**Focus not obscured**

* WCAG
* Keyboard

August 1, 2025

The Web Content Accessibility Guidelines 2.2, or WCAG 2.2 for short, introduced eight new criteria. Two of them are related to the headache of keyboard users (and also the accessibility specialists community): The **obscuring of a focused element by other content**. These are the level AA criterion [2.4.11: Focus Not Obscured (Minimum)](https://www.w3.org/TR/WCAG22/#focus-not-obscured-minimum) and the level AAA one [2.4.12: Focus Not Obscured (Enhanced)](https://www.w3.org/TR/WCAG22/#focus-not-obscured-enhanced).

**Short explanation**

According to the level AA criterion 2.4.11, elements in a focused state must be at least *partially* visible when they are overlapped by others.

Exceptions to this criterion include elements activated by the user and components that can be moved around the page, for example, by dragging.

Another exception is the focus indicator itself, such as the button's outline. There is a separate criterion for this: 2.4.12 (AAA).

**Problematic patterns**

Both criteria prevent **navigation issues for keyboard users**. For example, they aim to eliminate situations where items from a dropdown menu are reachable via keyboard but are *not visible* at the same time.

Elements with a keyboard focus can also be obscured by:

* Other **dropdown elements** (combo boxes, hamburger menus, date pickers)
* **Sticky, fixed elements** (headers, footers, sidebars, cookie banners)
* **Pop-ups** (chatbot messages, notifications, tooltips, and other hints)
* **Non-modal and (pseudo)modal dialog boxes without focus trap** (cookie information or newsletter subscription)
* **Movable elements** (stickers on a virtual Kanban board).

**Common real-life cases**

Not all **dropdowns** are bad. The criteria consider two main cases, each with a dropdown. The first case is an *open-by-default* element, for example, a sidebar menu with navigation. The second situation is an incorrectly coded dropdown. For instance, cumbersome navigation with submenus opens with the keyboard and doesn’t close by pressing the Esc key.

It’s possible that **fixed navigation** with a lot of items doesn’t meet the minimum criterion. In this case, the focus problem becomes especially noticeable when you navigate with the keyboard to the top of the page.

**Fixed cookie banner** is another problematic design pattern. Banners with controls and descriptive texts are at risk of violating the level AA criterion. In many cases, a cookie banner is the last element on a page. Keyboard users need to go to the bottom of the page to close the cookie message.

*Cookies* are text files with data about the user, for example, about their browser or device. Website owners collect this data mainly for marketing purposes.

**Pop-ups** can also violate the minimum criterion 2.4.11. Annoying **tooltips** that automatically open when a related element receives focus, blocking part of the rest of the content until the very end. “Until the very end” means that only refreshing the page can help.

**Dialog boxes** are another level of pain. Most non-modal dialogs that are open by default and placed in the center of the screen will violate the level AA criterion 2.4.11. Again, the most common pattern here is cookie settings.

Just like with dropdowns, the criterion 2.4.11 allows **modal dialogs**. A modal dialog does not violate the criterion by itself, even if it’s not coded quite correctly. The user can always scroll the page with the modal dialog using a mouse or another pointing device. However, if the modal **completely** prevents keyboard focus from being set on elements behind it, that’s a violation.

A non-modal window doesn’t block the main page content. In contrast, a modal dialog prevents accessing the page while it’s open.

**Movable elements** fail the minimum criterion when they obscure focusable elements before they are moved by users.

**The case of semi-transparent elements**

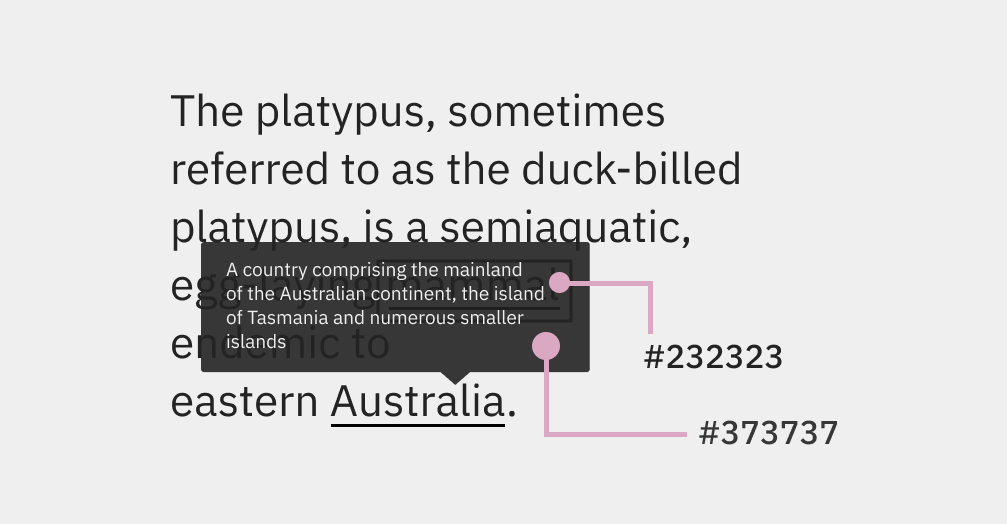
**Elements with glass effect or opaque styles**, such as semi-transparent tooltips and fixed navigation, are the second special case.

A special case described by 2.4.11 is **elements with a glass effect or with an opaque background**. Two of the examples are semi-transparent tooltips and fixed navigation.

Here, to meet the minimum criterion, a “translucent” component must also meet another criterion, [1.4.11: Non-text Contrast](https://www.w3.org/TR/WCAG22/#non-text-contrast). For this, you check the contrast ratio between the color of the interactive element and the background of the overlaying one.

Imagine a semi-transparent tooltip that completely obscures a link. You need to compare the link color resulting from blending with the tooltip background and the background of the pop-up hint itself.

For example, the link color after blending with the background is #232323 (young night), and the background of the tooltip is #373737 (muted gray). The contrast ratio between colors is 1.32:1. Thus, criterion 2.4.11 is not met even if the partially hidden element is slightly visible. In cases like this, what “slightly” means is clearly defined by other criteria.

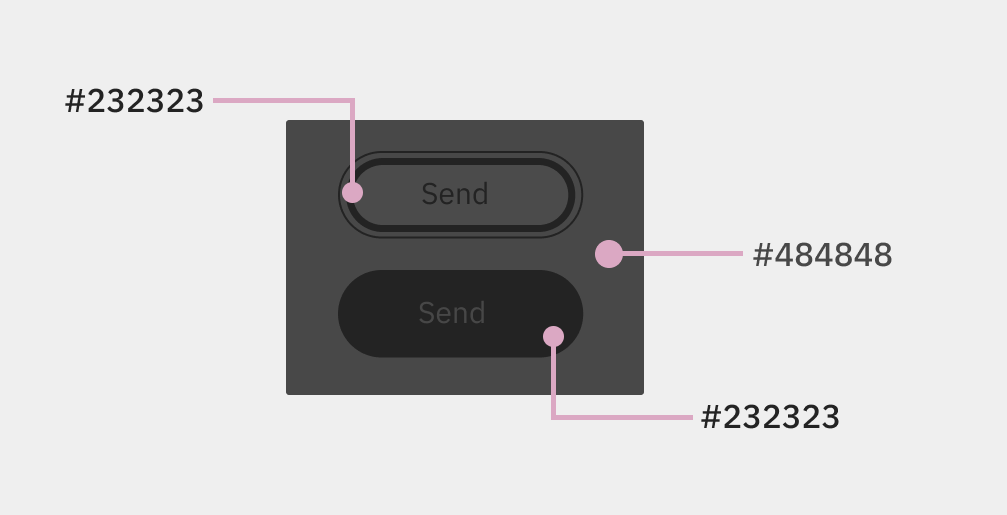


An opaque tooltip example.

In this imaginary example, we didn’t compare the color of the **focus indicator**. The only two cases we care about the appearance of the indicator are:

* If the focus indicator is inside the focused element
* If the element in a focus state changes significantly.

In the case of a button with an inner outline, compare the color of the outline with the background of the overlapping element. For a button that, when focused, *changes its background*, you need to compare its new background color with the background of the other element.

Two types of focused buttons.

**Complicated cases**

It would seem that everything is clear. Nevertheless, while studying the new criteria, I came up with a **couple of questions**:

* How should we check **card elements** with images, headings, and short descriptions? For example, a link containing both text and graphics (or only graphics), where only part of the image is visible.
* What should be done with a **focused link** when its text blends into the background of the page and **only the outline is visible**? After all, even if the link is partially overlapped, we can’t see *the element itself*.

At the time of writing this text, I haven’t found any clear answers to these questions in the WCAG documentation.

**Meaning for users**

As I mentioned at the beginning of the article, partially or completely obscured elements in the focus state are a barrier for **keyboard users**.

**Screen magnifier users** who combine them with keyboard navigation also need to be able to see elements in a focused state.

*A screen magnifier* is an assistive technology that enlarges specific areas of the screen.

It's also important for **users of alternative inputs**. They are still *visually* navigating through interface. For example, during voice control or eye tracking. Don’t forget about those **who use pointers** for navigation outside of the usual mouse, like styluses, head pointers, or joysticks.

Finally, it’s also essential for **neurodivergent users** with disabilities affecting memory and attention. For example, people with Attention Deficit Hyperactivity Disorder (ADHD), Long COVID, or Traumatic Brain Injuries (TBIs). Such users may get distracted more often than others and forget their current position and recent actions on the page.

**How to meet the criteria**

The simplest solution is **not to fix cumbersome navigation in the header and footer**, especially on small screens. You can save space by reducing the sticky header height during scrolling.

The CSS property scroll-padding could be handy. The property **increases the distance** between the fixed element and the rest of the page content.

Don’t abuse **automatically opening non-modal dialogs and interface hints** that can’t simply be closed with Esc. If you have a lot of free space along the edges of the page, such elements can be safely placed there.

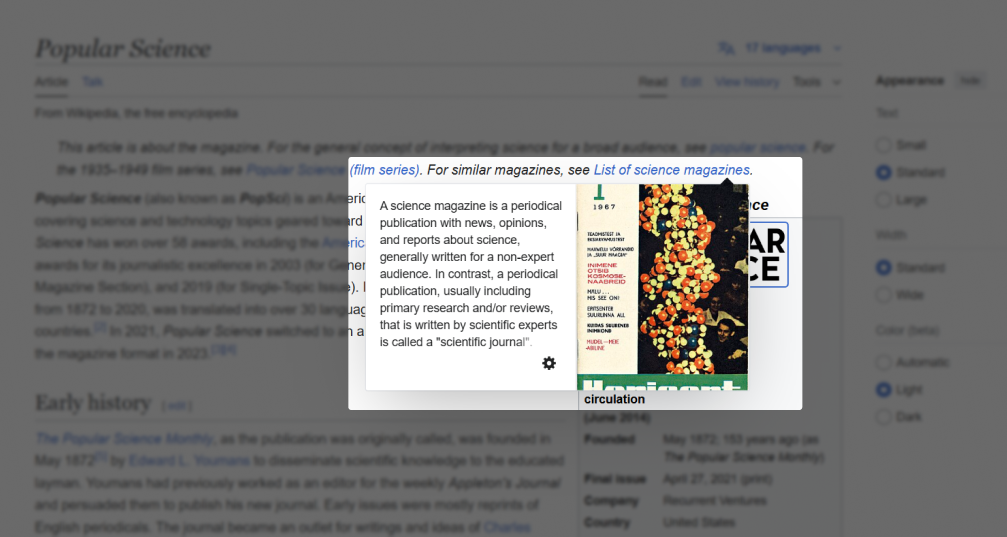
Code **modal dialogs** correctly. Don’t allow keyboard users to go to the page without pressing dialog buttons or keyboard keys to close the window.

To solve the **problem with cookie messages that cover everything**, you can

* Place it at the top of the page, and don’t fix it at all
* Show it as the *true* modal dialog.

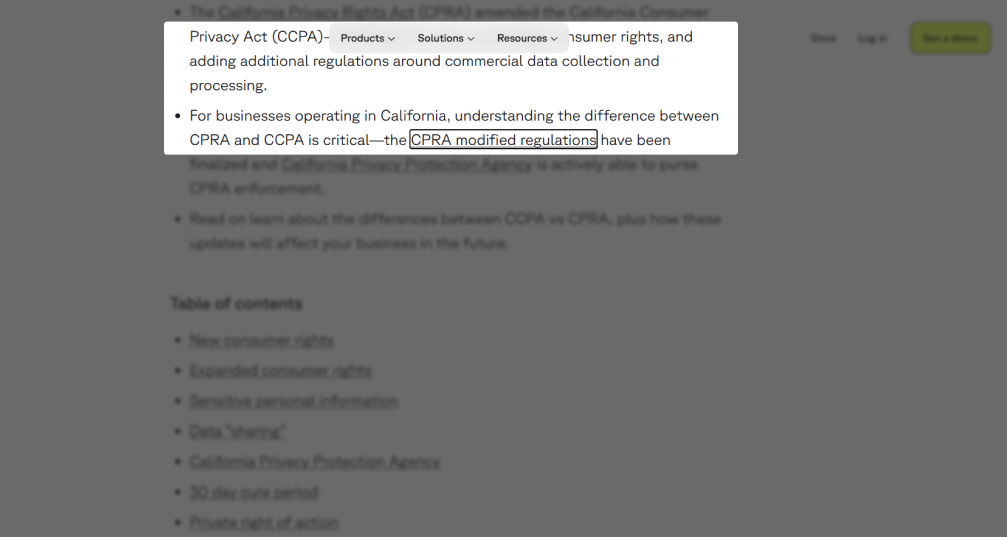
**Meet-the-criteria examples**

**Encyclopedia link previews**. When you focus on links in Wikipedia, tooltips with images and text appear. The open tooltip partially obscures some links. The element disappears when you go to a new link and cannot be closed with Esc.

The “Popular Science” magazine article on Wikipedia.

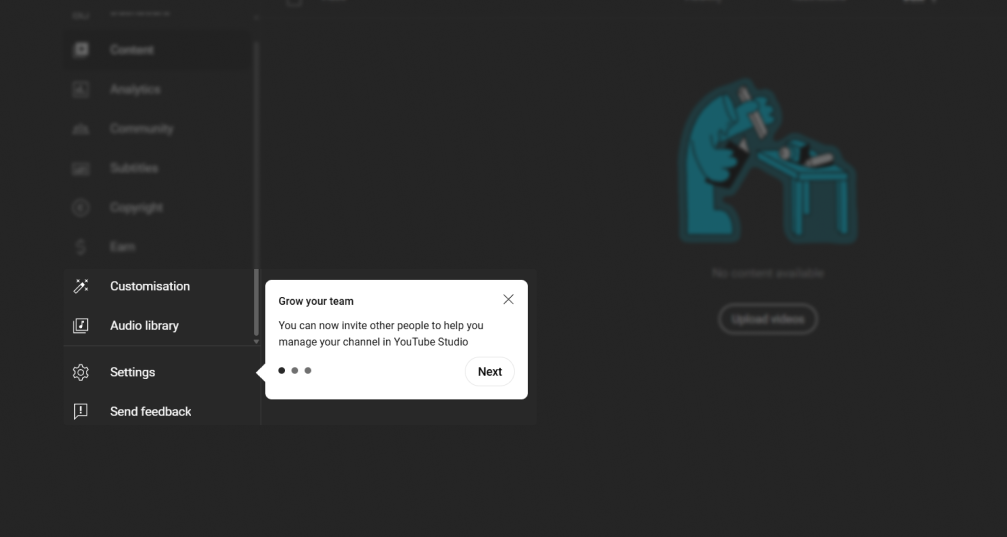
Since the tooltip only partially obscures elements in a focus state, the page **complies** with the criterion 2.4.11: Focus Not Obscured (Minimum).

**Blog fixed header**. In the [Transcend blog](https://transcend.io/blog), the fixed header is moving when you scroll the page. If you use the keyboard for page navigation, the header doesn’t obscure controls.

The Transcend blog post.

The page **complies** with the level AAA criterion 2.4.12: Focus Not Obscured (Enhanced).

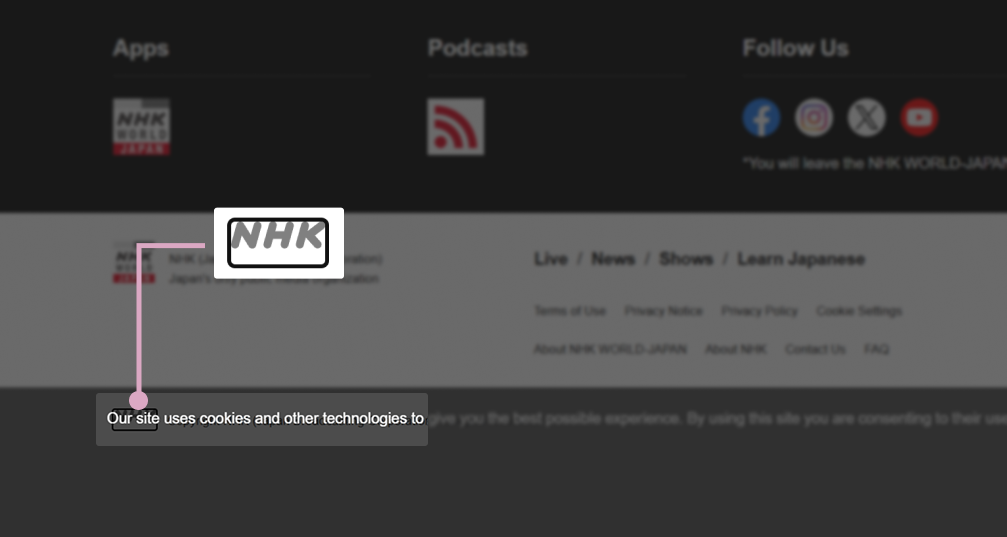
**YouTube Studio hints**. In the main tab of YouTube Studio, the notification with an interface hint automatically appears. This element doesn’t cover active links. The page **pass** the level AA criterion 2.4.11.

YouTube Studio.

**Not-complying examples**

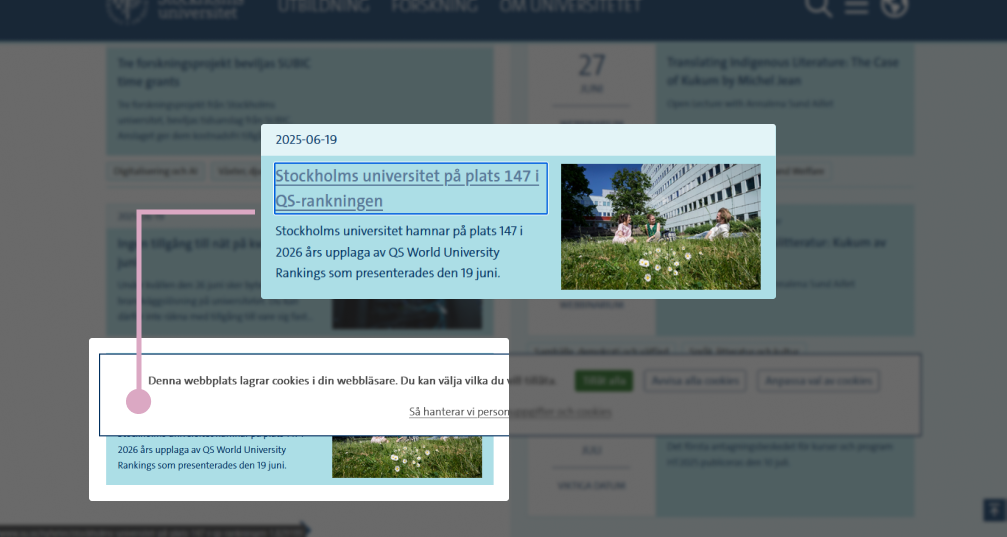
**NHK World Japan and the semi-transparent cookie banner**. On the homepage of the [NHK World Japan](https://www3.nhk.or.jp/nhkworld/) broadcasting company, a cookie banner with a semi-transparent background is fixed at the bottom of the page. Some controls are partially hidden behind the banner when navigating the page using the keyboard.

The banner *completely* obscures one of the footer links. Since the **banner background is semi-transparent**, we need to check the contrast. The link text color is #808080 (gray), and the cookie banner background is #4c4c4c (quartz gray). The contrast ratio between the two colors is 2.17:1, which is **below the minimum value** for non-text elements (3:1). Because of this, the website **doesn’t meet** the level AA criterion 2.4.11.

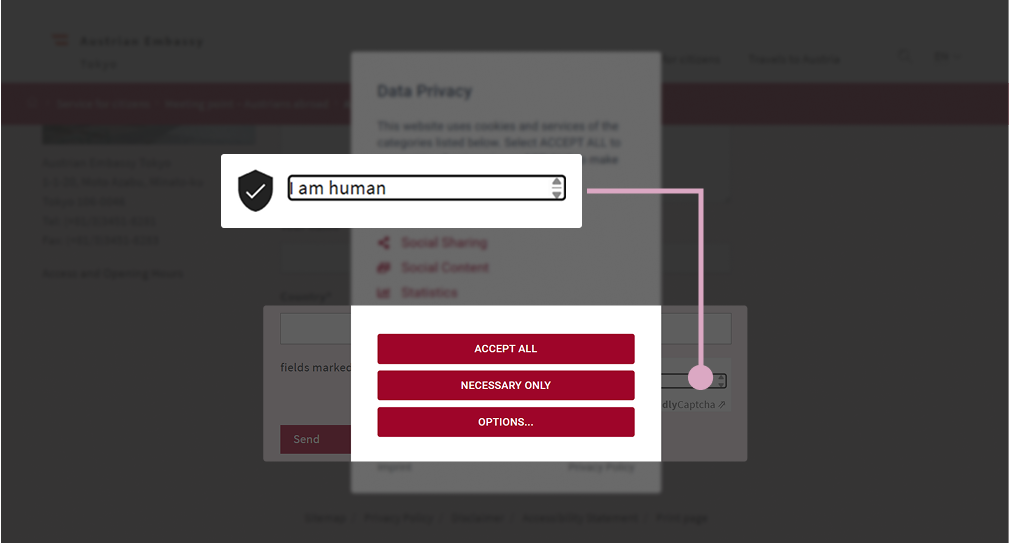
The NHK World Japan homepage.

**University website with a non-modal cookie banner**. On the [Stockholm University](https://www.su.se/) website, the non-modal dialog with cookie information is centered on the screen.

The non-modal dialog obscures some focused links completely. Based on that result, the website **doesn’t meet** the minimum criterion 2.4.11.

The Stockholm University homepage.

**(Pseudo-)modal cookie banner**. When you first open the homepage of the [Austrian Embassy in Tokyo](https://www.bmeia.gv.at/en/austrian-embassy-tokyo), a dialog with a cookie message appears. Visually, because of the darkened background, the dialog looks modal. However, you can still navigate through the page with the keyboard while the dialog is open. As a result, the some fields labels, links, and buttons are completely obscured. This means the page **doesn’t meet** the level AA criterion 2.4.11.

The Austrian Embassy in Tokyo homepage.

**Test methods**

Both criteria can be tested **automatically** and **manually**.

**Bookmarklets** can help with automated testing. That is, they can *theoretically* help. So far, I haven’t come across any **existing solutions** that simultaneously display focus styles for all controls and bring the necessary components to the interface top layer.

A temporary semi-automatic solution is the single bookmarklet [Show focus styles](https://code.jasonmorris.com/bookmarklets/show-focus-styles/) and manually open windows, dropdowns, and similar elements.

For **manual testing**, you’ll need a keyboard. The approximate testing steps are as follows:

1. Make sure there are **controls** on the page (buttons, links, etc.)
2. Find overlay elements
3. Navigate between controls using Tab and Shift Tab
4. Repeat the previous steps for other breakpoints.

Additional steps for controls **obscured by opaque top layer elements**:

1. Find out the **background color of the top element**
2. Determine the **focused element color**. Pick the mix of its original color with the background of the overlaying component (without the focus indicator's color)
3. Check the **contrast ratio** between two colors.

You can test color ratio with color contrast checking tools. A few names of them are [TPGi’s Colour Contrast Analyser (CCA)](https://www.tpgi.com/color-contrast-checker/) and [Accessible Web’s WCAG Color Contrast Checker](https://accessibleweb.com/color-contrast-checker/).

If every interactive element in a focus state is visible completely or partially, the page **meet** 2.4.11: Focus Not Obscured (Minimum) (Level AA).

If semi-transparent elements completely obscure controls in a focus state, and the contrast ratio between two colors is 3:1 and higher, the page also **meet** the level AA criterion.

When no focused element is obscured by others, the page **meet** 2.4.12: Focus Not Obscured (Enhanced) (Level AAA).

**Further reading**

* [2.4.11 Focus Not Obscured (Minimum) — WCAG 2.2](https://www.w3.org/WAI/WCAG22/Understanding/focus-not-obscured-minimum.html)
* [2.4.12 Focus Not Obscured (Enhanced) — WCAG 2.2](https://www.w3.org/WAI/WCAG22/Understanding/focus-not-obscured-enhanced.html)
* [Cookie banner — GOV.UK Design System](https://design-system.service.gov.uk/components/cookie-banner/)
* [2.4.11: Adversarial Conformance](https://adrianroselli.com/2023/10/2-4-11-adversarial-conformance.html) by Adrian Roselli
* [How to test 2.4.11: Focus Not Obscured (Minimum)](https://www.tpgi.com/how-to-test-2-4-11-focus-not-obscured-minimum/) by TPGi
* [Prevent focused elements from being obscured by sticky headers](https://www.tpgi.com/prevent-focused-elements-from-being-obscured-by-sticky-headers/) by TPGi.