zijn om te voorkomen dat de gebruiker ze onderbreekt en zo het risico loopt dat er belangrijke informatie verloren gaat.

Assertive: aria-live="assertive"

- De schermlezer onderbreekt zijn lopende tekst en leest de aankondiging voor;
- Na het voorlezen van het bericht, neemt de schermlezer zijn taak niet weer op waar hij werd onderbroken. De gebruiker kan de inhoud die hij aan het beluisteren was opnieuw lezen of naar een andere inhoud overstappen;
- De meldingen worden niet in een wachtrij geplaatst. Als meerdere meldingen op elkaar volgen, wordt de eerste door de volgende onderbroken zonder opnieuw te worden voorgelezen.

Polite: aria-live="polite"

- De melding wordt in een wachtrij geplaatst om te worden voorgelezen zodra de schermlezer zijn lopende tekst heeft afgewerkt.
- Als de gebruiker niet ingrijpt, worden de opeenvolgende meldingen één na één voorgelezen in de volgorde waarin ze verschijnen.

Attributen van aria-live-regions

Aria-atomic

- Aria-atomic="false" (standaard) leidt enkel tot het voorlezen van de gewijzigde inhoud (bijvoorbeeld belangrijk nieuws in een nieuwsfeed).
- Aria-atomic="true" leest opnieuw de hele regio waarin de wijziging gebeurde om de gebruiker meer context te geven.

Aria-relevant

- Aria-relevant="all": alle wijzigingen worden gemeld. Moet spaarzaam worden gebruikt om de ergonomie niet te schaden;
- Aria-relevant="additions": enkel toegevoegde knooppunten in de regio worden gemeld;
- Aria-relevant="removals": enkel gewiste knooppunten in de regio worden gemeld. Mag enkel worden gebruikt als de wijziging belangrijk is om het leescomfort te vrijwaren;
- Aria-relevant="text": enkel wijzigingen van de tekstuele context (met inbegrip van de tekstalternatieven in alt) worden gemeld.

Andere belangrijke soorten aria-live-regions

Role="alert"

Role="alert" is een specifiek type assertieve live-region dat wordt gebruikt om belangrijke informatie aan de gebruikers van een schermlezer te melden. De werking is dezelfde als voor aria-live="assertive", met toevoeging in sommige lezers van de melding 'waarschuwing'.

Het onderstaande voorbeeld toont de code die moet worden gebruikt voor een waarschuwingsbericht dat de gebruiker waarschuwt over de wijziging van zijn voorkeuren.

De oorspronkelijke HTML-code bevat een lege container die het waarschuwingsbericht zal ontvangen:

```
<form id="successForm" method="post" action="javascript:void(0)">
  <p><button>Save my preferences</button></p>
  <div class="msg" role="alert">
    <span class="msgTxt"></span>
  </div>
</form>
```

Le code final après injection du texte en JavaScript :

```
<form id="successForm" method="post" action="javascript:void(0)">
  <p><button>Save my preferences</button></p>
  <div class="msg" role="alert">
    <span class="msgTxt">Your preferences have been saved.</span>
```

</div>

</form>

Save my preferences



Andere soorten Live Region Roles worden beschreven [in de richtlijnen van het W3C](#)

g) Toetsenbordvriendelijkheid

Een website moet helemaal kunnen worden gebruikt zonder muis, enkel via de toetsen van het toetsenbord. Talrijke gebruikers met een handicap en gebruikers van hulptechnologieën kunnen de muis niet gebruiken of kunnen er niet goed mee werken.

De eerste test die op een website moet worden uitgevoerd is om uitsluitend via het toetsenbord te navigeren, met de tabtoets, de entertoets, de spatiebalk en de navigatiepijltjes. Zijn alle interactieve zones bereikbaar? Is de focus die wordt gegeven zichtbaar?

De volgende aanbevelingen moeten worden nageleefd:

Correct gebruik van WAI-ARIA design patterns

Elk [design pattern van dynamische componenten en widgets](#) vermeldt het toetsenbordgedrag dat de ontwerpers moeten voorzien.

Tabtoets tot aan de component, navigatiepijltjes in de component

De tabtoets moet toegang geven tot de component, waarbinnen vooral via de navigatiepijltjes wordt genavigeerd (ook al zijn andere toetsen mogelijk, naargelang de componenten).

De focus moet altijd zichtbaar zijn: schakel de outline nooit uit

Als standaardinstelling volstaat de outline voor elementen die de focus kunnen krijgen. Het is echter mogelijk om die in CSS te versterken, zowel voor het hover-effect als voor het actieve element en de verkregen focus.

Voorbeeld van outlineversterking in CSS:

```
a:focus, a:hover, a:active {
  background-color: #fdf6e7;
  outline: 1px solid #8cc63f;
}
```

Opgelet, zoals in het onderstaande voorbeeld schakelen CSS-resets de outline vaak uit. Controleer dit voor gebruik!

```
a:focus {
  outline: 0; /* or outline: none; */
}
```

Een logische en samenhangende tabvolgorde tussen het DOM en de schermweergave

De tabvolgorde moet zo logisch mogelijk zijn en de verschijningsvolgorde in het DOM volgen zonder de volgorde in CSS te wijzigen.

De gebruiker verwacht te beginnen in de linkerbovenhoek en zich dan te verplaatsen van links naar rechts tot in de rechterbenedenhoek.

Vermijd toetsenbordvallen

De gebruiker mag niet in een component vastlopen zonder eruit te kunnen komen.

Gebruik tabindex correct

[TabIndex="0" om een element via de tabtoets bereikbaar en focusseerbaar te maken](#)

Elementen zoals <p>, <hx>, <div> ... hoeven in principe niet door de tabtoets bereikt te worden. In bepaalde situaties kunnen die elementen echter aan de tabstroom van het toetsenbord worden toegevoegd.

- Mogelijkheid om een code te bereiken die uitsluitend via de muis kan worden bereikt omdat ze in JavaScript in plaats van systeemeigen HTML werd gegenereerd;
- Mogelijkheid om Aria/JavaScript-componenten te bereiken, zoals bijvoorbeeld het tab panel (<li role="tab" tabindex="0");
- De schermlezer dwingen om in een formulier tekst te lezen die anders zou kunnen worden overgeslagen.

[TabIndex="-1" om een element focusseerbaar maar niet via de tabtoets bereikbaar te maken](#)

Deze parameter laat niet toe om naar het element te tabben, maar wel om de focus te geven. Die techniek kan in verschillende situaties nuttig zijn:

- De focus verplaatsen naar een boodschap (bijvoorbeeld een foutmelding) om ervoor te zorgen dat zowel ziende personen als gebruikers van schermlezers die zien en horen;
- De focus verplaatsen naar een component in JavaScript;
- In een skip-link. Sommige browsers, zoals Safari, vereisen een bestemming die een systeemeigen focus kan krijgen (link, knop ...) of een element dat een tabindexwaarde bevat. Zoniet scrollt de viewport naar de gewenste positie, maar plaatst de volgende tab de focus op het element dat meteen na de skip-link volgt. Door aan de bestemming een tabindex="-1" toe te voegen, wordt de focus op dat element geplaatst zonder de eventueel op de pagina aangebrachte tabvolgorde te verstoren.

[Gebruik nooit een positieve tabindex](#)

Een element met een positieve tabindex krijgt de focus als eerste. De normale stroom op de pagina wordt daardoor volledig verstoord en de dynamische inhoud zal misschien niet meer correct functioneren.

S. Afbeeldingen

Elke niet-tekstuele inhoud vereist een tekstalternatief. De gebruikers van schermlezers kunnen een afbeelding niet lezen, slechtziende mensen moeten kunnen (uit)zoomen, de kleur aanpassen, de betekenis van de afbeeldingen wordt niet altijd goed begrepen ...

Zowel afbeeldingen als knoppen, video's, audiobestanden ... moeten een tekstalternatief of -vertaling aanbieden die iedereen kan lezen en aan zijn behoeften kan aanpassen.

Alternatieven voor afbeeldingen

Niet-decoratieve afbeeldingen moeten over een technisch gekoppeld, expliciet en beknopt alternatief beschikken (zeker niet meer dan 150 tekens).

Als de afbeelding veel uitleg vereist (zoals bijvoorbeeld een infographic), moet het alternatief niet noodzakelijk aan de afbeelding worden gekoppeld, maar moet het wel op de pagina aanwezig zijn. De afbeelding wordt dan een decoratief element zonder tekstalternatief.

Het is de bedoeling om de mogelijkheid te bieden om dezelfde informatie te verstrekken aan blinde en ziende gebruikers, zonder het (voor)lezen te besmetten met overvloedige of redundante informatie.

a) Informatieve afbeeldingen

Informatieve afbeeldingen voegen onzichtbare informatie toe aan de pagina. Als de afbeelding wordt verwijderd, verliezen we het begrip van de hele pagina of van een deel ervan.

Afbeeldingen die inhoud toevoegen, moeten een technisch gekoppeld expliciet alternatief bevatten

Schermlezers kunnen de betekenis van de afbeelding (nog) niet zelf afleiden. Ze lezen de inhoud van het alt-attribuut van de -tag.

Goed voorbeeld: Afbeelding met een alternatief in het alt-attribuut

Uitgaande van de veronderstelling dat de volgende afbeelding in haar context noodzakelijke informatie voor het begrip van de pagina bijdraagt, kan het alternatief luiden: 'A Singer model antique sewing machine'.

Als hij langs de afbeelding loopt, zal de schermlezer zeggen: 'Graphic (of afbeelding), A Singer model antique sewing machine'.

De gebruiker weet dan dat het over een afbeelding gaat (het is niet nodig om hem daar in de tekst aan te herinneren) en hoort het tekstalternatief.




```

```

Slecht voorbeeld: Afbeelding zonder alt-attribuut

Opgelet, als de afbeelding geen alt-attribuut bevat, probeert de tekstlezer het alternatief af te leiden door de naam van de afbeelding en/of haar pad voor te lezen. Dat is niet comfortabel voor de luisteraar en kan tot misverstanden leiden. In het onderstaande geval zal de schermlezer zeggen: 'Afbeelding, sewmach2 dot J P G'.

```

```

Het alternatief voor de afbeelding moet expliciet zijn en gekoppeld aan haar context en haar nut op de pagina waarin ze zich bevindt.

De volgende punten moeten in acht worden genomen:

- Het alternatief voor dezelfde afbeelding kan variëren volgens de context. Bijvoorbeeld voor een logo: de merknaam vermelden of, bij een afbeelding met een hyperlink naar de homepage van de website, vermelden: 'homepage van de website';
- Het alternatief mag niet algemeen zijn. Bijvoorbeeld voor de afbeelding van een stempel die een informatie bevestigt, is het alternatief niet 'stempel', maar 'bevestigd';
- Gebruik niet de woorden 'afbeelding van ...' of 'foto van ...' omdat de rol van de afbeelding duidelijk wordt herkend door de schermlezer, die deze informatie zal vermelden vóór hij het alternatief voorleest;
- Maximum 150 tekens;
- Maak gebruik van leestekens;
- Vermeld geen sleutelwoorden voor zoekmachines (SEO);
- Vermeld in het alt-attribuut geen informatie die voor alle internetgebruikers zichtbaar moet zijn. Bijvoorbeeld: een copyright, de auteur van een foto, een bijschrift ...

b) Decoratieve of redundante afbeeldingen

Een decoratieve of redundante afbeelding voegt geen aanvullende informatie toe aan de inhoud van de pagina. Dat betreft:

- Afbeeldingen toegevoegd voor de layout van de pagina, om 'sfeer' te scheppen, om de aandacht te trekken ...
- Afbeeldingen die inhoud toevoegen die al op de pagina aanwezig is (bijvoorbeeld: een grafiek die een datatabel met correcte HTML-structuur illustreert).

Om te beslissen of een afbeelding in die categorie valt, moeten we ons de pagina zonder de afbeelding voorstellen en bekijken of we de hele pagina nog steeds begrijpen zonder informatie te verliezen. In de praktijk blijkt dat afbeeldingen vaak decoratief zijn en dus geen tekstalternatief vereisen.

Decoratieve afbeeldingen of afbeeldingen die redundante informatie op de pagina aanbrengen moeten een leeg alternatief alt= "", een attribuut aria-hidden="true", of een aria-rol role="presentation" hebben of als achtergrond in CSS worden geïmplementeerd.

Het is de bedoeling om gebruikers van schermlezers de mogelijkheid te bieden om zo snel mogelijk te lezen zonder met nutteloze informatie te worden overstelpt.

Met deze technieken vertellen de meeste schermlezers zelfs niet aan hun gebruiker dat de afbeelding op de pagina aanwezig is (en dat is ook de bedoeling).

Vergeet niet om het lege alt-attribuut alt="" echt toe te voegen. Zonder dat attribuut probeert de schermlezer het alternatief af te leiden door de naam en/of het pad van de afbeelding te lezen.

Goed voorbeeld: Redundante afbeelding met leeg alt-attribuut

In dit voorbeeld bevat de link naar de homepage een afbeelding en de expliciete tekst 'Home Page'. Het alternatief voor de afbeelding moet dus leeg zijn.

Als de link echter de tekst 'Home Page' niet bevat, moet het alternatief voor de afbeelding alt="Home Page" zijn.



```
<a href="https://eqla.be">
```

```

```

Home Page

```
</a>
```

c) Activeerbare afbeeldingen (links, knoppen, bedieningselementen ...)

Afbeeldingen die dienst doen als links, knoppen of andere bedieningselementen moeten een alternatief hebben dat voldoet aan dezelfde criteria als beschreven voor de vorige soorten afbeeldingen.

Vermijd dus alternatieven zoals 'link naar ...', 'Eqla-logo' ... in plaats van 'Terug naar de Home Page', lege alternatieven ...

Formulierknoppen van het type input type="image" moeten aan dezelfde voorschriften voldoen en een alternatief hebben dat de naam op de afbeelding herneemt.



```
<input type="image" name="submit" src="submit-button.png" alt="Submit">
```

Het alternatief moet expliciet zijn over zijn functie en bestemming.

d) Complexe afbeeldingen

Complexe afbeeldingen zijn afbeeldingen die niet expliciet en volledig kunnen worden beschreven in minder dan 150 tekens.

Complexe afbeeldingen moeten kort worden beschreven via het alt-attribuut EN moeten een volledige beschrijving aanbieden.

Er bestaan verschillende technieken om een lange beschrijving aan een complexe afbeelding toe te voegen. Naargelang de functie van de website, de huisstijl, de ruimte op de pagina ... kan een van de onderstaande technieken worden gebruikt:

- De volledige beschrijving geven in de tekst van de pagina zelf (naargelang de gevallen mag het alt-attribuut leeg blijven);
- Onder het beeld een knop voorzien die een regio met de lange beschrijving toont en verbergt (accordion);
- Een knop voorzien die een dialoogvenster met de lange beschrijving opent;
- Een link voorzien naar de lange beschrijving in HTML op een andere pagina;

- De afbeelding inlassen tussen <figure>-tags waarbij de beschrijving wordt opgenomen in de <figcaption>-tag tussen de openings- en sluitingstags van <figure>.

Links, knoppen ... moeten voor iedereen zichtbaar, expliciet en technisch gekoppeld zijn.

[Goed voorbeeld: Lange beschrijving met aria-describedby in de context van het document](#)

In dit voorbeeld verschijnt de afbeelding en wordt het tekstalternatief voorgelezen (dat mag ook leeg zijn als de tekst elders wordt opgenomen), waarna ook de volledige beschrijving 'Last year ...', gekoppeld aan de afbeelding, wordt voorgelezen.

```
<h1> Proportion of Josephine's Mealtime Squirrel Sightings, by Month</h1>
```

```

```

```
<div id="description-extended">
```

```
  <p>Last year, Josephine kept track of the number of times...</p>
```

```
  <ul>
```

```
    <li>January: 14%</li>
```

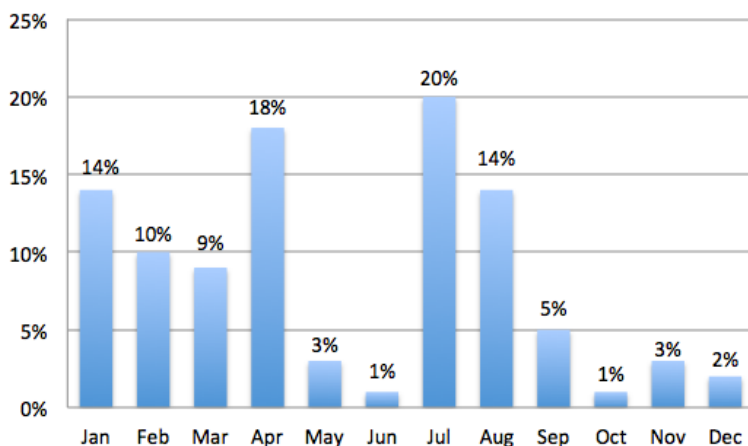
```
    <li>February: 10%</li>
```

```
    <li>...</li>
```

```
  </ul>...
```

```
</div>
```

Proportion of Josephine's Mealtime Squirrel Sightings, by Month



Last year, Josephine kept track of the number of times...

- January: 14%
- February: 10%

e) Tekstafbeeldingen

Een tekstafbeelding geeft tekst weer in de vorm van een afbeelding. Ofwel stelt de afbeelding enkel de tekst voor (bijvoorbeeld een logo), ofwel wordt de tekst in beeldvorm toegevoegd aan een afbeelding of een achtergrond.

Dergelijke beelden zouden slechts mogen worden gebruikt voor logo's of voor een gelijkaardig gebruik.

Ze zijn niet leesbaar voor schermlezers en dus moet er een tekstalternatief worden voorzien. Bovendien lopen ook mensen die op hun scherm moeten (uit)zoomen door de pixelvorming (behalve bij SVG-afbeeldingen) het risico om de tekst minder goed te begrijpen. Kleurenblinden, die de kleur van de tekst moeten aanpassen, of gebruikers met dyslexie die liever hun specifieke lettertypes gebruiken, kunnen dat niet doen.

f) Map-afbeeldingen

Map-afbeeldingen kunnen toegankelijk worden gemaakt, maar ze moeten voldoen aan een reeks criteria die vaak al voor de andere afbeeldingstypes werden besproken.

- De alternatieven moeten technisch gekoppeld zijn en de afbeelding voldoende en expliciet beschrijven;
- De alternatieven mogen niet meer dan 150 lettertekens lang zijn;
- De alternatieven van de <area>-tags moeten allemaal aan dezelfde voorschriften als hierboven voldoen;
- ...

Goed voorbeeld van map-afbeeldingen: het zonnestelsel waarin elke planeet naar een andere website verwijst

```

```

```
<map name="Map3">
```

```
<area shape="rect" coords="115,158,276,192"
```

```
href="http://en.wikipedia.org/wiki/Mercury_%28planet%29" target="_blank" alt="Mercury (Wikipedia article)">
```

```
<area shape="rect" coords="115,193,276,234"
```

```
href="http://en.wikipedia.org/wiki/Venus" target="_blank" alt="Venus (Wikipedia article)">
```

```
<area shape="rect" coords="118,235,273,280"
```

```
href="http://en.wikipedia.org/wiki/Earth" target="_blank" alt="Earth (Wikipedia article)">
```

```
<area shape="rect" coords="119,280,272,323"
```

```
href="http://en.wikipedia.org/wiki/Mars" target="_blank" alt="Mars (Wikipedia article)">
```

```
<area shape="rect" coords="119,324,322,455"
```

```
href="http://en.wikipedia.org/wiki/Jupiter" target="_blank" alt="Jupiter (Wikipedia article)">
```

```
<area shape="rect" coords="118,457,352,605"
```

```
href="http://en.wikipedia.org/wiki/Saturn" target="_blank" alt="Saturn (Wikipedia article)">
```

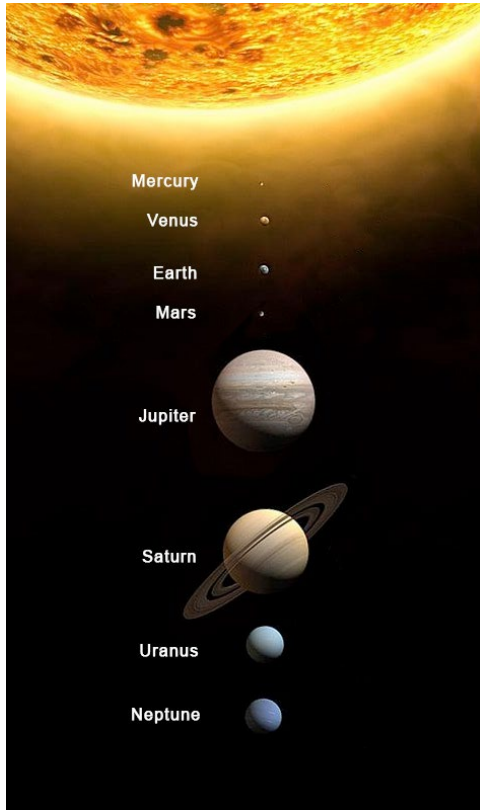
```
<area shape="rect" coords="119,606,308,666"
```

```
href="http://en.wikipedia.org/wiki/Uranus" target="_blank" alt="Uranus (Wikipedia article)">
```

```
<area shape="rect" coords="117,664,305,732"
```

```
href="http://en.wikipedia.org/wiki/Neptune" target="_blank" alt="Neptune (Wikipedia article)">
```

</map>



SVG-afbeeldingen

Alle informatie en meer bepaald de informatie over de toegankelijkheid van SVG is online beschikbaar, [met name op css-tricks.com](http://met.name.op.css-tricks.com)

Het SVG-formaat is interessant in verband met toegankelijkheid door zijn lage gewicht en zijn vermogen om te worden uitvergroot zonder kwaliteitsverlies. Mensen die op hun scherm moeten inzoomen kunnen dus steeds op dezelfde beeldkwaliteit rekenen.

Zoals voor alle soorten afbeeldingen moeten bepaalde toegankelijkheidscriteria in acht worden genomen om ervoor te zorgen dat de informatie aan iedereen kan worden meegedeeld.

Er bestaan verschillende manieren om SVG in een webpagina te integreren:

- Het gebruik van de -tag met vermelding van de bron. Bijvoorbeeld: ;
- SVG rechtstreeks (inline) in de HTML-broncode integreren met het <svg>-element;
- SVG integreren tussen <iframe>- of <embed>-tags of via een verwijzing als attribuut in een <object>.

Voor de toegankelijkheid geven we de voorkeur aan de twee eerste methodes omdat schermlezers fouten vertonen bij het voorlezen van SVG's in <iframe> en <object>.

a) SVG's geïntegreerd via -tags moeten een rol="img" hebben en (behalve voor decoratieve elementen) een expliciet en beknopt alternatief via alt, aria-label of aria-labelledby

De role="img" zorgt ervoor dat de schermlezer de afbeelding als dusdanig herkent en we komen hier niet terug op het belang van alternatieven.

Goede voorbeelden van SVG geïmplementeerd met de -tag en een goed alternatief

```

```

```

```

```
<p id="caption1">This is a caption above an image</p>
```

```

```

b) Verplichte toegankelijkheidsaspecten voor inline geïntegreerde SVG's met de <svg>-tag

- De <svg>-tag moet de role="img" bevatten.
- Informatieve of activeerbare SVG's moeten een tekstalternatief hebben in een <title>-element dat het eerste onderliggende knooppunt van de SVG moet zijn.
- Het alternatief in de <title> moet technisch worden gekoppeld aan het <svg>-element via aria-labelledby (voor een maximale compatibiliteit met alle schermlezers)
- Alle teksten in de afbeelding die door de schermlezer moeten worden voorgelezen, moeten via aria-labelledby aan het <svg>-element worden gekoppeld.

Voorbeeld van een uittreksel van een inline SVG-afbeelding met al die elementen

```
<svg role="img" aria-labelledby="title desc jan feb mar apr may jun jul aug sep oct nov dec">
```

```
<title id="title">Total Widgets Purchased during 2016</title>
```

```
<desc id="desc">
```

The graph displays the total number of widgets purchased from The ABC Store during 2016, displayed by month.

```
</desc>
```

...

```
<g id="jan" class="bar labels x-labels">
```

```
<rect x="25" y="195" width="33" height="45" fill="#111"></rect>
```

```
<text x="32" y="260" fill="#000">Jan.</text>
```

```
<text x="33" y="220" fill="#fff">45</text>
```

```
</g>
```

```
<g id="feb" class="bar labels x-labels">
```

```
<rect x="62" y="160" width="33" height="80" fill="#111"></rect>
```

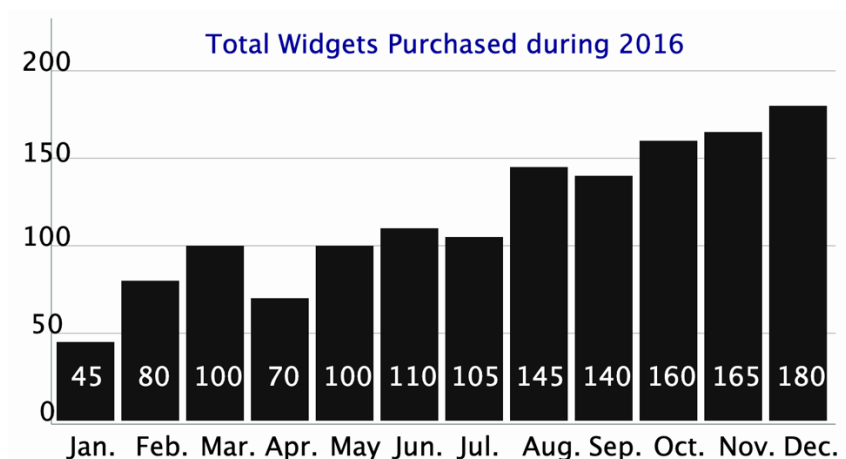
```
<text x="70" y="260" fill="#000" >Feb.</text>
```

```
<text x="71" y="220" fill="#fff">80</text>
```

```

</g>
<g id="mar" class="bar labels x-labels">
  <rect x="99" y="140" width="33" height="100" fill="#111"></rect>
  <text x="107" y="260" fill="#000" >Mar.</text>
  <text x="103" y="220" fill="#fff">100</text>
</g>
...
</svg>

```



Icon Fonts

a) Informatieve Icon Fonts

Icon Fonts zonder zichtbare tekst die informatie bijdragen, moeten een role="img" en een tekstalternatief hebben

Forms of payment accepted:



```
<p>Forms of payment accepted:<br>
```

```
  <span class="fa fa-cc-mastercard fa-2x" role="img" aria-label="MasterCard"></span>
```

```
  <span class="fa fa-cc-visa fa-2x" role="img" aria-label="Visa"></span>
```

```
  <span class="fa fa-cc-paypal fa-2x" role="img" aria-label="PayPal"></span>
```

```
</p>
```

Activeerbare Icon Fonts zonder zichtbare tekst moeten een tekstalternatief hebben

Goed voorbeeld: Knoppen in de vorm van Icon Fonts met een alternatief via aria-label en een role="img"



```
<p id="text-editor">
```

```

<button><span role="img" class="fa fa-bold" aria-label="Bold"></span></button>
<button><span role="img" class="fa fa-italic" aria-label="Italic"></span></button>
<button><span role="img" class="fa fa-underline" aria-label="Underline"></span></button>
<button><span role="img" class="fa fa-indent" aria-label="Indent"></span></button>
<button><span role="img" class="fa fa-outdent" aria-label="Outdent"></span></button>
<button><span role="img" class="fa fa-link" aria-label="Create link"></span></button>
<button><span role="img" class="fa fa-unlink" aria-label="Remove link"></span></button>
<button><span role="img" class="fa fa-list" aria-label="Bulleted list"></span></button>
<button><span role="img" class="fa fa-list-ol" aria-label="Numbered list"></span></button>
</p>

```

[Goed voorbeeld: Link in de vorm van Icon Fonts met een alternatief via aria-label](#)

In dit voorbeeld wordt het tekstalternatief aangebracht via het attribuut aria-label. Schermlezers behandelen het pictogram als een afbeelding en de link wordt overgeërfd van het alternatief van de Icon Font.



```

<p>
  <a href="#">
    <span class="fa fa-facebook-official fa-2x" aria-label="Our Facebook page">
    </span>
  </a>
</p>

```

[b\) Decoratieve of redundante Icon Fonts](#)

Dit begrip werd al uitgelegd in het [hoofdstuk over alternatieven voor afbeeldingen](#)

De betrokken Icon Fonts moeten voor de schermlezers verborgen blijven via het attribuut aria-hidden="true"

Goed voorbeeld: Een hulppictogram naast een hulptekst verbergen

[Help](#) 

```

<p>
  <a href="#">
    Help
    <span class="fa fa-question-circle" aria-hidden="true"></span>
  </a>

```


</p>

IV. Conclusie

Hoewel de Covid-crisis de digitalisering van de overheidsdiensten ongetwijfeld heeft versneld, heeft 40% van de Belgische bevolking het nog steeds moeilijk met de toenemende digitalisering van de samenleving⁶.

Om deze uitdaging aan te gaan en ervoor te zorgen dat gedigitaliseerde overheidsdiensten echt inclusief en toegankelijk zijn, heeft deze studie aangetoond dat zij niet uitsluitend digitaal mogen zijn.

Ondanks alle inspanningen zal een groot deel van de bevolking nog lange tijd menselijk contact nodig hebben om hun administratieve procedures uit te voeren. Zo blijkt immers dat *“bijna zes van de tien internetgebruikers tussen 16 en 74 jaar (58%) in het beste geval in beperkte mate gebruik maken van het internet om hun administratie te doen: 32% van hen stelt geen enkele digitale handeling, 26% slechts in beperkte mate”*.⁷

Een andere bevinding is dat de digitalisering van procedures de last van de ondersteuning van burgers verlegt naar maatschappelijk werkers, ondersteuningsdiensten en zelfs computerspecialisten (wier taak het niet is om mensen op te leiden om een ID-kaartlezer te gebruiken of om namens hun cliënten online bankoverschrijvingen te verrichten).

Voorts bleek uit de studie ook dat (voor overheidsdiensten verplichte) toegankelijkheidsnormen een solide maar ontoereikende basis vormen om rekening te houden met alle gebruikers van digitale diensten. Het is dan ook van essentieel belang dat de 38 aanbevelingen in dit verslag worden uitgevoerd en dat de platforms voor gedigitaliseerde diensten reeds in de ontwerpfase worden getest door een panel dat uit verschillende groepen bestaat. Deze aanbeveling sluit aan bij eerdere werkzaamheden die zijn beschreven in het verslag van Idéalic⁸ en in de brochure over digitale inclusie⁹ van de BOSA-catalogus.

Om deze aanbevelingen te implementeren zal aan de ontwerpers een technisch referentiekader ter beschikking worden gesteld om hun uitvoering te verzekeren.

Ten slotte moeten er procedures komen voor ondersteuning, opleiding en toezicht op de naleving van deze aanbevelingen door de projecten, om ervoor te zorgen dat bij de ontwikkeling van gewestelijke digitale instrumenten rekening wordt gehouden met alle doelgroepen.

⁶ Koning Boudewijnstichting, (2020). Barometer Digitale Inclusie 2020

⁷ Koning Boudewijnstichting, (2021). *Digitale Inclusie. Essentiële diensten online: toegankelijk voor iedereen?* blz. 24 Online via <https://kbs-frb.be/nl/digitale-inclusie-essentiele-diensten-online-toegankelijk-voor-iedereen>

⁸ Faure, L & Brotcorne, P (2021). *Guide pour une conception inclusive des services numériques*. Idéalic.be. p.20-24.

⁹ Digitale inclusie, wat is het? Online via <https://digitalopen.belgium.be/nl/playbook/tools/digitale-inclusie-brochure>

V. Checklist voor toegankelijke en inclusieve DOD's

- ✓ Aanbeveling 1: Naast de onlinediensten hebben de burgers de mogelijkheid om voor elke vraag naar een fysiek loket te gaan, waar een ambtenaar is die hen kan helpen met hun procedures
- ✓ Aanbeveling 2: De site voldoet aan de normen voor digitale toegankelijkheid en bevat een conforme toegankelijkheidsverklaring, opgesteld of gevalideerd door een bevoegde instantie
- ✓ Aanbeveling 3: Uniformiseer de lay-out van alle digitale overheidsdiensten
- ✓ Aanbeveling 4: De kleur van een tekst moet voldoende contrasteren met de achtergrondkleur
- ✓ Aanbeveling 5: Gebruik een toegankelijk lettertype en zorg ervoor dat de tekengrootte kan worden gewijzigd
- ✓ Aanbeveling 6: Informatie mag niet alleen door kleur of vorm worden verstrekt
- ✓ Aanbeveling 7: Pas het responsive design toe
- ✓ Aanbeveling 8: Zorg voor ten minste de volgende drie navigatiepaden: het navigatiemenu, de sitemap en een zoekmachine
- ✓ Aanbeveling 9: Voorzie een krachtige en spraakgestuurde zoekmachine
- ✓ Aanbeveling 10: Respecteer de normen voor het structureren van websites
- ✓ Aanbeveling 11: Voer skip-links in
- ✓ Aanbeveling 12: Geef informatie weer om duidelijk te maken waar men zich bevindt in de boomstructuur van de site
- ✓ Aanbeveling 13: De hele site moet toetsenbordvriendelijk zijn
- ✓ Aanbeveling 14: Gebruik expliciete termen die de actie van een element (link/knop) nauwkeurig beschrijven
- ✓ Aanbeveling 15: Beperk de lengte van de teksten en illustreer de inhoud met visuals
- ✓ Aanbeveling 16: Vertaal de site in het Frans, Nederlands en Engels
- ✓ Aanbeveling 17: De gebruiker moet voortdurend in dezelfde taal kunnen blijven navigeren
- ✓ Aanbeveling 18: Zet administratieve acties om in eenvoudige taal en in gebarentaal
- ✓ Aanbeveling 19: Voeg een tekstalternatief toe voor afbeeldingen die informatie bevatten
- ✓ Aanbeveling 20: Implementeer de aanmeldprocedure via CSAM
- ✓ Aanbeveling 21: Geef een bericht dat de status van aanmelden/afmelden bevestigt
- ✓ Aanbeveling 22: Geef een waarschuwingsbericht als een gebruiker een venster sluit zonder eerst af te melden
- ✓ Aanbeveling 23: Verleng de maximale duur voor de aanmeldprocedure
- ✓ Aanbeveling 24: Waarborg de toegankelijkheid van de formulieren
- ✓ Aanbeveling 25: Vraag alleen strikt noodzakelijke informatie
- ✓ Aanbeveling 26: Maak het mogelijk om persoonlijke gegevens manueel in te voeren en aan te passen
- ✓ Aanbeveling 27: Specificeer het formaat van de gevraagde informatie
- ✓ Aanbeveling 28: Geef bij een fout duidelijk aan welke informatie moet worden gewijzigd
- ✓ Aanbeveling 29: Geef een bericht dat de status van indiening van het formulier bevestigt
- ✓ Aanbeveling 30: Identificeer de aard en het gewicht van het downloadbare document
- ✓ Aanbeveling 31: Leg uit hoe men documenten kan scannen en uploaden
- ✓ Aanbeveling 32: Bied betaling aan met debetkaart (Bancontact) en kredietkaart (Visa en Mastercard)
- ✓ Aanbeveling 33: Geef een bericht dat de status van de onlinebetaling bevestigt

- ✓ Aanbeveling 34: Video- en geluidsbestanden moeten altijd voor iedereen toegankelijk zijn
- ✓ Aanbeveling 35: Afspraken moet men telefonisch en via een onlinekalender kunnen boeken
- ✓ Aanbeveling 36: Bied ten minste een toegankelijke e-mail of contactformulier en het telefoonnummer aan (aangevuld met vertolking op afstand in gebarentaal)
- ✓ Aanbeveling 37: Laat de DOD vanaf de ontwerpfase testen door een diverse groep burgers
- ✓ Aanbeveling 38: Maak een gebruiksaanwijzing voor de DOD's
- ✓ Aanbeveling 39: Geef professionals een opleiding in inclusieve aanbevelingen
- ✓ Aanbeveling 40: Stel een onlinetestplatform ter beschikking aan ondersteunings- en opleidingsinstanties waarmee gebruikers tijdens opleidingen onlinediensten kunnen uitproberen

Inclusieve aanbevelingen en technisch referentiekader om digitale overheidsdiensten te verbeteren en te optimaliseren voor mensen wie een digitale kloof dreigt

CAWaB voor Paradigm

