# Accessibility Review: Storyblok

## Management summary

Many [Web Content Accessibility Guidelines (WCAG)](https://www.w3.org/TR/WCAG21/) criteria were ensured and violations were recorded in this document.

We can clearly see that some measures were taken to make Storyblok accessible. But there still exist a lot of barriers that need to be removed and even UX topics that can be optimised (which in turn optimised accessibility as well).

The review started out in the default version of Storyblok; when I realised that there is a V2 preview, I continued in that mode, as this one is probably the one who will receive future updates. That's the reason why some findings are still for V1. I leave it in the document as the given insights can be generalised easily.

To offer actionable first steps, I only focused on **keyboard-only operability** and **colour contrasts** (using the [Colour Contrast Analyser](https://www.accessibility-developer-guide.com/setup/helper-tools/colour-contrast-analyser/) tool). To ensure **screen reader operability** would mean additional, quite intense testing that would require profound code improvements, which is probably out of the currently actionable scope.

## Source citation

The Internet is teeming with information, including on the topic of accessibility. In addition to official offers, there is also an abundance of academic publications up to private opinions. Depending on the focus of the authors, statements on the same topic may differ or even contradict each other. It is a challenge to check the validity of such information with regard to one's own project.

Accordingly, in the following we draw on two resources that have proven very useful. They address the topic of accessibility in a neutral, comprehensive, and solution-oriented manner: the **Accessibility Checklist 2.1** (<https://a11y.digitaldialog.swiss/de/>) and the **Accessibility Developer Guide** (ADG, <https://www.accessibility-developer-guide.com/>). Both are projects of the Swiss foundation "Access for All" (<https://access-for-all.ch/>); they were developed in cooperation with other stakeholders, such as the Swiss Confederation as well as renowned web agencies. Both resources are strictly based on the internationally recognized **Web Content Accessibility Guidelines 2.1** (WCAG, <https://www.w3.org/TR/WCAG21/>).

Much of the content of this document is based on the aforementioned resources. Citations are always linked, and we recommend following these links to read the content in detail (e.g. further explanations, examples, exceptions, as well as codes).

Josua Muheim, author of this document, was significantly involved in both projects during his many years of work for the "Access for All" Foundation.

## Keyboard focus visibility

**The keyboard focus must be clearly visible at all times. This enables people with visual impairments to navigate with the keyboard.**

If the keyboard focus is not or insufficiently visible, this makes navigation via keyboard considerably more difficult. The very finely displayed browser standard is sufficient as keyboard focus strictly according to WCAG (as long as it remains visible throughout all elements), but for many people it is not sufficiently well perceivable.

### Findings

The following shows an unfocused Add button:



CleanShot 2022-04-15 at 14.18.31@2x

The focused one only has a slightly darker background color:



CleanShot 2022-04-15 at 14.20.48@2x

Often, the keyboard focus is not visible at all, for example for the following checkboxes:



CleanShot 2022-04-15 at 14.22.00@2x

The focus in the main navigation is hardly visible:



CleanShot 2022-06-01 at 16.27.20@2x

The focus on the "Filter" and "Sort" dropdowns of the "Search in ..." input is not visible at all:



The focus in the "Sort" menu is hardly visible:



CleanShot 2022-06-01 at 16.42.41@2x

Most of the following interactive elements do not have a visible focus state:



**Action:** Make sure that the keyboard focus is highly visible at all times by setting a prominent style for it. A highly visible border distinguishes focused links, buttons, radio buttons, checkboxes, linked graphical elements, etc. The CSS attribute outline is particularly suitable for this, e.g. outline: 2px dotted black.

You may also want to look into the ADG's chapter [Colour is not enough](https://www.accessibility-developer-guide.com/knowledge/colours-and-contrast/colour-is-not-enough/).

## Keyboard operability / focusability

**Content and functionalities must be accessible and operable with the keyboard alone. This enables people who cannot operate pointing devices (such as a mouse or touchscreen) to reach all content and use all functions.**

Physically impaired people often cannot operate a mouse or similar pointing devices and are dependent on the keyboard. Assistive input devices also use the same as an interface. It is therefore necessary that all functionalities of a page can also be operated with a keyboard alone. This applies in particular to JavaScript widgets of various kinds (such as video players, date pickers, tooltips).

To learn more about this topic, please consult the ADG: [How to implement websites that are ready for keyboard only usage](https://www.accessibility-developer-guide.com/knowledge/keyboard-only/how-to-implement/), especially the parts [Focusability](https://www.accessibility-developer-guide.com/knowledge/keyboard-only/how-to-implement/#focusability) and [Keyboard operability](https://www.accessibility-developer-guide.com/knowledge/keyboard-only/how-to-implement/#keyboard-operability).

### Findings

#### Real-time inline-editing

Real-time inline-editing does not work for keyboard. The overlaid clickable containers cannot be focused by keyboard, and as such they can't be activated.



Also, the appearing menu cannot be controlled using keyboard.

To make the containers focusable, instead of <div>s, use <button> elements, as they can inherently be focused. If for some serious reason you must use <div>s, you can alternatively add a tabindex="0" to the it.

**Action:** To make the menu controllable, make sure all items are focusable and that they are in a natural DOM order (ie. they can intuitively be reached by pressing Tab or Shift+Tab). It is probably a good idea to focus the "Feature" element immeditaly when the overlay was clicked.

In addition, provide an additional menu item that allows to jump directly into the content editor on the right side (so keyboard users do not need to extensively hit Tab until they finally reach the element in question through the natural tab order, when wanting to edit its values).

#### Editor (CMS)

The "Filter" menu can only partly be controlled by keyboard only. For example when focusing a checkbox (menu item) and pressing Space or Enter, the menu closes, and nothing else happens. Also, Shift+Tab does not move the focus into the other direction (back).



CleanShot 2022-06-01 at 16.42.41@2x

The elements "Back to V1", "Help & Inspiration", and "My account" in the main navigation are not focusable at all:



CleanShot 2022-06-01 at 16.48.44@2x

It is impossible to focus any content element, thus it can't be accessed (viewed and edited) at all:



When tabbing through the interactive elements, there seem to be many focusable and visually hidden elements. This means that one has to tab dozens of times to move the focus from the search button to the "Name" field:



## Colour contrasts

**Texts and symbols must stand out from the background in color with sufficient contrast. This enables people with visual impairments to recognize and read text.**

People with impaired vision and defective vision depend on elements standing out clearly from the background. Therefore, texts and informative elements (such as symbols) must have a sufficiently high contrast value to the background.

Relevant are all texts of the content and of control elements. Their state changes (such as hover and focus) must also meet the requirements; however, there are no strict contrast requirements for distinguishability between the same.

Minimum contrast required:

* Normal font: contrast ratio of at least 4.5:1
* Large type (18pt or larger or 14pt + bold): Contrast ratio of at least 3:1

To learn more about this topic, please consult the ADG: [Colours and contrast](https://www.accessibility-developer-guide.com/knowledge/colours-and-contrast/), especially the part [Colour contrast for user interface components](https://www.accessibility-developer-guide.com/knowledge/colours-and-contrast/user-interface-components/).

### Findings

The "Search in spaces" input's placeholder has very low contrast:



CleanShot 2022-06-01 at 16.18.41@2x

The "Search in ..." input's borders have very low contrast:



The checkbox next to each menu item inside the "Filter" menu is hardly perceivable: some people will not see it and may have a hard time understanding that multiple options can be chosen.



CleanShot 2022-06-01 at 16.42.41@2x

## How to proceed

At [Nothing](https://www.nothing.ch), we have all the knowledge and skills to support you during the process of optimising your product regarding accessibility. We offer:

* In-depth **accessibility reviews** of your product according to [WCAG 2.1](https://www.w3.org/TR/WCAG21/), including concise advise how to fix any issues.
* Pragmatic **knowledge transferring** from our team to yours by holding workshops for designers, developers, and content producers (in person or remote).
* Active **collaboration** inside your code base through **parallel programming**.

You can find more information about our services on our website:

* [Our accessibility competences](https://www.nothing.ch/competences/accessibility/)
* [Accessibility first aid: When your website needs an upgrade, fast](https://www.nothing.ch/showcase/accessibility-first-aid-when-your-website-needs-an-upgrade-fast/)
* [Investment in inclusion: Accessibility for your audience](https://www.nothing.ch/showcase/invest-in-inclusion-accessibility-for-your-audience/)

Please reach out to us: manuel@storyblok.com