



Wit-Gele Kruis - witgelekruis.be

Accessibility audit

Client: Wit-Gele Kruis
Project: witgelekruis.be
Date: 09/05/2025

Table of contents

Table of contents	2
Introduction	5
Scope of the evaluation	5
Out of scope	5
Sample pages for review	6
Summary	7
Priorities	7
1. Keyboard	7
2. Visibility	7
3. Ajax	7
4. Feedback & forms	7
5. Content	7
Audit results	8
No Accessibility Statement page ('Toegankelijkheidsverklaring')	8
Success Criterion 1.1.1 Non-text content (minimum) (A)	9
Problem	9
Success criteria:	9
References:	10
Success Criterion 1.3.1 Info and Relationships (Minimum) (A)	11
Problem	11
Success criteria:	11
References:	12
Success Criterion 1.3.5 Identify Input Purpose (AA)	13
Problem	13
Success criteria:	13
References:	13
Success Criterion 1.4.3 Contrast (Minimum) (A)	15
Problem:	15
Success criteria:	15
References:	16
Success Criterion 1.4.10 Reflow (A)	17
Problem	17
Success criteria:	17
References:	18
Success Criterion 2.1.1 Keyboard (Minimum) (A)	19

Problem	19
Success criteria:	20
References:	20
Success Criterion 2.4.1 Bypass Blocks (Minimal) (A)	21
Problem	21
Success criteria:	21
References:	21
Success Criterion 2.4.4 Link purpose (in context) (A)	22
Problem	22
Success criteria:	22
References:	23
Success Criterion 2.4.7: Focus Visible (Minimum) (A)	24
Problem	24
Success criteria	24
References	25
Success Criterion 2.4.8 Location (AA)	26
Problem	26
Success criteria:	26
References:	26
Success Criterion 3.2.2 On Input (A)	27
Problem	27
Success criteria:	27
References:	27
Success Criterion 3.3.3 Error Suggestion (AA)	28
Problem	28
Success criteria:	28
References:	29
Success Criterion 4.1.2 Name, Role, Value (Minimum) (A)	30
Problem	30
Success criteria:	31
References:	31
WAI ARIA 1.1 - ARIA for non-semantic HTML	32
Problem	32
Solution	33
References:	36

Introduction

- This report describes the **conformance** of the submitted website with **W3C's Web Content Accessibility Guidelines (WCAG)**.
- A set of sample pages, for the submitted website or application, was tested and the WCAG Violations were documented with **code snippets and/or screenshots** when relevant.
- This report provides **advice** with **code suggestions** (not complete production code) and/or **links** to relevant resources
- The issues are **ordered by WCAG topic**
- We write **in English**, as most references and excerpts are from English-language WCAG documentation.
- To compose a **representative sample**, we base ourselves on a sampling of pages, aiming for at least 1 page per content type.
- Depending on the nature and complexity of the website, the **sample size** needed can vary, but we aim to include at least these:
 - The homepage,
 - A contact page (if it exists)
 - Representative forms
 - Basic page, eg. an 'about us'-page (or similar content page).
 - The aforementioned content types overview and a detail page

Scope of the evaluation

- **Website name:** Wit-Gele Kruis
- **Website base url:** <https://www.witgelekruis.be/>
- **WCAG Version:** 2.2
- **Conformance target:** AA
- **Additional evaluation requirements:** *not provided*

Out of scope

The following are out of scope for this review:

- Integrated external services (eg. Google maps, embedded video players, ...), as these should have their own documentation about accessibility. If they are not accessible, this should be noted in the Accessibility Statement page or an alternative found.
- Digital documents (eg. PDF's), as these are seen as part of content within responsibility of the client
- Logged-in sections (eg. admin area)
- Pages that are on a different domain (eg. <https://nieuwsbrieven.witgelekruis.be/>)

Sample pages for review

See the audit results for links and context

Summary

Based on this evaluation, the website does **not** meet WCAG 2.2, Conformance Level AA. Detailed review results are available below.

Priorities

We recommend first focussing on elements of higher impact. These often center around keyboard navigation, Ajax and visibility. This list orders the priorities from higher to lower.

1. Keyboard

1. [Success Criterion 2.1.1 Keyboard \(Minimum\) \(A\)](#)
2. [Success Criterion 2.4.7: Focus Visible \(Minimum\) \(A\)](#)

2. Visibility

1. [Success Criterion 1.4.3 Contrast \(Minimum\) \(A\)](#)
2. [Success Criterion 1.4.10 Reflow \(A\)](#)

3. Ajax

1. [Success Criterion 3.2.2 On Input \(A\)](#)
2. [WAI ARIA 1.1 - ARIA for non-semantic HTML](#)

4. Feedback & forms

1. [Success Criterion 3.3.3 Error Suggestion \(AA\)](#)
2. [Success Criterion 4.1.2 Name, Role, Value \(Minimum\) \(A\)](#)

5. Content

1. [Success Criterion 2.4.4 Link purpose \(in context\) \(A\)](#)
2. [Success Criterion 1.3.1 Info and Relationships \(Minimum\) \(A\)](#)
3. [Success Criterion 1.3.5 Identify Input Purpose \(AA\)](#)
4. [Success Criterion 1.1.1 Non-text content \(minimum\) \(A\)](#)
5. [Success Criterion 2.4.1: Bypass Blocks](#)
6. [Success Criterion 2.4.8 Location \(AA\)](#)

Audit results

No Accessibility Statement page ('Toegankelijkheidsverklaring')

Public bodies in countries that implement the EU Web Accessibility Directive are required to provide an accessibility statement.

Accessibility statements are important for several reasons:

- Show your users that you care about accessibility and about them
- Provide them with information about the accessibility of your content
- Demonstrate commitment to accessibility, and to social responsibility

Success Criterion 1.1.1 Non-text content (minimum) (A)

All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for the situations listed below.

Problem

- **Logo:** it's an inline SVG that functions as an image, but there's no label for it
- **Teaser images:**
 - Teaser images of `Het Wit-Gele nieuws` section don't have an alt-attribute. Even if it is a decorative image, there should be an alt-attribute on the img tag (leave the value empty for decorative images)
This is a recurring issue with teasers across the site.
[Here is some info about what decorative images are](#)

- **[Tips en zorgverhalen](#)**

- There is an SVG used to provide a logo on the 'Powered by' element. This essentially functions to name the company and so it needs a label and since it behaves as an image it should also have the right role



- **[Info voor zorgpartners:](#)**

- When picking 'Antwerpen' as region, you will find there is an image of a map at the bottom of the page. It contains a list of people and their contact details. First off, it's blurry and barely readable for people with no visual impairments, secondly the info is not detectable by screen readers.

Success criteria:

- **Logo:** we recommend adding a title tag inside of the SVG itself, with an id on it:
Eg. `<title id="website-logo-title">Logo of Wit-Gele Kruis</title>`.
On the SVG itself, you should add the attribute:
`aria-labelledby="website-logo-title"`
And thirdly: Add attribute `role="img"` attribute on the SVG
[You can read more about making SVG's accessible over here.](#)
- **Teaser images:** make sure that img-tag properties, also contain an alt-attribute. You can also get rid of the title-attribute, it is not used and can cause issues for screen readers.
- **[Tips en zorgverhalen:](#)** Similarly to the site logo
 - add a `<title>` tag inside of an inline SVG
 - add an `aria-labelledby` attribute on the SVG itself to point to the title ID
 - add `role="img"` attribute
- **[Info voor zorgpartners:](#)**

- Display the textual information in the page, in a way that is readable by screen readers. Eg. a visually hidden box that lists the people and their contact info. The names of the areas are less important, as there is no relevant info about those places. But also, make sure there is a sharp image of that contact info that is clear for everyone looking at it. This one is too blurry.

References:

- **Source:** <https://www.w3.org/TR/WCAG22/#non-text-content>
- **Understanding:** <https://www.w3.org/WAI/WCAG22/Understanding/non-text-content.html>
- **Techniques:**
 - <https://www.w3.org/WAI/WCAG21/quickref/#non-text-content>
 - <https://www.w3.org/WAI/tutorials/images/decorative/>

Success Criterion 1.3.1 Info and Relationships (Minimum) (A)

Instructions provided for understanding and operating content do not rely solely on sensory characteristics of components such as shape, color, size, visual location, orientation, or sound.

Problem

- **Footer navigation**
 - I would recommend excluding the 'doormat' portion (with the lists of links) from the footer itself and making it an aside instead. The page footer should contain the info relevant to the page or the site itself. The blocks are essentially 'related content' like a site navigation, which belongs in its own sectioning element.
 - The headings for the lists of links in the footer (aka 'doormat'), should not be an H2. This messes up the outline of the page used by screen readers, as it makes these blocks an equal weight as page content or the blocks of page content, that also use H2. For the same reason, this also messes with SEO of the site.
- **Homepage:**
 - 'Onze Diensten'
 - It has a list of link-blocks. But they the markup is using articles instead of list markup
 - You should not have multiple H1 tags on a page because it messes up SEO as well as Screen readers' ability to produce an outline of headings for the page.
- **[Praktische informatie:](#)**
 - 'Welke afspraken maken we met jou?'
 - Teaser cards appear which all use <h2> headings. This doesn't really make sense, as the title of the section is also an <h2>. This probably happens in other pages as well, if they use the same component.

Success criteria:

- **Footer navigation**
 - Make an <aside> element above the footer, to put all the doormat elements into. Give it the attribute: `role="complementary"`
 - The headings for the lists of links for the 'doormat', should be H3 or lower.
- **Homepage:**
 - Onze Diensten:
 - The article tag is meant to be used for for independent, self-contained content, such as blog or news posts. Not to display just 1 link with an image, or even description. So use an unordered list: with list items: and style the text to look like you need it to look (add divs or spans as needed)
 - For SEO and to maintain a proper overview for screen readers, there should only be 1 H1 and that is for the page title. And since there is no actual content besides a link, it doesn't make sense to use headings here anyway. So just use links without a heading.

- **Praktische informatie:**
 - 'Welke afspraken maken we met jou?'
 - The boxes should have a lower heading, such as <h3> instead

References:

- **Source:** <https://www.w3.org/TR/WCAG21/#info-and-relationships>
- **Understanding:** <https://www.w3.org/WAI/WCAG21/Understanding/info-and-relationships.html>
- **Techniques:** <https://www.w3.org/WAI/WCAG21/quickref/#info-and-relationships>

Success Criterion 1.3.5 Identify Input Purpose (AA)

Some people with cognitive disabilities may not understand the input's purpose from the label alone, so it's important to it easier to fill in forms. We can use code to indicate the purpose of common inputs.

Problem

- **Any** pages with forms: if it needs data from the user (an email, name, address, ...) should allow for autofill.
 - Eg. [sidebar form in various pages like this](#)
 - Eg. [the jobalert page form](#)
 - ...

Success criteria:

- **Any** pages with forms: give the user fields an autocomplete property with the proper value ([see here for reference](#)). That goes for email, name, addresses fields and phone. This is not needed for search forms, as there is no personal info to submit.

References:

- **Source:** <https://www.w3.org/TR/WCAG22/#identify-input-purpose>
- **Understanding:** <https://www.w3.org/WAI/WCAG22/Understanding/identify-input-purpose.html>
- **Techniques:**
 - <https://www.w3.org/WAI/WCAG22/Techniques/html/H98.html>
 - <https://html.spec.whatwg.org/multipage/form-control-infrastructure.html#autofill-field>

Success Criterion 1.4.3 Contrast (Minimum) (A)

The visual presentation of text, and images of text, needs to have a contrast ratio of at least **4.5:1**, **except** for the following:

- **Large Text:** text that is at least **24px** and images of large-scale text have a contrast ratio of at least **3:1**
- **Bold text:** at least **18.5 pixels**, the contrast is at least **3:1**
- **UI components** need a contrast of at least **3:1** (eg. think of form field borders)
 - User interface components that you need to see to understand a function, are parts of sub-menus, edit buttons, tabs, or thumbnails of media.
- **Incidental:** Text or images of text that are part of an inactive user interface component, that are pure decoration, that are not visible to anyone, or that are part of a picture that contains significant other visual content, have no contrast requirement.
- **Logotypes:** Text that is part of a logo or brand name has no contrast requirement.
- [Tips en zorgverhalen](#) & [Stages](#): The contrast of the province name is too low (yellow text on white background).

Problem:

A variety of colors don't follow the contrast rules. For example:

- **Homepage:**
 - Yellow arrows on a white background is too low contrast. They are part of the UI (not decorative), it needs a contrast of 3:1
 - eg. 'Het Wit-Gele nieuws' - the 'Lees meer' arrows
 - Eg. the links for jobs in [Vacatures op jouw maat](#)
- There are various yellow icons on a white background, like map and distance icons. Since they are a UI element (they convey meaning, are not just decorative), they should be 3:1
 - Eg. map and distance icons on the [jobs page teasers](#)
 - Eg. phone, fax and mail icons [in sidebar of certain pages](#)
 - Sidenote: the grey icons of a clock and such ARE decorative, they have no contrast requirements. The text is clear on its own.
- Yellow bullets on a white or grey background: unordered lists [on various pages](#): make them darker so they have 3:1 contrast, as they are UI elements (you need to see them to know this is a list)
- The placeholder text in inputs is too low contrast (grey on white), it should be 4.5:1, this will be the case for any input field with placeholder text.
 - Eg. [Filtered jobs](#): the text field for a postal code, in the sidebar
- [Stages](#): The contrast of the province name. Font-size is larger than 18.5px, so make the color #B88C00 to get 3:1 contrast

Success criteria:

Ensure all these elements have sufficient color contrast between the text in the foreground and background color behind it.

Contrast should be checked directly [using a color contrast checker like this one](#) instead of relying on a testing suite or Chrome/Firefox devtools contrast checks as they don't measure correctly.

References:

- **Source:** <https://www.w3.org/TR/WCAG22/#contrast-minimum>
- **Understanding:** <https://www.w3.org/WAI/WCAG22/Understanding/contrast-minimum.html>
- **Techniques:**
 - <https://www.w3.org/WAI/WCAG22/quickref/?showtechniques=143#contrast-minimum>
 - <https://www.w3.org/WAI/WCAG22/Techniques/general/G207>

Success Criterion 1.4.10 Reflow (A)

Content can be presented without loss of information or functionality, and without requiring scrolling in two dimensions for:

- Vertical scrolling content at a width equivalent to 320 CSS pixels;
- Horizontal scrolling content at a height equivalent to 256 CSS pixels.

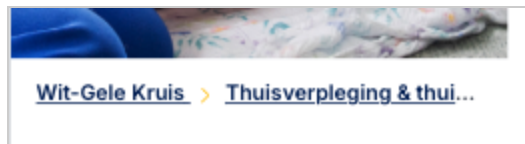
This means, text should not be clipped when the page is resized. Notably, for a website which is designed to scroll vertically (as a majority of websites are), at minimum 320px wide, you should be able to read the text without having to scroll horizontally. The text should also not be clipped (cut off) horizontally.

For horizontal designs, the opposite is true. Then we use 256px height as the reference.

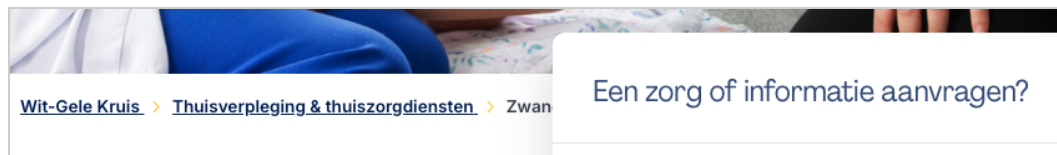
There is an exception, when the site is designed to scroll in 2 dimensions.

Problem

- **Breadcrumbs:**
 - On mobile screens, the second breadcrumb item is clipped and displayed with an ellipsis. That makes this item, and any subsequent items, no longer readable.



- [On pages with 3 breadcrumb items](#), at a desktop size (1024px or wider), there is a sidebar that overlaps part of the breadcrumb. So the text is clipped here as well.

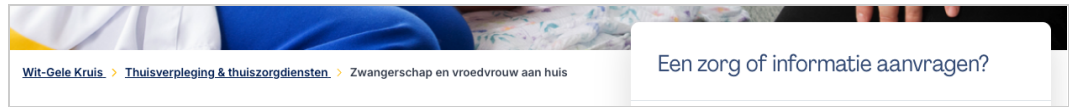


Success criteria:

- **Breadcrumbs:**
 - For mobile screens, you could render all breadcrumb items underneath each other. Also make sure that multi-word items can reflow into new lines



- For the desktop, the overlap could be fixed by removing the max-width limitation and adding some padding on the right, to match the width of the sidebar (500px + a bit of extra space). But only when a sidebar is present + at 1024px screen size and wider.



References:

- **Source:** <https://www.w3.org/TR/WCAG22/#reflow>
- **Understanding:** <https://www.w3.org/WAI/WCAG22/Understanding/reflow.html>
- **Techniques:** <https://www.w3.org/WAI/WCAG22/quickref/#reflow>

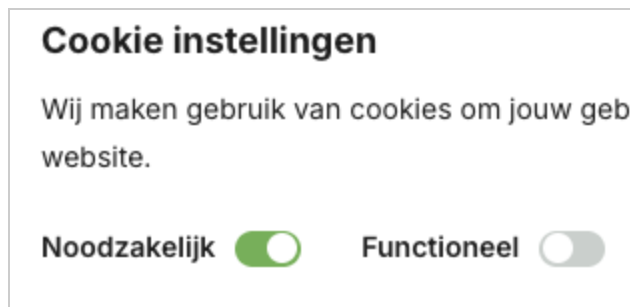
Success Criterion 2.1.1 Keyboard (Minimum) (A)

All functionality of the content is operable through a keyboard interface without requiring specific timings for individual keystrokes, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints.

Problem

- **Language menu** and various other items in the header, can not be operated using the keyboard
- In **mobile** view, the main **menu** is impossible to access or navigate. Screen readers depend on the same logic as keyboard navigation so this is not usable.
- **Cookie banner:**
 - The first option ('Noodzakelijk') is visualized as an active toggle element (green for the 'on' state) but can not be operated (not by keyboard, not even by mouse). This is because it is a disabled option.

If it is not meant to be operated, it should be styled as a disabled option.



- Once you pick your **cookie preferences**, there is also a 'Cookie-instellingen' button (looks like a cookie) fixed to the bottom of the page. This is also not reachable by keyboard. Or if it is, there's no way to determine visually because no focus style to indicate it is.



- **Teasers**
 - [Tips en zorgverhalen](#): In the '**Meest recent**' section, there are teasers with links that have aria-hidden attributes. You should not set these on interactive elements. We're guessing the purpose is to only have 1 link per teaser be focusable, when multiple of the same link exist (eg. on the title, a 'read more' and/or other) but this is a bad practice and possible problem for screen readers.
- **Location component** (seen in '**Waar kunnen we je helpen?**' on homepage, but also other places) can not be accessed and used by keyboard.

Success criteria:

- **Language menu** and other elements in the header: make it accessible via keyboard
- **For mobile menu:**
 - Make sure you can tab and focus to the toggle button (hamburger) for the menu, and operate it using the Enter (submit)
 - **Focus should remain visible** at all times, so the user knows where they are
 - There should be **visual indicators for submenus** so the user can use those to open/close a submenu
- **Cookie banner:**
 - Style the first option ('Noodzakelijk') in a disabled style, that is visually distinct from the other toggles.
Since the 'off' state of the other toggles is already a light grey, we propose to use this grey style for disabled toggles, and make the 'off' state black or a dark color. This is to avoid confusion between the disabled and the 'off' state of a toggle.
 - When the banner is closed, make sure the button can be reached by keyboard (and has a focus style).
- **Teasers:**
 - [Tips en zorgverhalen](#): In the '**Meest recent**' section: remove link you don't want to be visible to screen readers, in a teaser. Just keep the one on the title and use the [pseudo-element technique](#) to make the card clickable: use a pseudo-element of the title link to cover the entire card in an invisible way. Turn the 'lees meer' into a span with attribute `aria-hidden="true"`.
 - Also make sure the focus state works (a repeated issue with links throughout the site)
- **Location component** (seen in '**Waar kunnen we je helpen?**' on homepage).
 - Specifically, you should be able to focus on the location or postal code and if suggestions are displayed, you need to be able to navigate and choose them using keyboard (eg. arrow keys to change and Enter to confirm).
[The listbox pattern](#) should give you insights into the behaviour we would expect from such a list.

References:

- **Source:** <https://www.w3.org/TR/WCAG21/#keyboard>
- **Understanding:** <https://www.w3.org/WAI/WCAG21/Understanding/keyboard.html>
- **Techniques:**
 - <https://www.w3.org/WAI/WCAG21/quickref/#keyboard>
 - <https://www.w3.org/WAI/ARIA/apg/patterns/menuubar/>

Success Criterion 2.4.1 Bypass Blocks (Minimal) (A)

A mechanism is available to bypass blocks of content that are repeated on multiple Web pages.

Problem

- There is no **Skip to content** link

Success criteria:

- **Skip to content:** can be done by placing a visually hidden link in the top of the HTML so it is the first item you can focus on using the keyboard.
This link points to an anchor/target like an ID on the wrapper of the page content or a div above the page content. As such it can be used to jump directly to the content of the page. It should become **visible** when **focussed** on.

References:

- **Source:**
- **Understanding:** <https://www.w3.org/WAI/WCAG21/Understanding/bypass-blocks.html>
- **Techniques:**
 - <https://www.w3.org/WAI/WCAG22/quickref/#bypass-blocks>
 - <https://www.w3.org/WAI/WCAG22/Techniques/general/G1>
 - <https://www.w3.org/WAI/WCAG22/Techniques/html/H64>

Success Criterion 2.4.4 Link purpose (in context) (A)

The purpose of each link can be determined from the link text alone or from the link text together with its programmatically determined link context, except where the purpose of the link would be ambiguous to users in general.

Problem

- **'MijnWGK' button-link in header:** the button style suggests this link does something different than just any navigation item but there is no clear link description for it (eg. does it lead to a user login? Does it lead to an info page? Does it open a popup? ...)
- **Language navigation**
 - The title attributes just prints the abbreviation of the language, no info given.
- **Breadcrumbs:**
 - the active item, which is not a link, somehow has a title-attribute. This should not be the case.
- **[Publicaties:](#)**
 - 'Meer info' links should have a less generic text description.
- **[Pers:](#)**
 - similar remarks
- **[Publicatie detail:](#)**
 - Download link's text doesn't describe the link purpose accurately (what is being downloaded?)

Success criteria:

- **'MijnWGK' button-link in header:** add a title attribute that says something like "read more about MijnWGK app" to avoid confusion.
- **Language navigation:**
 - The title attributes should show the actual language name, as well as prefix what clicking it will do. Eg. 'See this site in Dutch' or at the very least say full language name 'Dutch'
- **Breadcrumbs:**
 - the active item should not have a title attribute on it
- **[Publicaties:](#)**
 - Ideally, you would make the teasers consistent everywhere, and only add a link to the title element. That way you can use [the pseudo-content technique](#) to make the entire card clickable: use a pseudo-element of the title link to cover the entire card in an invisible way. Turn the 'lees meer' into a span with attribute `aria-hidden="true"`.
 - Alternatively, you can add a 'title' attribute to the 'Meer info' link to say what they are linking to, eg. the title of the article: 'Lees meer over <title text>'.
- **[Pers:](#)**

- similar remarks -> get rid of the 'Meer info' link in favor of the pseudo-content technique
- [Publicatie detail:](#)
 - Download link: give it a bit of info in a title-attribute, eg. 'download document: <title of the document>'. If you can also include what type of document it is (eg. a PDF), that's a helpful extra but not strictly required.

References:

- **Source:** <https://www.w3.org/TR/WCAG22/#link-purpose-in-context>
- **Understanding:** <https://www.w3.org/WAI/WCAG22/Understanding/link-purpose-in-context.html>
- **Techniques:**
 - <https://www.w3.org/WAI/WCAG22/quickref/#link-purpose-in-context>
 - <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Attributes/aria-describedby>

<https://www.w3.org/WAI/ARIA/apg/patterns/breadcrumb/examples/breadcrumb/>

Success Criterion 2.4.7: Focus Visible (Minimum) (A)

This is partly custom styling, BUT the theme CSS should be checked to see that 'focus' states are not simply disabled by default for the elements described. That way it can be set up better by default.

The purpose of this success criterion is to help a person know which element has the keyboard focus. Authors are responsible for providing at least one mode of operation where the focus is visible. In most cases there is only one mode of operation so this success criterion applies. The focus indicator must not be time limited, when the keyboard focus is shown it must remain.

Note that a keyboard focus indicator can take different forms.

Problem

- **Almost no links or buttons have a focus style.** This is because the default outline has been disabled globally (`:focus {outline: none}`). This means most links in the site have no active or focus style when you tab through the page using the keyboard. Here are some examples:
 - **Cookie banner:** the 'cookieverklaring' and 'privacy' links don't have a visible focus style.
 - Any **text links** in page content
 - Links used for navigation, like the **header navigation** or the **footer links**
 - Any **buttons** and links styled to look like a button:
any button on the [homepage](#) or even submit buttons like [in the sidebar of jobs](#) or [the arrow-submit on the Press page](#)
 - [Info voor zorgpartners](#): 'Neem contact op'-button in the footer, and any other buttons in the footer
 - **Bold links**, eg. as part of contact info (see. [in this page sidebar](#))
 - The '**cards**' in the Section "Tips van onze experts"(homepage) and any other page with similar card component
 - Etc...
- While not explicitly demanded by the WCAG, we advise giving links (including button-style links) **hover states** as well, to make navigation easier. While some places have this (eg. the main navigation), a lot of them do not (eg. nearly all the links and buttons on the homepage)
- The '**Lees voor**' button that exists in a couple places, near the top of the content:
You either can't focus on it at all, or when tabbing forward, you start focussing on the hidden elements (like the settings of play function), without the panel actually opening.

Success criteria

Give focus styles to the focusable elements. Having a 2px border (or outline) works best.

- For **hover**, we advise having a 3:1 contrast difference between normal and hover state, for **focus** we advise doing that too, in addition having a focus indicator such as a 2px border.
This issue also relates to the 2.4.13 Focus Appearance (Level AAA) criterium, but can be argued in relation to 1.4.11 Non-text Contrast (Level AA) as well.

- The ‘Lees voor’ button: you should not be able to focus on the (visually) hidden elements. You should be able to focus on the main element to open the panel, and only then be able to access those elements when hitting ‘Enter’ to open its menu
This also overlaps with criterion 2.4.4 Keyboard
- **Cards components:** we suggest having a 2px blue outline around the entire card on focus, but also needs a hover state (can be the same)
 - Eg. the links in ‘Onze diensten’ on the homepage
 - Eg. the teasers in ‘Het Wit-Gele nieuws’
 - Eg. ‘Tips van onze experts’

References

- **Source:** <https://www.w3.org/TR/WCAG21/#focus-visible>
- **Understanding:** <https://www.w3.org/WAI/WCAG21/Understanding/focus-visible.html>
- **Techniques:**
 - <https://www.w3.org/WAI/WCAG21/Understanding/focus-visible.html#techniques>
 - <https://www.w3.org/WAI/WCAG21/Techniques/general/G195>

Success Criterion 2.4.8 Location (AA)

Users need to know where they are in a set of pages. This can be done by providing breadcrumbs, site maps, or other indicators to give context. This reduces confusion for people with cognitive disabilities.

Problem

- **Breadcrumbs:** while breadcrumbs are provided and styled, the section is not labeled as such, making it more difficult to isolate by screen-readers.

Success criteria:

- **Breadcrumbs:** an easy fix, is to add an aria-label or aria-labelledby pointing to a (visually hidden) heading (h2 for example) labeling it as 'Breadcrumbs'. This is done on the on the nav element and will make it identifiable for screen-readers

References:

- **Source:** <https://www.w3.org/TR/WCAG22/#location>
- **Understanding:** <https://www.w3.org/WAI/WCAG22/Understanding/location.html>
- **Techniques:** <https://www.w3.org/WAI/WCAG22/Techniques/general/G65>

Success Criterion 3.2.2 On Input (A)

We want content changes to be predictable, so users have to be warned if their context will change based on their input.

Problem

- [Tips en zorgverhalen](#):
 - **Navigation menu**: the **Thema's** item is a dropdown that leads the user to a subpage, when you select any of the options. This is not expected behavior from a dropdown menu item, which is normally part of a form, with a submit button. Simply styling it like a link, is not enough to provide context to visually impaired users and assistive tooling. This is sometimes detected by accessibility checkers as an infraction of [the 'Identify input purpose'](#) criterion, but we believe the current criterion is the correct one given the navigation context.
- [Stages](#): a similar thing happens here with choosing a province via "Informatie over". Only, it changes the content on the page instead of loading a new one.
- [Pers](#): a similar thing here: changing the dropdown with the province name, changes the results. Which is weird, because there is a submit button as well. But it's not in use.

Success criteria:

- [Tips en zorgverhalen](#)
 - There's 2 ways around this:
 - Add a submit button instead of initiating the submit on the change event
 - Or explain in the text above the item, that choosing an item will immediately send you to a subpage
 - In addition, we also advise changing the title and aria-label to be more instructional.
- [Stages](#): Same solution here as for 'Tips en zorgverhalen', but you say the results on the page will change.
- [Pers](#): make sure the submit button is what triggers the change in results, not the onChange of the select element itself.

References:

- **Source**: <https://www.w3.org/TR/WCAG22/#on-input>
- **Understanding**: <https://www.w3.org/WAI/WCAG22/quickref/#on-input>
- **Techniques**:
 - <https://www.w3.org/WAI/WCAG22/Techniques/html/H32>
 - <https://www.w3.org/WAI/WCAG22/Techniques/general/G13>
 - <https://www.w3.org/WAI/WCAG22/Techniques/client-side-script/SCR19>

Success Criterion 3.3.3 Error Suggestion (AA)

If an input error is automatically detected and suggestions for correction are known, then the suggestions are provided to the user, unless it would jeopardize the security or purpose of the content.

Problem

- [Sidebar forms](#): found in several places
 - Uses native `required` property which we can't rely on. It relies on native browser implementation to handle error messages for form fields and this is poorly implemented. Screen readers can do very little with this.
- [Job alert](#):
 - The error message when trying to submit without filling in an email address or phone number, or filling it in wrong, sometimes displays in English instead of Dutch. Some errors are translated, others are not.
 - There is no indication on the fields themselves, that they are filled in wrong.
-

Success criteria:

- [Sidebar forms](#):
 - These are things we can do to improve the error communication:
 - An **error message** (div) should be added underneath fields (eg. when submitting without filling in all required fields), it should have an ID. The field itself should have an attribute `aria-errormessage` to match the message ID.
 - For the **message content**: it should be in the language of the page. Other than that, customize the message to include what is wrong and how to correct it. Eg. 'Fill in a valid email address'
 - Remove the native `required` property from fields and do error checking on submit instead
 - **If the submit validation is based on Ajax**, the error message div should always be present in the source code (empty) and have an `aria-invalid` attribute set on the input field + `aria-atomic="true"` and `aria-live="true"` on the error message div. The content is then changed dynamically when an error occurs.
 -
- [Job alert](#):

- Make sure to translate all the possible field errors to Dutch. Eg. not just for empty value when it is required but also the one for email values not being the correct format. Same for phone field.
- It's good that there is no native **required** property on the fields. But we still need improvements in error communication, like before:
 - An **error message** (div) should be added underneath fields (eg. when submitting without filling in all required fields), it should have an ID. The field itself should have an attribute **aria-errormessage** to match the message ID.
 - For the **message content**: it should be in the language of the page. Other than that, customize the message to include what is wrong and how to correct it. Eg. 'Fill in a valid email address'
 - Remove the native **required** property from fields and do error checking on submit instead
 - **If the submit validation is based on Ajax**, the error message div should always be present in the source code (empty) and have an **aria-invalid** attribute set on the input field + **aria-atomic="true"** and **aria-live="true"** on the error message div. The content is then changed dynamically when an error occurs.

References:

- **Source:**
- **Understanding:** <https://www.w3.org/WAI/WCAG21/Understanding/error-suggestion.html>
- **Techniques:**
 - <https://www.w3.org/WAI/WCAG22/quickref/?versions=2.1#error-suggestion>
 - <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Attributes/aria-error-message>
 - <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Attributes/aria-describedby>

Success Criterion 4.1.2 Name, Role, Value (Minimum) (A)

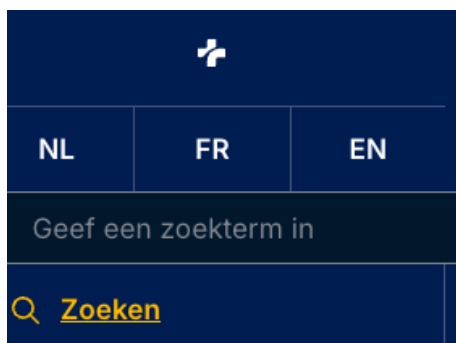
For all user interface components (including but not limited to: form elements, links and components generated by scripts), the name and role can be programmatically determined; states, properties, and values that can be set by the user can be programmatically set; and notification of changes to these items is available to user agents, including assistive technologies.

Problem

- The link on the **logo in the header** has no title attribute or other type of labeling
- In **mobile view**: the **main menu** has toggle-buttons (arrow icons), which are used to open up submenus. These are not labeled in any way.



- The **Search field** on mobile has no label (placeholder attribute doesn't count)



- On the **homepage**: The 'Waar kunnen we je helpen?' section
 - has an input field for postal code that is visually hidden but without label
- **Publicaties**:
 - "Zoek een document" field: has no label (HTML tag) or attribute that functions to label the field
- **Pers**
 - "Zoek een bericht" field, "Kies een provincie" Select and Submit button: same remark here

Success criteria:

- **Generally**, keep these things in mind when working with interactive elements like links and form items (input fields and buttons):
 - Add meaningful link descriptions, using the title-attribute or aria-label to links
 - Add title and aria-label to buttons
 - Add a label (can be visually hidden label tag using the `for` attribute and input ID) or aria-label for input fields
- **Logo link in header**: can add title attribute with something like 'go to homepage' in it
- In **mobile view**: the **main menu** submenu toggles: give the buttons a title attribute or visually hidden text + an aria-attribute (to bypass issues with screen readers). Let it say something like "toggle the submenu"
- **Search field**: give it a label (add attribute "for" attribute pointing to the ID of the input field) or aria-label.
- **Homepage**: The 'Waar kunnen we je helpen?' section
 - you should add a label tag (visually hidden if needed) or aria-attribute, if these are functional and visible.
- **Publicaties**:
 - Search filter: give it a label (tag with "for" attribute and ID of the input field) or put an aria-label attribute on the input itself.
- **Pers**
 - label the input & the select as mentioned above
 - The submit button (blue with an arrow on it) needs a title attribute + an aria-label attribute. Currently there is a bit of text in the button that has 'display: none', that isn't going to get picked up by anything.

References:

- **Source**: <https://www.w3.org/TR/WCAG22/#name-role-value>
- **Understanding**: <https://www.w3.org/WAI/WCAG22/Understanding/name-role-value.html>
- **Techniques**: <https://www.w3.org/WAI/WCAG22/quickref/?versions=2.2#name-role-value>
 - <https://www.w3.org/WAI/WCAG22/Techniques/html/H91>
 - <https://www.w3.org/WAI/WCAG22/Techniques/failures/F68>

WAI ARIA 1.1 - ARIA for non-semantic HTML

WAI-ARIA is used to create an accessible Rich Internet Application. There are recommended usage patterns that should be followed.

Languages used to create rich and dynamic web sites, e.g., HTML, JavaScript, CSS, and SVG, do not natively include all the features required to make sites usable by people who use assistive technologies (AT) or who rely on keyboard navigation.

So when using **design patterns (like dropdown menus)**, please refer to [the ARIA documentation](#) to apply the correct attributes and make these patterns accessible but note that simply applying ARIA attributes does NOT automatically apply styling or behaviour. You still need JS and CSS for that. ARIA simply adds elements to the page's accessibility tree.

Problem

- The **main menu** uses buttons to toggle submenus on mobile view. These need extra properties for the submenus to be accessible using assistive tools. It's called a 'disclosure pattern'
- The **search item** in the header toggles the appearance of a block with a search input, but this functionality is not made accessible for screen readers. A **dialog** tag is also used to wrap the form but there are problems with this:
 - a. this component does not apply any of the functionality of a dialog: no closing button, no keyboard interaction, etc...
- Whenever you **select a bit of text**, there is a pop-up that opens, with the option to read or translate that bit of text. This is not blocking other accessible functions, but since it's undocumented behavior with no (visual & detectable) indicator, it is almost undetectable by the user and confusing (even a mild annoyance) when you accidentally trigger it (eg. by touch-dragging a for a bit too long on touch-screen or by using assistive technology to select text).
- **Cookie banner:**
 - a. Once you pick your **cookie preferences**, there is also a '**Cookie-instellingen**' button fixed to the bottom of the page. There is no connection in the HTML between that button and the hidden content that can be detected by assistive tooling (ARIA attributes).
- **Homepage**
 - a. **Veelgestelde vragen:** this component is not accessible, as you never enter the component or focus on anything while using keyboard or an assistive tool. Making it fail [Success Criterion 2.1.1 \(keyboard\)](#).
It also doesn't use any HTML that connects the trigger to the hidden content (like a disclosure pattern using ARIA or even standard html like 'details' and 'summary' tags)
- [Some detail pages of 'andere zorgdiensten'](#) also have similar FAQ components
- [Publicaties:](#) also seen in [Pers](#), on the Search overview and other places with filters and/or overview results (teasers)

- a. **Filtering results:** The filtering of results is Ajax based and changes to the results is not communicated to screen readers.
- b. **The pager:** the page change is also done using Ajax, without communication to screen readers.
- [Pers:](#)
 - a. Same issue with the pager
 - b. **Carousel of images:** this is again a type of component that is not fully accessible and needs ARIA attributes and patterns to make it so
 - c. Enlarge a carousel image (colorbox):
 - The 'aria-label' attribute contains stringified json instead of readable text.
 - There's a lack of focus styles within the carousel so you can't see if you get to the arrows or if there's maybe a close button you can activate (infractions against both the '[focus](#)' criterion and the '[keyboard accessible](#)' criterion).
- [Detail pages for 'andere zorgdiensten'](#)
 - a. Also has a carousel, but with text quotes. Similar issues as the image carousel (except for the image problems, obviously)
- [Tips en zorgverhalen](#)
 - a. 'Meer tips en verhalen' button: Near the bottom of the page, this link appends more content to the list of teasers but needs to tell screen-readers about the content change
- [jobs filtered:](#)
 - a. The 'active filters' in this example, contains 1 active filter, constructed from a button. The problem is that the button is simultaneously the listing of the filter (using visual text) but also has a function (clicking it removes the filter) that is not communicated.

Solution

- **main menu: toggle buttons for submenus on mobile:**
 - a. There is a [hybrid disclosure navigation pattern](#) explaining how to construct this better. Take a look at the link for specifics. It uses buttons with aria attributes to toggle the submenu's:
 - aria-controls (referencing the ID of the content wrapper) and aria-expanded true or false (depending on the state of the content)
- The **search item** in the header: there are 2 ways to fix this, either by implementing the dialog fully or by making an accessible collapsible component (probably easier).
 - a. If you're going for a dialog, apply a dialog pattern, eg. [as outlined in this article](#) or even [in the this WAI-ARIA pattern example](#). Some things to pay special attention to:
 - The trigger should be a button (not a link, as is the case now), and it should only open the dialog, not toggle it.
 - Label the dialog so its meaning/use is clear
 - Should be able to open and close it with keyboard (eg. ENTER & ESCAPE) as well as mouse
 - Make it fast & easy to dismiss, best way is a close-button in the dialog

- focus goes into the pop-up when opening and leaves when closing (is already the case, with the input field)
 - When closing, focus should go back to the triggering element, or a next logical element
 - Focus trap (so you can't tab out of it while it is visually blocking the page or part of it)
 - b. Alternatively, you could use [a disclosure pattern like these menu buttons instead](#).
Some things to pay attention to then:
 - Don't use the dialog-tag (use divs instead, for example)
 - Still need a button to trigger it: make sure to use the appropriate aria attributes: aria-controls and aria-expanded
 - Move the search block to be underneath the button, in the markup.
- **Text-selected popup:** at the very least, we would advise to document this behavior, near the top of the page to help with the confusion. But ideally, we would advise to remove this plugin entirely (including the separate 'read' button).
All Operating Systems have built-in functions to read content aloud and there are (free) screen-readers and related applications available. The users that need such assistance are well-aware of the options and are more qualified to decide for themselves how to deal with text in websites.
- **Cookie banner:**
 - a. **cookie preferences:** since this component is a form of a 'disclosure' component, it is advised to take a look at [the disclosure WAI ARIA pattern](#) for how to change the HTML to make it accessible. Eg. the button needs an `aria-controls` property that has an ID as value, that ID should be on the hidden content container.
 - **The same goes for the accordion inside the hidden content.** This also needs to use the disclosure pattern
- **Homepage**
 - a. **Veelgestelde vragen:** since this component is a form of a 'disclosure' component, it is advised to take a look at [the disclosure WAI ARIA pattern](#) & [FAQ example](#) for inspiration on how to change the HTML (and possibly JS) to make it accessible and usable via keyboard.
- [Some detail pages of 'andere zorgdiensten'](#) also need to have their FAQ components changed in a similar way
- [Publicaties:](#)
 - a. **Filtering results:** You should also **announce the changes** for any screen readers. This can be done by using `aria-live="polite"` and `aria-atomic="true"` attributes to the 'publicaties getoond' wrapper element (`class="view__head"`).
It is NOT suggested to use it on the results wrapper itself, because it will read the entire grid of items which could be distracting.
 - b. Since loading can take a while and there is already a preloader, you can additionally set an attribute `aria-busy="true"` while the feed is loading and setting it to `aria-busy="false"` when the loading is done.

- c. **Pager:** same fix as for the 'filtering results' item, communicate content change on page change by using the preloader as indicated before: toggle aria-busy as feed is loading and when the content is done.
If you have used aria-live and aria-atomic on the 'publicaties getoond' item, this will communicate on which page the user now is
 - Another things could be: move the focus to the wrapper of the search results so a user can start tabbing through the new search results
- [Pers:](#)
 - a. **Pager, etc...:** Same solution as in 'Publicaties'
 - b. **Carousel 'Fotobibliotheek':** you can refer to [this pattern found in here](#) to understand the desired carousel behavior. [Here is also a code example](#) for reference.
The important parts are that keyboard navigation can be used to cycle through and all slides are reachable/readable & focusable:
 - Add a property aria-roledescription="carousel" on the carousel wrapper, to indicate to screen readers that this is a carousel. As a sidenote: you can't add tabindex to elements that don't have a role, so it's doubly necessary to fix this.
 - Use buttons for the controls + they need to be keyboard accessible + focus style: the arrows are okay but the paging 'dots' are not
 - The pager wrapper needs `role="tablist"` and the pager buttons will need `role="tab"` and `aria-selected="<false or true>"`
 - Make sure users can see what they are focussed on (focus style):
 - Eg. if there is a link that opens up the image in a popup, you need a focus style (eg. outline or overlay on the entire image)
 - Eg. if there is other links in the slide, that should be visible
 - Enlarging a carousel image (colorbox):
 - The 'aria-label' of the link, needs to explain that clicking the link will open up the image in a larger format. Not mention what is in the image (that is what the alt-attribute on the image itself is for) and not contain json code
 - The open colorbox needs focus styles on the UI-elements, such as arrows and close button (if one exists, which should be the case)
- [Detail pages for 'andere zorgdiensten'](#)
 - a. Similar fixes for the Testimonials carousel
- [Tips en zorgverhalen](#)
 - a. **'Meer tips en verhalen'** button: like with the 'Publicaties' pager, you can have the preloader toggle an aria-busy property to indicate activity is happening and when it is finished.
Another thing that can help, since there is no message displaying the number of results, is to have a visually hidden message for that, which uses aria-live and aria-atomic="true" to display the change in items. Eg. saying "x items displayed"
- [Jobs filtered:](#)
 - a. Active filters:

- Adding a title & aria-label to the button, saying something like 'remove filter <name of filter> and reload the results' or 'show results without filter <name of filter>' will make the function clear to screen readers
- You should also be using an unordered list to list the teasers, instead of divs

References:

- **Source:** <https://www.w3.org/TR/wai-aria-1.1/>
- **Practices:** <https://www.w3.org/WAI/ARIA/apg/practices/>
- **Patterns:**
 - <https://www.w3.org/WAI/ARIA/apg/patterns/> (eg. for disclosure/faq, dialogs)
 - Dialog pattern: <https://www.w3.org/WAI/ARIA/apg/patterns/dialog-modal/>
 - Dialog widget: <https://www.accessibility-developer-guide.com/examples/widgets/dialog/>