



# Credendo - credendo

## Accessibility audit

Client: Credendo  
Project: Credendo  
Date: 17/09/2024  
Last edited: 24/10/2024

# Table of contents

<b>Table of contents</b>	<b>2</b>
<b>Introduction</b>	<b>4</b>
Scope of the evaluation	4
Out of scope	4
<b>Summary</b>	<b>5</b>
Priorities	5
1. Keyboard	5
2. Ajax	5
3. Feedback	5
4. Visibility	5
5. Content	5
<b>Audit results</b>	<b>6</b>
Success Criterion 1.1.1 Non-text content (minimum) (A)	6
Problem	6
Success criteria:	6
References:	6
Success Criterion 1.4.3 Contrast (Minimum) (A)	7
Problem:	7
Success criteria:	8
References:	8
Success Criterion 2.1.1 Keyboard (Minimum) (A)	9
Problem	9
Success criteria:	9
References:	10
Success Criterion 2.4.7: Focus Visible (Minimum) (A)	11
Problem	11
Success criteria	11
References	11
Success Criterion 3.2.2 On Input (A)	12
Problem	12
Success criteria:	12
References:	12
Success Criterion 3.3.3 Error Suggestion (AA)	13
Problem	13
Success criteria:	13

References:	13
Success Criterion 4.1.2 Name, Role, Value (Minimum) (A)	14
Problem	14
Success criteria:	14
References:	14
<b>Accessibility statement page</b>	<b>16</b>
How to:	16
What is (and isn't) part of that page:	16
Examples of Accessibility Statement pages:	16

## Introduction

- This report describes the **conformance** of the submitted website **with W3C's Web Content Accessibility Guidelines (WCAG)**.
- It uses a set of sample pages, and the WCAG Violations are documented with **code snippets, screenshots and/or links** 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** and assigned a level by **severity and impact** as much as possible, to make prioritizing easier.
- We write **in English**, as most references and excerpts are from English-language WCAG documentation.

## Scope of the evaluation

- **Website name:** Credendo
- **Website base url:** <https://credendo.com/en>
- **WCAG Version:** 2.1
- **Conformance target:** AA
- **Additional evaluation requirements:** *not provided*

## Out of scope

- External services integrations
- Parts of the site that require a login

# Summary

Based on this evaluation, the website does not meet WCAG 2.1, 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.

### 1. Keyboard

1. [Success Criterion 2.1.1 Keyboard \(Minimum\) \(A\)](#)
  - Fix the language navigation and submenus so the site can be navigated properly
2. [Success Criterion 2.4.7: Focus Visible \(Minimum\) \(A\)](#)
  - This is related to keyboard navigation: the keyboard user needs visible focus styles to know where they are

### 2. Ajax

1. [Success Criterion 3.2.2 On Input \(A\)](#)
  - This relates to predictability of the user interface. Eg. users with visual impairments knowing that actions on the page result in a change of context.

### 3. Feedback

1. [Success Criterion 3.3.3 Error Suggestion \(AA\)](#)
  - a. User needs correct feedback from forms to know when there are errors and understand what/how to correct mistakes

### 4. Visibility

1. [Success Criterion 1.4.3 Contrast \(Minimum\) \(A\)](#)
  - This will quickly improve visibility for a number of users

### 5. Content

1. [Success Criterion 4.1.2 Name, Role, Value \(Minimum\) \(A\)](#)
  - Components should be correctly labeled so their function can be determined (eg. by screen readers)
2. [Success Criterion 1.1.1 Non-text content \(minimum\) \(A\)](#)
  - Text alternatives for visual information

# Audit results

## 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

- There are several images that have filled in alt-properties that print the file name instead of giving a description of the image
- There are several images that are clearly not decorative, yet have an empty alt-property. Eg. [in this Knowledge Hub article](#), there is a graph.

### Success criteria:

- Don't use the file name for descriptions, these don't mean anything. If the images are decorative, you can leave the alt empty but a lot of images at least add something meaningful to the content so it is advisable to add proper descriptions of what is seen in the picture.
- For graphs and infographics, you can try to give a good description of what is shown but it might be easier to have a visually hidden block of text or even a table, that lists the information in a logical fashion.

### References:

- **Source:** <https://www.w3.org/TR/WCAG21/#non-text-content>
- **Understanding:** <https://www.w3.org/WAI/WCAG21/Understanding/non-text-content.html>
- **Techniques:** <https://www.w3.org/WAI/WCAG21/Techniques/html/H67>

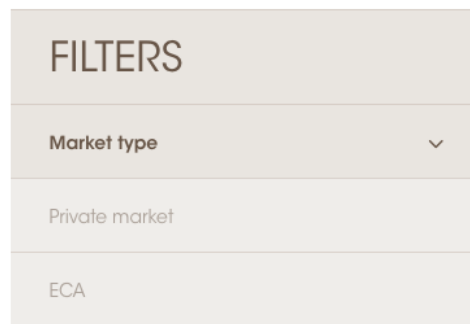
## Success Criterion 1.4.3 Contrast (Minimum) (A)

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:** Large-scale text (> 24 pixels) and bold text of at least 18.5px and images of large-scale text have a contrast ratio of at least 3:1;
- **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:

- Navigation (and other places): the orange text on white is too low contrast. It should be 4.5:1 because the text is < 24px (it's 16) when not bold.  
You also see this on the 'read more' links of the 'Latest articles' of the Homepage and other pages.
- There are various instances of light-grey text on white that need 4.5:1 contrast ratio
  - Homepage and Pressroom page:
    - the grey 'published on' text of the 'Latest articles' and 'Press articles' teasers is too light.
    - Grey color of the pager is too light
  - Solutions: the grey color of the open filters is too light for the background



- Light-brown text on orange: eg. on the Homepage, the light text 'Solution type' in the 'Find what you need' block is too light. That goes for all pages where these styles of blocks are used. Needs to be 4.5:1 contrast ratio
- Form field borders, as well as sometimes other parts, are too low contrast:
  - [This form](#) needs inputs to have darker borders (goes for all forms on light backgrounds)
  - [This form](#) is an example of a form on a grey background, where the inputs don't have enough contrast vs the grey background. You can solve this by giving the input borders a color that contrasts better with the white background. Or by making the grey background darker. We're looking for a 3:1 contrast ratio

- Search field in header: borders as well as placeholder and visible text, icon, ... is too low contrast on both mobile and desktop. For the text, we need the 4.5:1 contrast ratio, for borders and icon it's enough to have 3:1

### Success criteria:

Resolve the contrast issues by changing the background or text color.

If you are using a [color contrast checker](#) (chrome dev tools is unreliable), these are the contrast & px values we need.

- The main rule for text  $\leq 24$  pixels, is that contrast needs to be at least 4.5:1

For certain text and UI elements with certain characteristics, a lower contrast of at least 3:1 applies:

- for large text ( $> 24$  pixels), it should be at least 3:1
- for bold text of at least 18,5 pixels, it should be at least 3:1
- UI components also 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.

### 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/WCAG21/Techniques/general/G145>
  - <https://www.w3.org/WAI/WCAG21/Techniques/general/G18>

## 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

- The Language submenu is not accessible via keyboard
- [Knowledge hub articles](#): There is a 'Share article' block that contains links without href attributes
  - `<a class="a2a_button_email"></a>`
  - `<a class="a2a_button_twitter"></a>`
  - `<a class="a2a_button_linkedin"></a>`

### Success criteria:

- While the 'About credendo' menu is accessible, the language menu is not. Make sure you can reach and control it via keyboard.
  - Also make sure the item links get focus to comply with Success Criterion 2.4.7: Focus Visible
  - Also test this works on mobile and a zoomed-in page (400%), where the mobile navigation (hamburger menu) appears.
- Knowledge hub articles: these links are elements from a share-function that works with Javascript. As such, you should be using [a disclosure pattern](#):
  - Wrap the hidden content in a div (those fake links) and make it hidden from screen-readers instead of just visually hidden. Eg. display: none works, or aria-hidden
  - Use the aria-attributes on the trigger and the content wrapper, as explained in the pattern
  - Make sure the links have a href attribute and it is filled in.
  - If the Share function is not used, don't print it in the html

### 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/menubar/>

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

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 in WCAG 2.1 as long as it has enough contrast with the normal state.

But we recommend using [the Focus Appearance criterion from WCAG 2.2 AAA](#) for extra clarity

### Problem

- Some things on the (dutch) homepage **missing a visible focus**:
  - Language menu items (they use the hover style but should we would advise using outlines, like is used on other items)
  - Buttons, eg. [of forms](#) (no focus), and [search form page](#) (becomes invisible apart from the outline)
- A focus that is **not clear enough**:
  - As mentioned in [the contrast criterion](#), input fields under certain conditions need better border contrast, same for placeholder text of search field, ...

### Success criteria

- Give decent focus styles (preferably an outline) to the focusable elements described above. For example:
  - For **buttons**: give them an outline or border with contrasting color (eg. like the text links do) and make sure the text doesn't disappear (on some buttons, the text color is the same as the button background when focussed)
  - For The links in **submenu's of the primary and secondary header navigation**, give these their own hover/focus style that is different from the normal state.

### 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 3.2.2 On Input (A)

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

### Problem

- Some form-like components don't contain a submit button but auto-submit option changes. This is a problem as there is no advance warning.
  - See the 'Country and risks insights' block on Homepage, as well as 'Find what you need' block and variants of this component that use the 'chosen' form element with autosuggest list, on various pages
    - Eg. <https://credendo.com/en/solutions?f%5B0%5D=categories%3A278>
    - Eg. <https://credendo.com/en/country-risk> (got same block on homepage too)
    - Eg. <https://credendo.com/en/solutions>
    - ...

### Success criteria:

- These 'chosen' forms are complex ajax-driven components (with autocomplete list and such). But we can do some things to make it more accessible:
  - **Either provide a submit button or Explain** how the component works, that making a choice will redirect you. Do this in the introductory text right above the element. Eg. "clicking the element shows an input field in which you can search and also opens a list of suggested results".
  - The 'Select a country or continent' item is a link with no href attribute. Since it doesn't function as a link but a trigger, it should be a button instead.
  - There seems to be some stray/unused HTML, that needs to be cleaned up: `<b></b>`
  - When making changes, make sure the user can still use the **keyboard to navigate** to the component: like reach the input field by tabbing and move to/through the open list.

### References:

- **Source:**
- **Understanding:** <https://www.w3.org/WAI/WCAG21/Understanding/on-input>
- **Techniques:** <https://www.w3.org/WAI/WCAG21/quickref/#bypass-blocks>
  - <https://www.w3.org/WAI/WCAG21/Techniques/html/H32>
  - <https://www.w3.org/WAI/WCAG21/Techniques/general/G13>
  - <https://www.w3.org/WAI/WCAG21/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

- **Contact form:**
  - While the forms show error info after submit, there is no link between the messages and the matching input field.
  - The input field themselves are also not marked as required using html attribute 'required'. There is only a visual indicator on the label (\*) with no further information

### Success criteria:

- You can link input to error message by using [aria-errormessage](#) and info relevant to input fields, using [aria-describedby](#)
- You should add the 'required' attribute on form fields that are required to fill in.
- Instead of just having the visual \* indicator, a (visually hidden) message will go a long way to warn screen reader users that there are required fields.

### 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 search form submit button (in header), has an aria-label that makes no sense. It says 'button' but should be 'Submit'.
- [The search form on the search page itself](#), has no label of the input field.
- There are several instances of divs using aria-label attributes. These attributes are only allowed on elements with distinct roles, eg. interactive elements like buttons or nav-tags in need of labeling.
  - `id="block-searchredirectblock"`
  - `<div id="block-rocketship-theme-starter-content"`
  - Most (if not all) block types
  - ...
- [In pages with document downloads](#), the alt attribute simply prints the file name. The description should be more descriptive of what will happen when clicked. Eg. 'open or download the document <title of file>'

### Success criteria:

- For the search submit: change the aria-label to say 'Submit' and we'd advise adding a title attribute with the same value.
- For the search form in the search page, add an aria-label on the input or a label-tag referencing the input.
- Do a search for aria-label in the page source and remove or move them for all the divs. You can:
  - Move the label to a more appropriate element: eg. if it's for a navigation (nav-tag), move the label to that nav-tab
  - Use a heading instead (can be visually hidden). Eg. for the wrapper of the search in the header.
  - Remove it. If it can't be repurposed in the ways described above, it's probably not needed for the end user. Eg. the breadcrumb has a label set on its nav-tab, there's no need to have an aria-label on the div that is wrapped around that. The same goes for all the block types, eg. image block. The label does not add any value. You could also argue a heading isn't really needed for the search block either.

### 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>

## Accessibility statement page

The site **requires an Accessibility statement page** to be compliant (and linked to, for example, from the footer of the site).

We recommend adding it once work has been done to fix the majority of the issues listed in this report. Dropsolid can help write this page.

If some aspects are not yet accessible, these have to be mentioned, alongside the WHY and what is planned to fix this.

### How to:

- Generators and assistance is available:
  - <https://accessibility.belgium.be/nl/invulassistent-toegankelijkheidsverklaring>

### What is (and isn't) part of that page:

- (en) [Developing an Accessibility Statement](#)
- (nl) [Informatie van de Vlaamse overheid](#)

### Examples of Accessibility Statement pages:

- FOD - IBZ
  - <https://ibz.be/fr/declaration-sur-laccessibilite>
  - <https://ibz.be/nl/toegankelijkheidsverklaring>
- DNS belgium:
  - <https://www.dnsbelgium.be/en/accessibility-statement>
  - <https://www.dnsbelgium.be/nl/toegankelijkheidsverklaring>