

# **Searching accessible components**

Sami Keijonen

# Recourses

- [Guide to accessible components in Smashing Magazine.](#)
- [WAI-ARIA practises.](#)
- [Scott O'Haras accessible components.](#)

# Components like (1/2)

- Focus styles
- Navigation
- Icon only components (social links)
- “Cards” (image, heading, text linking to article)
- Modal dialogs

# Components like (2/2)

- Accordion
- Tabs
- Styling checkboxes and radio buttons
- “Autocomplete” (dynamic search results)

# A11y Figma Kit

- [A11y Figma Kit](#) for inspiration and reminder.
  - [Article explaining the a11y Figma Kit.](#)
- [Accessibility plugins for Figma.](#)

# Focus styles

- All interactive elements (links, form elements etc.) should have focus styles.
  - Crucial for keyboard users.
- Remember to style focus styles in design process (not just hover).

# Focus styles

Color contrast?

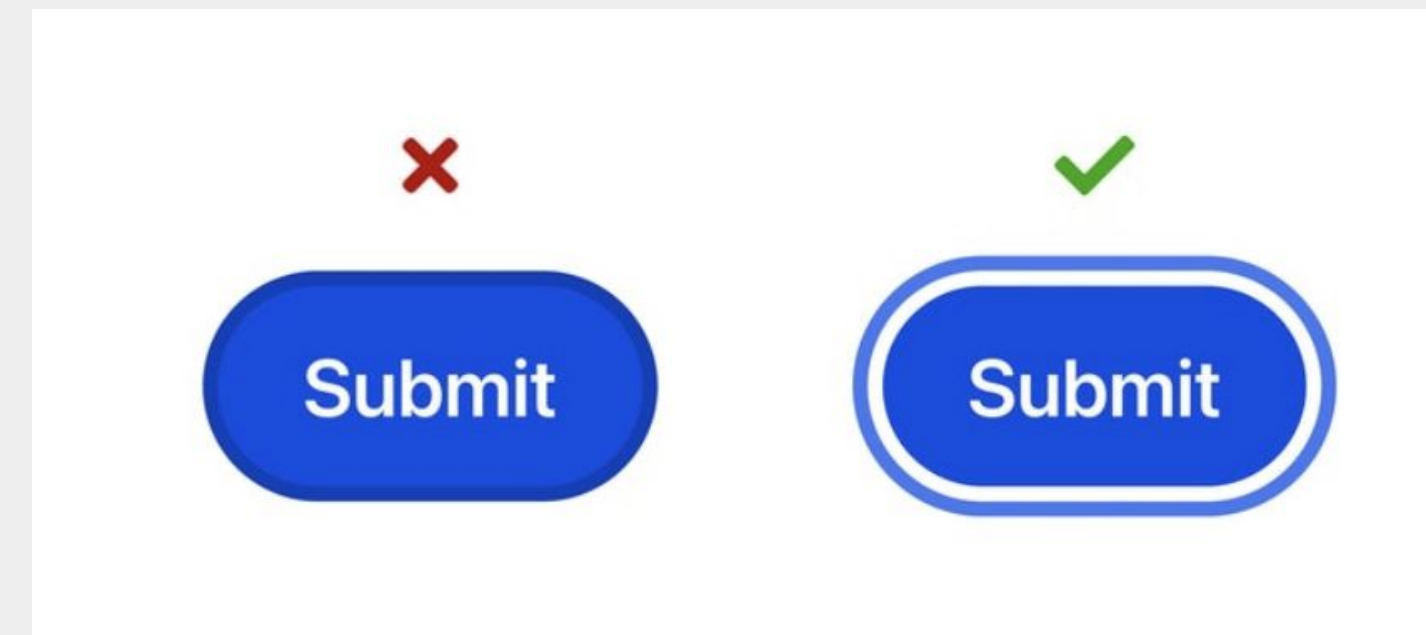
In WCAG 2.1 there is only color contrast 3:1 guideline.

```
:focus {  
  outline: 0.25rem solid #222;  
}
```

In 2.2 draft there are also guidelines for change of contrast and minimum area.

# Focus styles

## Good examples



Note that if box-shadow is used, still use transparent outline for WIN high contrast mode.

Never use `:focus { outline: 0 } CSS`

```
:focus {  
  box-shadow: ...  
  outline: 0.25rem solid transparent;  
}
```



# Focus styles

What about when focus styles are triggered also using mouse click?

[More about focus styles.](#)

**:focus-visible** is for that and now also Safari 14.1 supports it.

```
// Base focus styles
```

```
:focus {  
  outline: 0.25em solid $color;  
  outline-offset: 0.25em;  
}
```

```
// Be careful to remove focus styles  
// for “mouse” users.
```

```
:focus:not(:focus-visible) {  
  outline: none;  
}
```

# Navigation

Start with semantic markup

```
<nav>  
  <ul>  
    <li><a href="url">Link 1</a></li>  
    <li><a href="url">Link 2</a></li>  
  </ul>  
</nav>
```

# Navigation

If there are several navigation, it's good to add name using **aria-label** attribute.

Word “nav” is not needed in the name, it's exposed already to screen readers.

```
<nav aria-label="Main">  
  <ul>  
    <li><a href="url">Link 1</a></li>  
    <li><a href="url">Link 2</a></li>  
  </ul>  
</nav>
```

# Navigation

On smaller screens there is usually toggle button to open and close the navigation.

If possible, toggle `<button>` is good to be inside the nav.  
(Why?)

```
<nav aria-label="Main">  
  <button aria-expanded="false">  
    Menu  
  </button>  
  <ul>  
    <li><a href="url">Link 1</a></li>  
    <li><a href="url">Link 2</a></li>  
  </ul>  
</nav>
```

# Navigation

Toggle `aria-expanded` value (true or false) to indicate if the nav is open.

Use CSS to show or hide the nav items. For example using `is-opened` class.

```
<nav aria-label="Main">
  <button aria-expanded="true">
    Menu
  </button>
  <ul class="is-opened">
    <li><a href="url">Link 1</a></li>
    <li><a href="url">Link 2</a></li>
  </ul>
</nav>
```

# Navigation

Ask at least these questions

- How to style active page which uses `aria-current="page"`?
  - Underline is commonly used (color alone does not meet WCAG)
- Do you need submenus?
  - Are there sub sub menus?
  - How does sub menus look like?
- How navigation works on smaller screen?
  - “Hamburger” icon nav?
  - Is it full screen → Remember to trap focus inside the nav → visibility: hidden to other elements might be enough

# Navigation

Ask at least these questions

- [How does sub menus open?](#)
  - By click or by hover?
  - If by hover, how submenus open with keyboard?
    - For example `<button aria-label="submenu"><svg ...></button>` to open submenus
  - How submenus open on smaller screens?
    - All submenus at the same time or one by one?

# Navigation

Ask at least these questions

- Is the nav fixed at the top?
  - If yes, is it also fixed on smaller screens?
- Does something happen when user scrolls?
  - Revealing and hiding nav?
- How does animations work (sub menus, “hamburger”)?



# Navigation

## Examples

- [In Praise of the Unambiguous Click Menu](#)
- [Link + Disclosure Widget Navigation](#)

# Icon only components

## [Like social links](#)



- In most cases SVG icon is decorative and can be hidden for assistive technology.
- Always provide accessible name to link (or button).
  - For example using aria-label or **visually-hidden** class.
  - In WP it's screen-reader-text.
  - [sr-only class example in HTML5 boilerplate.](#)

# Icon only components



```
<a href="https://twitter.com">  
  <svg  
    aria-hidden="true"  
    focusable="false"  
    ...  
  </svg>  
  <span class="screen-reader-text">  
    Twitter  
  </span>  
</a>
```

# Icon only components

Close button



Same applies for example Search or Close buttons.

```
<button aria-label="Close">  
  <svg  
    aria-hidden="true"  
    focusable="false"  
    ...  
  </svg>  
</button>
```

# Cards

Gets complicated when the whole card “needs” to be clickable.



VALKKUNURKKA

## Viisi vinkkiä ketterään arkeen

Ketterät ketut jakavat oivalluksiaan arjen sujuvoittamisesta vertaistuen ja tiimioppimisen avulla.

# Cards

## Examples

- Do not put image, title, and text inside the same link.
  - That would be bad experience for screen reader users with long link name.
- Instead [link can be only inside heading](#) and use `a::after {}` pseudo class to make whole card as clickable area.
  - Note that selecting text becomes hard.
- [More details in inclusive components.](#)

# Cards

Alternative where image link and read more link is hidden from assistive technology and keyboard users (not whole card is a link).

```
<article>
  <a href="URL" aria-hidden="true"
tabindex="-1">
    <img alt="" src="">
  </a>
  <h3><a href="URL">Headline</a></h3>
  <p>Some intro text</p>
  <a href="URL" aria-hidden="true"
tabindex="-1">Read more</a>
</article>
```

# Modal dialogs

[Details on WAI-ARIA practises](#)

the window and interact with other parts of the page when they should not be able to. This is partially due to the way screen reader software interacts with the browser.

To open a modal dialog, you need to use the `aria-modal` attribute. On a page with a modal dialog, the `aria-modal` attribute is used to indicate that the dialog is the only focusable content on the page. This is done by setting `aria-modal="true"` on the dialog's root element. This is done in the following example:

Fill in the ridiculously small form below to receive our ridiculously cool newsletter!

**Email (required)**

tested using Cypress to make sure it works as intended.

**No more excuse now. Make your dialog windows accessible.**



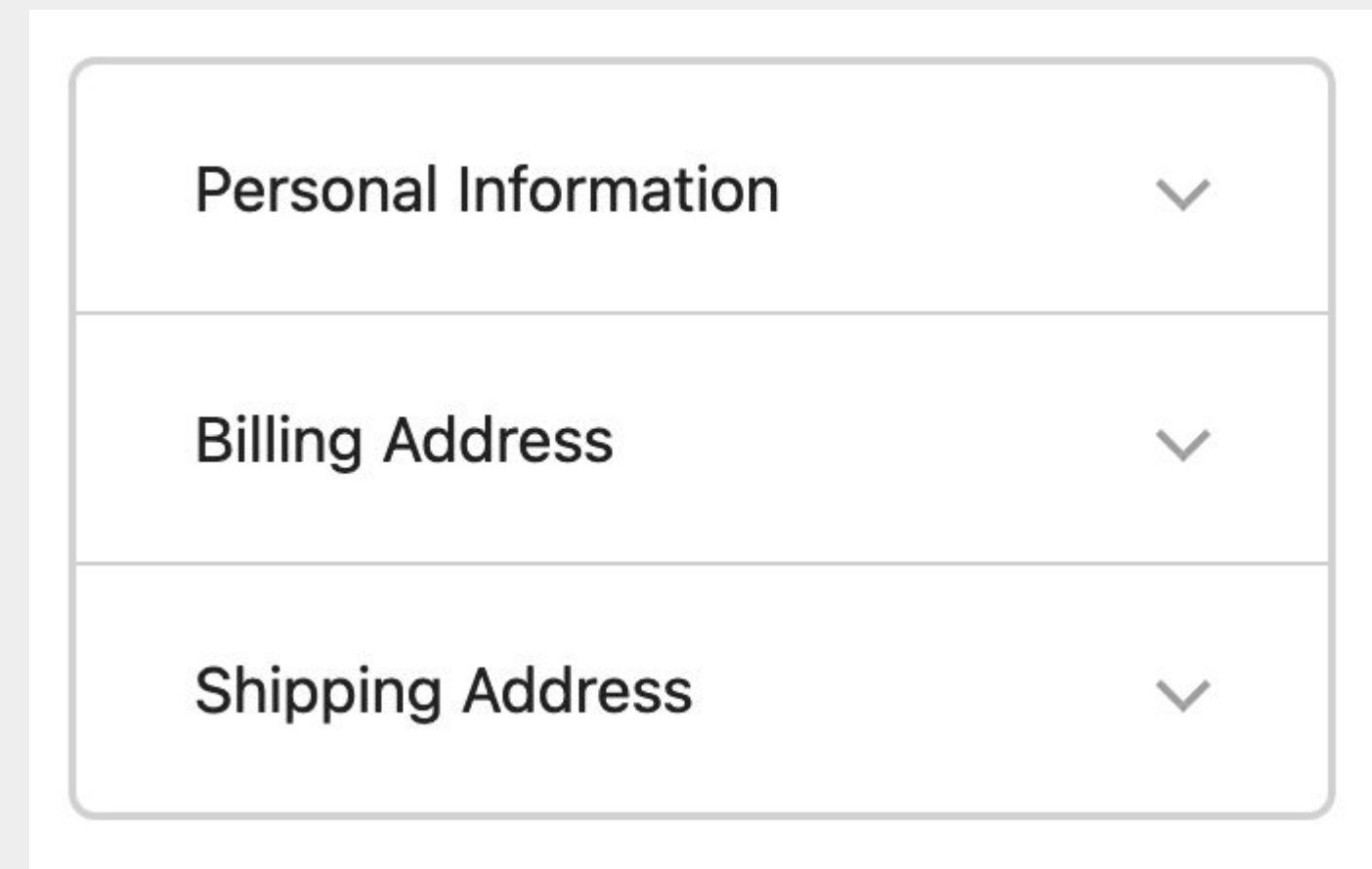
# Modal dialogs

[Great JS script by Kitty.](#)

- Focus moves to the modal when opened.
- Focus moves back to the button which opened it when closing the modal.
- Esc key and clicking outside the modal closes the modal.
- Focus traps inside the modal.
- Dialog should have a name, e.g. a h2 heading, referenced from wrapper element using aria-labelledby.
- [Note that HTML5, in this case <dialog>, is not always the best solution.](#)

# Accordion

[Accordion details on WAI-ARIA practises](#)



Personal Information

Billing Address

Shipping Address

# Accordion

## Base questions

- Is accordion content available when JS is not around? Should be.
- Are there animations when accordion opens / closes?
- Does icon indicate open / close state? It should.
- What content is allowed inside the accordion?
- Good for FAQs, don't overuse just to save vertical space.

# Accordion

## Minimum requirements

- Use button for opening and closing the accordion content: `<button aria-expanded="false">Accordion item</button>`
  - Toggle between true/false
- Focus should not move into accordion content before it's opened.

# Accordion

Example JS scripts

- [10up accordion component.](#)
- [Oncode accordion.](#)

# Tabs

[Tabs details on WAI-ARIA practises](#)

Nils Frahm

Agnes Obel

Joke

Nils Frahm is a German musician, composer and record producer based in Berlin. He is known for combining classical and electronic music and for an unconventional approach to the piano in which he mixes a grand piano, upright piano, Roland Juno-60, Rhodes piano, drum machine, and Moog Taurus.

# Tabs

## Base questions

- Is Tabs content available when JS is not around? Should be.
- How does selected Tab design look like visually?
- How does Tabs look like on smaller screens?
- What content is allowed inside the Tabs?

# Tabs

## Minimum requirements

- Use correct roles (tablist, tab, tabpanel).
- Use button for Tab items: `<button aria-selected="false">Tab item</button>`
  - Toggle between true/false.
- Left and right arrow keys should move focus between Tab items.
  - Tab key moves focus inside the Tab content.



# Tabs

Example JS scripts

- [Heydons example Tabs](#)
- [10up Tabs component](#)

# Styling checkboxes and radio buttons

[Styled form controls by Scott O'hara](#)

What type of accessibility issues you see most often?

- There are no issues
- Focus styles are not present
- HTML markup is used in bizar way. Also what happens if the label text is very loooooooooong, like this one.

What type of accessibility issues you see most often?

- There are no issues
- Focus styles are not present
- HTML markup is used in bizar way. Also what happens if the label text is very loooooooooong, like this one.

# Styling checkboxes and radio buttons

Minimum requirements

- Remember to still use correct semantic HTML: `<input type="checkbox">` or `<input type="radio">`.
- Remember to still have labels.
- Remember to still have focus styles.
- Remember to still have design for “checked” state visually.
- Base trick is to visually hide the checkbox by setting opacity almost zero and probably `position: absolute` for removing from the flow.

# Autocomplete

[Combobox details on WAI-ARIA practises](#)

Select your country

fi

Fiji

Finland

# Autocomplete

Comboboxes can be tricky  
with or without autocomplete

- Do you really need autocomplete, would `<select>` still be OK?
- [Deep dive in combobox and examples.](#)
- [Excellent JS script by GOV UK.](#)