



Boerenbond DEP

Accessibility audit

Client: Boerenbond
Project: Boerenbond DEP
Date: 27/11/2025

Table of contents

Table of contents	2
Introduction	4
Scope of the evaluation	4
Out of scope	4
Summary	5
Priorities	5
1. Keyboard & navigation	5
2. Visibility	5
3. Ajax	5
4. Feedback & forms	5
5. Content and structure	5
Audit results	6
No Accessibility Statement page	6
Success Criterion 1.1.1 Non-text content (minimum) (A)	7
Problem	7
Success criteria:	7
References:	8
Success Criterion 1.2.2 Captions (Prerecorded) (A)	9
Success Criterion 1.2.3 Audio Description or Media Alternative (Prerecorded) (A)	9
Problem:	9
Success criteria:	9
Success Criterion 1.3.1 Info and Relationships (Minimum) (A)	11
Problem	11
Success criteria:	11
References:	11
Success Criterion 1.3.2 Meaningful Sequence (A)	12
Problem	12
Success criteria:	12
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)	14
Problem:	14
Success criteria:	14
References:	15

Success Criterion 2.1.1 Keyboard (Minimum) (A)	16
Problem	16
Success criteria:	16
References:	16
Success Criterion 2.4.1 Bypass Blocks (Minimal) (A)	18
Problem	18
Success criteria:	18
References:	18
Success Criterion 2.4.4 Link purpose (in context) (A)	19
Problem	19
Success criteria:	19
References:	19
Success Criterion 2.4.7: Focus Visible (Minimum) (A)	20
Problem	20
Success criteria	20
References	20
Success Criterion 3.3.3 Error Suggestion (AA)	22
Problem	22
Success criteria:	22
References:	23
Success Criterion 4.1.2 Name, Role, Value (Minimum) (A)	24
Problem	24
Success criteria:	24
References:	25
WAI ARIA 1.1 - ARIA for non-semantic HTML	26
Problem:	26
Success criteria:	26
References:	27

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:** Boerenbond DEP
- **Website base url:** <https://boerenbond.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
- Areas not accessed by visitors (eg. admin area)
- Pages that are on a different domain, as this would be part of a separate report.

Summary

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

Priorities

This list orders the priorities from higher to lower by impact. The higher priority ones often center around keyboard navigation, Ajax and visibility.

1. Keyboard & navigation

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 \(Minimum\) \(A\)](#)

3. Ajax

1. [WAI ARIA 1.1 - ARIA for non-semantic HTML](#)

4. Feedback & forms

1. [Success Criterion 3.3.3 Error Suggestion \(AA\)](#)

5. Content and structure

1. [Success Criterion 1.3.2 Meaningful Sequence \(A\)](#)
2. [Success Criterion 2.4.4 Link purpose \(in context\) \(A\)](#)
3. [Success Criterion 1.3.1 Info and Relationships \(Minimum\) \(A\)](#)
4. [Success Criterion 1.3.5 Identify Input Purpose \(AA\)](#)
5. [Success Criterion 1.1.1 Non-text content \(minimum\) \(A\)](#)
6. [Success Criterion 2.4.1 Bypass Blocks \(Minimal\) \(A\)](#)
7. [Success Criterion 1.2.2 Captions \(Prerecorded\) \(A\)](#)
8. [Success Criterion 1.2.3 Audio Description or Media Alternative \(Prerecorded\) \(A\)](#)
9. [Success Criterion 4.1.2 Name, Role, Value \(Minimum\) \(A\)](#)

Audit results

No Accessibility Statement page

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 certain situations.

Problem

- **Logos (in header and footer):** the alt text should reflect the text of the logo, not the 'home' reference.
- **Some images don't have an alt-attribute:** every img should have one, even if it is empty
 - Eg. the cartoon on the Homepage, some images in the sidebar, various on the page, ...
- **About decorative vs. informative images:**
 - Most images here can be said to be **decorative**, notably teasers, hero images,... So they should have an empty alt-text. Because this kind of non-informative text can overwhelm a user who uses screen readers, especially on image-heavy pages. It makes it difficult to get to the information.
 - Eg. homepage → images next to text blocks: needs empty alt tags
 - Eg. ['word lid'](#) the blue icon images & the photos: need empty tags
 - Eg. [Nieuws overview and detail pages](#) → the photos need empty alt tags
 - Eg. [Diensten](#) → same for these teaser images
 - Eg. [the image of 6 icons with text below the video](#) is descriptive, but the text & meaning are repeated as actual text further down the page. Make the alt tag empty and treat it as decorative.
 - ...
 - For images that are supposed to **add information** to the content of the page, you should take care to use descriptive text that reflects what is being seen:
 - **Logos** → the alt text for jpg/png/gif (or the title if on an SVG), needs to include ALL of the text visible on the logo
 - Eg. ['word lid'](#) :
 - The long advertisement banners need alt tags that give you the written information that is part of the graphics
 - The orange promotion graphic: make the alt text the same as the text in the picture
 - the butterfly logo does this correct
 - **Cartoons** like [on this page](#) and [here](#) should get a description of what is happening, along with the text being said
 - **Text-heavy images & infographics:**
 - Eg. the linked banner under ["Ook genieten van Boer&Tuinder?"](#) → you should use the image's alt text to reflect the text that is in the image

Success criteria:

- **Logos (in header and footer):** replace the alt text with the word 'Boerenbond'

- **Some images don't have an alt-attribute:** check the img templates and make sure there is always an alt attribute rendered (even if empty)
- **decorative vs. informative images:** For images that have an alt-text, you have to ask yourself:
 - Is the image as decoration or is it functional/informative? [Here is a guide to help](#)
 - If it is decorative, leave the alt-text empty (for SVG, see previous points)
 - But if it is not decorative, you need to add text that is descriptive (for SVG, see previous points)

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/WCAG22/quickref/#non-text-content>
 - <https://www.w3.org/WAI/tutorials/images/decorative/>

Success Criterion 1.2.2 Captions (Prerecorded) (A)

Success Criterion 1.2.3 Audio Description or Media Alternative (Prerecorded) (A)

Enable people who are **deaf or hard of hearing** to watch synchronized media presentations. Captions provide the part of the content available via the audio track. Captions not only include dialogue, but identify who is speaking and include non-speech information conveyed through sound, including meaningful sound effects.

- Eg. a video slideshow would have captions, which display what the info slides are saying. All textual information presented, is described in the captions.

Provide people who are **blind or visually impaired** access to the visual information in a synchronized media presentation in the same human language as the video or page on which it appears:

- One approach is to provide audio description of the video content -> describe what is being seen, in the captions.
- The second involves providing all of the information (both visual and auditory) in text form -> a text transcript, which can be read by screen readers
- **Captions** are provided for all prerecorded audio content in synchronized media (except when the content already exists as text, and this is marked as such).
 - Eg. a video interview would have captions, which display what the people are saying. If visual information is presented, this is described in the captions. That could be descriptions of what is being seen in the scene or even what we are viewing on a graph.
- **Video: An alternative for time-based media or audio description** of the prerecorded video content is provided for synchronized media (except when the content already exists as text, and this is marked as such).
 - Eg. a podcast episode could have a transcript available.

Problem:

- There are presentation videos (eg. [About us](#)) that has no captions to display the textual information, for screen readers to access.

Success criteria:

- In case of the **videos**:
 - **Non-speech information**: include this in captions as needed. Eg. when an infographic or references to something visual is being presented, you should describe what is being seen (unless the people in the video describe it themselves of course).

- **Speaking:** add captions in the same language as the page the video is presented on. Since some parts of the site are multilingual, make sure videos displayed in other language pages are captioned in that language.
 - **Check the captions** for errors and typos, and correct them
- **Video players** can also cause issues with **shortcuts for screen readers**. So we recommend adding the property `disablekb="1"` to the iFrame properties to disable specific shortcuts and bypass the problem.

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

- **Use buttons for interactive elements:**
 - Eg. [for Favorites](#) → the 'star' icon is using a link instead of a button
- **Links should be styled to convey link behaviour**
 - Eg. [links in the 'Dalers' block](#) → they look like text but seem to be clickable links
 - Eg. [links in the table](#) → they just look like bold text

Success criteria:

- **Use buttons for interactive elements:**
 - Eg. [for Favorites](#) → the 'star' icon should be wrapped in a button instead of a link. Links should be kept for navigation purposes.
- **Links should be styled to convey link behaviour**
 - Eg. [links in the 'Dalers' block](#) → give the links an underline, this is consistent with other text links elsewhere
 - Eg. [links in the table](#) → same here

References:

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

Success Criterion 1.3.2 Meaningful Sequence (A)

The sequence in which content is presented, needs to be in a meaningful order for assistive technology to present it to users.

Problem

- **Homepage** is missing an H1
- **For overviews with a filter sidebar:**
 - The blocks of filters all have an H2 as their title, while overview of teasers has no main heading but the teasers have H3 headings. This can be confusing when assistive technology tries to make sense of the outline of the page
 - Eg. <https://www.boerenbond.be/projecten>

Success criteria:

- **Homepage:** all pages should have a page title
- **For overviews with a filter sidebar:**
 - We recommend giving the results a (visually hidden) title 'results' that is an H2.
 - At the same time, the filter blocks should get an H3 for their headings and the entire sidebar an H2 with title 'Filters'.

References:

- **Source:** <https://www.w3.org/TR/WCAG22/#meaningful-sequence>
- **Understanding:**
<https://www.w3.org/WAI/WCAG22/Understanding/meaningful-sequence.html>
- **Techniques:**
<https://www.w3.org/WAI/WCAG22/quickref/?showtechniques=132#meaningful-sequence>

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 make it easier to fill in forms. We can use code to indicate the purpose of common inputs.

Problem

- **All pages with forms:** any form that needs data from the user (an email, name, address, ...) should allow for autofill.

Examples:

- [Contact form](#) for people
- [Office contact form](#)

Success criteria:

- **All pages with forms:**
 - Any form that needs data from the user (an email, name, address, ...), should be given an autocomplete property with the proper value ([see here for reference](#))
This is not needed for search forms, as there is no personal info to submit.
Eg. for email fields, you can add the attribute `autocomplete="email"`

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 are elements such as 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.

Problem:

- Some colors don't follow the contrast rules

Success criteria:

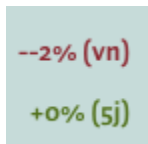
Ensure all these elements have sufficient color contrast between the text in the foreground and background color behind it.

Elements to fix:

- **Homepage sidebar:** Location name is too low contrast (should be 4.5:1)



- **Homepage marktberichten:** red and green numbers are too low contrast (should be 4.5:1)



- white on green text: white on green text is too low contrast (should be 4.5:1 for this size text). Make the green color darker when using it with white. Or make the text a dark color instead.
 - a. We see this in **Footer & login button in header**



- b. But we also see it in Focus styles: eg. **focussed links in the footer**

- **Blue buttons:** eg. on homepage, are too low contrast as well. Make the text 19px + bold or make the blue darker.



Contrast can be checked [using a color contrast checker](#) or the contrast checker in Chrome devtools.

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>

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

- **Cookie banner:**
 - When you confirm your choice, the banner closes and focus is set on the link around the logo (header). Unfortunately, this skips past the 'skip link'. So you can't easily jump to the content
- **Header navigation:**
 - The 'Marktinformatie' and the 'Advies' menu items can not be reached by keyboard
 - The foldout menus can be opened by focussing on the arrow button + hitting Enter but you can not close them the same way. They collapse when focusing on the arrow button.
- **Image carousel/slider:**
 - Eg. on the Homepage (bottom) we get a slider that can not be properly operated by keyboard. The arrow buttons are activated on focus instead of on Enter (focus + Enter).
 - When you focus on the prev/next button, the link inside the visible slide automatically becomes focussed as well, making for a very confusing experience.

Success criteria:

- **Cookie banner:**
 - When you confirm your choice, and the banner closes, focus should not be set. You should be able to start tabbing and get to the skip link first, before any other link.
- **Header navigation:**
 - All menu item needs to be reachable and operable
 - The foldout menus should not collapse when focussed, you should be able to close them the same way they are opened (focus on button + Enter)
- **Image carousel/slider:**
 - Make sure the previous/next buttons work only trigger when purposefully used: focus + Enter
 - You should only be able to reach the links inside a visible slide, when tabbing away from the slider's buttons. Eg. the prev & next buttons are the first focusable items, then the visible slide itself, and then the links inside a slide. [Here is an example of the desired behaviour.](#)

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/disclosure/examples/disclosure-navigation-hybrid/>
 - <https://www.w3.org/WAI/ARIA/apg/patterns/tooltip/>
 - https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Reference/Roles/tooltip_role

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

- **Iframes:** For media players, such as the podcast players ([on this page](#)) → these should have a non-empty 'title'-attribute to identify the frame.

Success criteria:

- **Iframes:** add a title attribute to the iFrames, with some description of what it shows.

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

- **Missing link description:** Some links have no 'title' nor other kind of labeling or text that is visible to screen readers.
 - Eg. the advertisement banners
- **Insufficiently descriptive link text.**
 - There are various generic 'Lees meer >>' links. Eg. <https://www.boerenbond.be/word-lid> → the teasers under 'Daarom hoor jij bij Boerenbond!'

Success criteria:

- **Missing link description:**
 - Eg. the advertisement banners → you can either add a 'title' to the link around the image, or add visually hidden text which describes the link purpose. Eg. 'download the app Boerenbond app'.
As a last resort, you could use the image alt text to describe the image + link purpose. But the usefulness will depend on the image complexity (eg. if there is a lot of text in the image, it needs to be in the alt text as well) so we wouldn't advise this approach.
- **Insufficiently descriptive link text.**
 - Make sure to provide a descriptive title-attribute if it can not be done in the visible text. Eg. 'read more about <title of article>'

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

- **Advertisement banners:** various pages (including Homepage), feature a long banner near the top of the page. This does not have a clear enough focus style, as the blue border is often almost as dark as the image + you can only see the bottom of it.
- **Teasers / clickable blocks:** noticeable on all Homepage blocks, News overview, [this calendar overview](#), ...
 - Various clickable items that combine a title with text (and sometimes image), have 2 focuses: a visible one and a hidden one.
You first focus on the title link, and when you tab again, an invisible focus is on the entire block (via a pseudo-element in the wrapping link).
 - Some blocks also have a (hidden) `link--more` link. This gets a hidden focus that is causing the layout to shift in some places (eg. The 'Vormingen' blocks on Homepage right sidebar).

Success criteria

- **Advertisement banners:** adding a few pixels of whitespace between the image and the border should fix things. Also make sure the outline goes around the whole image instead of just a border at the bottom
- **Teasers / clickable blocks:**
 - You should make sure only 1 element receives focus, and it should be visible.
Our recommendation is to give the block an outline when focussed and remove the link on the title. There is no use for 2 links going to the same destination.
 - Removing the hidden `link--more` link should prevent the layout from shifting (eg. on the Homepage blocks)

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 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

- **Forms:**
 - The error messaging isn't consistently communicated to screen readers

Examples:

- [Contact form](#) for people
- [Office contact form](#)

Success criteria:

Adding error messages to the fields is required if you use Ajax validation.

On page submit forms, you need a big notification but you can choose to put the errors in the big error notification directly (is the most readable) or use individual error messages.

- **If submitting on page load:**
 - set a big error message at the top of the page, with attributes `role="alert"` along with `aria-live="polite"` and `aria-atomic="true"` (those last 2 are hacks for certain screen readers to force this to be read)
 - mention there are errors in the submitted form
 - add an anchor link to the form (eg. id on the form, on a wrapper or on a heading)
 - If you use this box to list the errors:
 - use unordered list markup
 - make sure to mention the field label for each error and what is wrong + how should you correct it:
 - eg. "Last name" is a required field
 - eg. "email" needs to be a valid email address
 - You could each item to the corresponding field (optional)
- **Fields:**
 - We can't rely on HTML5 (eg. the `required` attribute), as it is the browser that interprets how to display any errors. Instead, implementing an aria based solution.
 - block browsers from rendering native form field error messages. The drastic way is to remove `required` and email input types, but there are [ways to override the default browser behaviour](#) (see also [this documentation](#)).
 - Set `aria-required` on required fields instead
 - Set `aria-invalid="true"` on the field, when it has an error
 - When you display an individual error message per field:

- set an `aria-errormessage="<ID of message div>"` attribute on the field itself
- set an ID on the message div
- should be in the language of the page.
- clear error text: what is wrong and how should you correct it? Eg. 'Fill in a valid email address'
- **If you use Ajax** to validate the form:
 - You don't need a big error message, but you can use that if you really want it. The div just needs to be already printed in the source code & content can be set dynamically (eg. injected or toggling `aria-hidden`)
 - However, you should have error messaging on the fields themselves instead of in a big message on top
 - The message divs need to already be printed in the source code on page load and have `aria-live="polite"` and `aria-atomic="true"` on it.
 - It has to be hidden from screen readers until the text is added, so using `aria-hidden`

References:

- **Understanding:** <https://www.w3.org/WAI/WCAG22/Understanding/error-suggestion.html>
- **Techniques:**
 - <https://www.w3.org/WAI/WCAG22/quickref/?versions=2.2#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

- **Homepage and other pages:** The `aria-labelledby` attributes are not allowed on the tags that have a generic role (like a plain `div`). These are only allowed on elements like interactive elements or for elements made interactive via other ARIA declarations. Some examples:
 - `<div id="navigation"`
 - `<div class="bb-advertisements"`
 - `<div id="block-boerenbond-contact-info"`
- **Header Search**
 - There are multiple labels pointing to the same input field. Basically, the 2 search blocks use the same `for` and `id` attribute values but should be unique.
- **Main navigation foldout menus**
 - The `aria-controls` attributes on the buttons for main navigation items 'Nieuws & info' and 'Marktinformatie' point to non-existing ID's:
 - `submenu-nieuws--info`
 - `submenu-marktinformatie`

Success criteria:

- **Homepage and other pages:** for the `aria-labelledby` elements: for most of these, you can remove the attribute. The title it refers to is directly above the content, so the relation is relatively straightforward for screen readers to pick up on.
 - For the navigation: add `role="region"` to the div with `id="navigation"`. You can do this with other items where the label is not directly above the content it connects with.
 - For the bb-advertisements banner: the label that is referred to does not exist. So the attribute definitely has to be removed.
- **Header Search**
 - The `for` and `id` attribute values for each search block should be changed to be unique.
- **Main navigation foldout menus**
 - Make sure the foldout menus have ID's & use those ID's in the corresponding buttons' `aria-controls` attributes

References:

- **Source:** <https://www.w3.org/TR/WCAG21/#name-role-value>
- **Understanding:** <https://www.w3.org/WAI/WCAG21/Understanding/name-role-value.html>
- **Techniques:** <https://www.w3.org/WAI/WCAG21/quickref/#name-role-value>
 - <https://www.w3.org/WAI/WCAG21/Techniques/html/H91>
 - <https://www.w3.org/WAI/WCAG21/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:

- **Foldout filters:**
 - [Some filters](#) consist of a button that toggles a list of checkboxes but they are not recognizable by assistive technology.
- **Filterable overviews:**
 - On some pages with filters, the results change immediately, when using the filters to change an option, this violates the 'On Input' criterion: you can't change content on such interactions without making the visitor aware.
 - This behavior is also inconsistent with [other pages that use a submit instead](#)
 - When the items change, this change is not communicated to screen readers
 - **Examples:**
 - <https://www.boerenbond.be/markten/varkens> → ajax based filtering
 - <https://www.boerenbond.be/projecten> → this one seems to cause a page load when clicking a filter
 - <https://www.boerenbond.be/ledenvoordelen> → same here

Success criteria:

- **Foldout filters:**
 - [These filters](#) should follow [this type of disclosure pattern](#), similar to the foldout menus of the main navigation. At the very least, using aria-expanded and aria-controls on the button, to communicate state and the relationship with its foldable content (using an ID).
- **Filterable overviews**
 - You can either tell the user that changing the filter will change the content (above the filters) or add a visible submit button
 - We advise the visible submit, because that would make it consistent with [other pages with such filterable overviews](#)
 - When using Ajax for the content change, you should also announce the changes for any screen readers. This can be done by using `aria-live="polite"` attribute with

an `aria-atomic="true"` attribute on a div that keeps the count of the results. Eg. "36 results found". This can be visually hidden if needed.

- 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.
- If ajax loading can take a while, 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. But this is not required.

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)
 - <https://www.w3.org/WAI/ARIA/apg/patterns/disclosure/examples/disclosure-navigatio-n-hybrid/>
 - <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Reference/Attributes/aria-hidden>