

SvelteKit & A11y:

Intro & Keyboard Navigation

Antonio Sarcevic
Full-Stack-Developer





Antonio Sarcevic

Full-Stack-Developer at Datawrapper

antonio@datawrapper.de

@sarcevic@mastodon.online

- Working as a Developer since 2019
- BSc in Media and Communications Informatics
- Co-Creator of SvelteLab.dev

SvelteKit A11y

10 min

Web Accessibility

A quick intro into the world of web a11y

10 min

Accessibility in Svelte + Kit

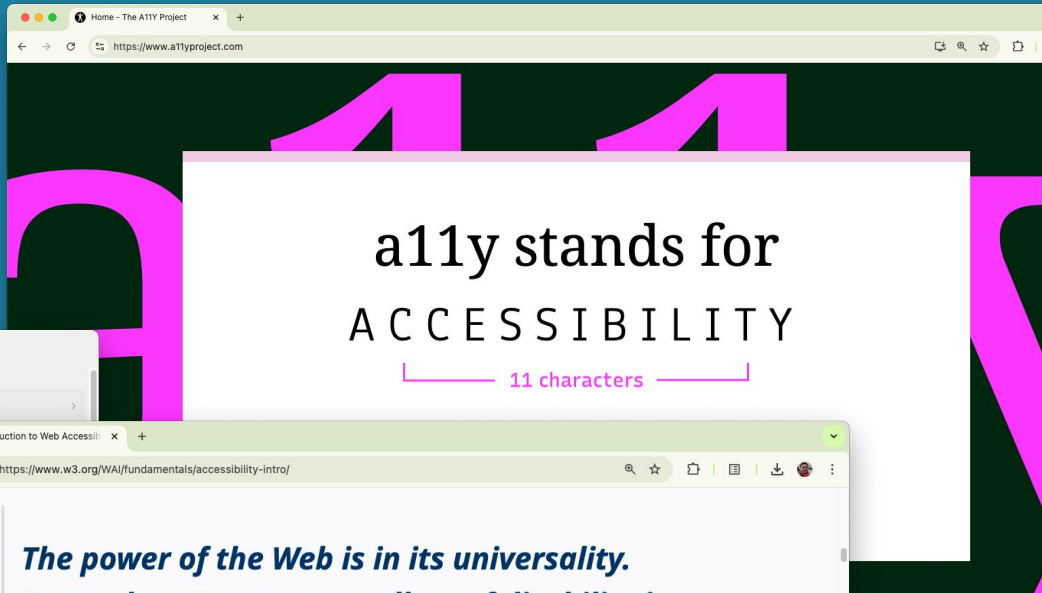
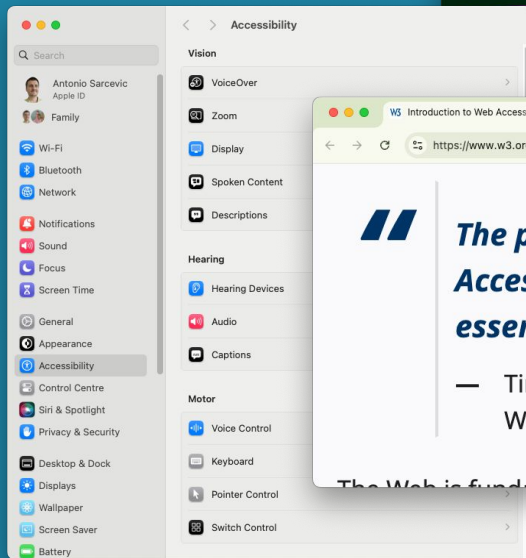
Where does Svelte help us?

20 min

Incrementally improving

Showcasing the recent **keyboard navigation improvements** we made in our Archive

Web Accessibility: A quick intro



What is Web Accessibility?

- Designing and developing so **people with disabilities can use sites and apps**
- Many different disabilities **affect access to the Web**
- **Accessibility benefits everyone!**

```

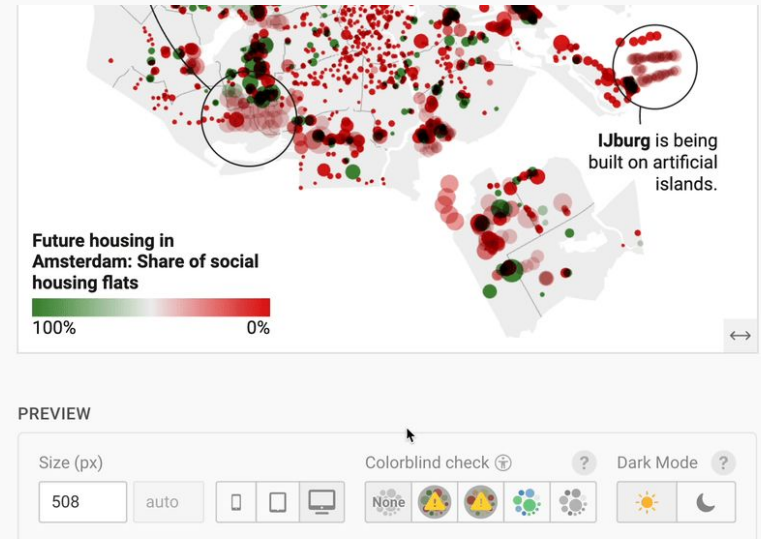
```

Accessibility is Important

- Web access **important** for a lot of **aspects of life**
- Access to information and communications technologies defined as **basic human right**
- **Strong business case** for accessibility
- A11y issues are **real bugs and technical debt**
- Web accessibility **required by law** in more and more cases
 - EU Web Accessibility Directive
 - European Accessibility Act

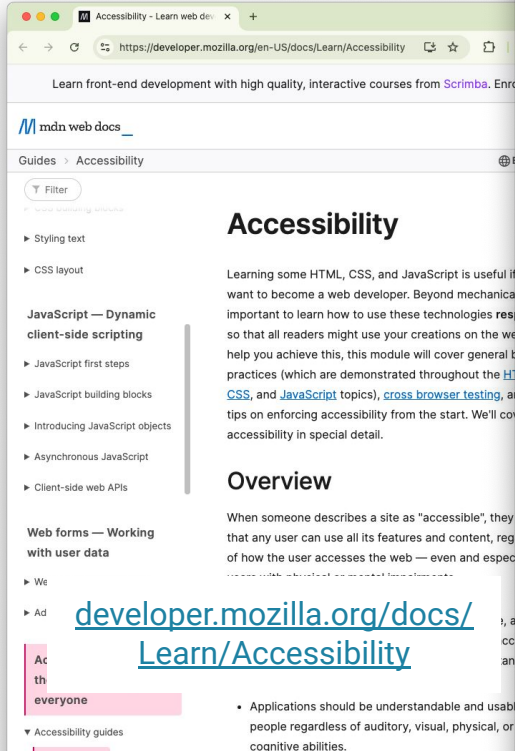
Accessibility @ Datawrapper (Visualizations)

- Colorblind check
- Alternative Text
- Semantic HTML
- Keyboard Navigation



Web Accessibility

Further reading



Learn front-end development with high quality, interactive courses from Scrimba. Enter

mdn web docs

Guides > Accessibility

Filter

- Styling text
- CSS layout
- JavaScript — Dynamic client-side scripting
- JavaScript first steps
- JavaScript building blocks
- Introducing JavaScript objects
- Asynchronous JavaScript
- Client-side web APIs

Accessibility

Learning some HTML, CSS, and JavaScript is useful if you want to become a web developer. Beyond mechanics, it's important to learn how to use these technologies responsibly so that all readers might use your creations on the web. To help you achieve this, this module will cover general best practices (which are demonstrated throughout the [HTML](#), [CSS](#), and [JavaScript](#) topics), [cross browser testing](#), and tips on enforcing accessibility from the start. We'll cover accessibility in special detail.

Overview

When someone describes a site as "accessible", they mean that any user can use all its features and content, regardless of how the user accesses the web — even and especially users with disabilities.

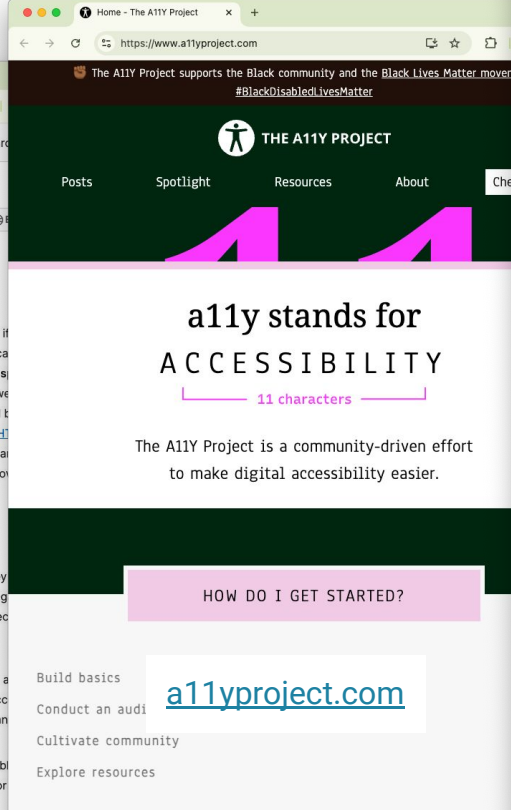
Web forms — Working with user data

- We
- Ad

developer.mozilla.org/docs/Learn/Accessibility

Accessibility guides

- Applications should be understandable and usable by people regardless of auditory, visual, physical, or cognitive abilities.



Home - The A11Y Project

<https://www.a11yproject.com>

The A11Y Project supports the Black community and the Black Lives Matter movement. #BlackDisabledLivesMatter

Posts Spotlight Resources About

a11y stands for ACCESSIBILITY

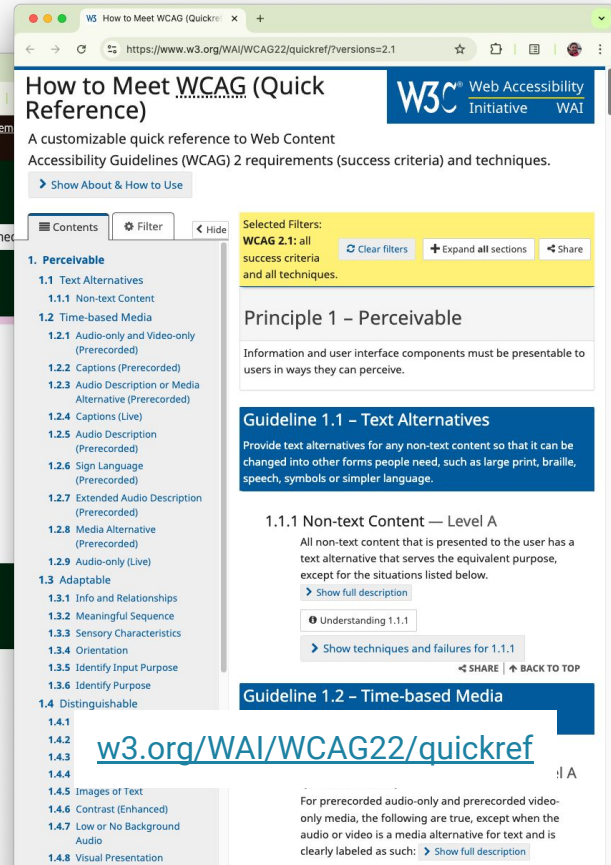
11 characters

The A11Y Project is a community-driven effort to make digital accessibility easier.

HOW DO I GET STARTED?

- Build basics
- Conduct an audit
- Cultivate community
- Explore resources

a11yproject.com



W3 How to Meet WCAG (Quick Reference)

<https://www.w3.org/WAI/WCAG22/quickref?versions=2.1>

W3C Web Accessibility Initiative WAI

How to Meet WCAG (Quick Reference)

A customizable quick reference to Web Content Accessibility Guidelines (WCAG) 2 requirements (success criteria) and techniques.

Show About & How to Use

Contents Filter

Selected Filters: WCAG 2.1: all success criteria and all techniques. Clear filters Expand all sections Share

Principle 1 – Perceivable

Information and user interface components must be presentable to users in ways they can perceive.

Guideline 1.1 – Text Alternatives

Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.

1.1.1 Non-text Content — Level 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.

Show full description

Understanding 1.1.1

Show techniques and failures for 1.1.1

Guideline 1.2 – Time-based Media

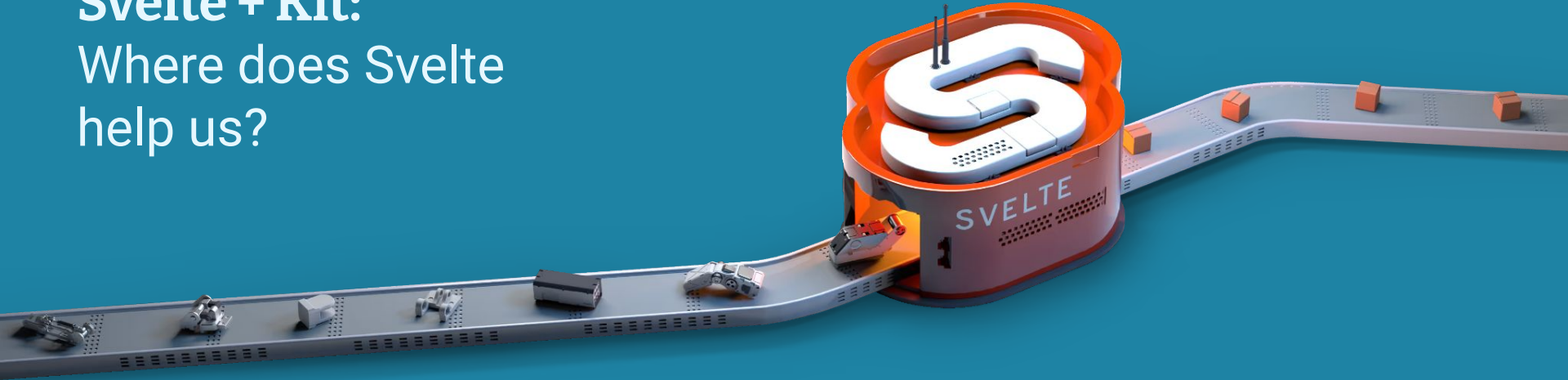
For prerecorded audio-only and prerecorded video-only media, the following are true, except when the audio or video is a media alternative for text and is clearly labeled as such: Show full description

1. Perceivable

- 1.1 Text Alternatives
 - 1.1.1 Non-text Content
- 1.2 Time-based Media
 - 1.2.1 Audio-only and Video-only (Prerecorded)
 - 1.2.2 Captions (Prerecorded)
 - 1.2.3 Audio Description or Media Alternative (Prerecorded)
 - 1.2.4 Captions (Live)
 - 1.2.5 Audio Description (Prerecorded)
 - 1.2.6 Sign Language (Prerecorded)
 - 1.2.7 Extended Audio Description (Prerecorded)
 - 1.2.8 Media Alternative (Prerecorded)
 - 1.2.9 Audio-Only (Live)
- 1.3 Adaptable
 - 1.3.1 Info and Relationships
 - 1.3.2 Meaningful Sequence
 - 1.3.3 Sensory Characteristics
 - 1.3.4 Orientation
 - 1.3.5 Identify Input Purpose
 - 1.3.6 Identify Purpose
- 1.4 Distinguishable
 - 1.4.1
 - 1.4.2
 - 1.4.3
 - 1.4.4
 - 1.4.5 Images of Text
 - 1.4.6 Contrast (Enhanced)
 - 1.4.7 Low or No Background Audio
 - 1.4.8 Visual Presentation

Accessibility in Svelte + Kit:

Where does Svelte
help us?



Accessibility warnings

- Warns about **potentially problematic markup**

```
<img src={src} />
```

A11y: element should have an alt attribute

- Many different checks: **a11y-missing-attribute**, a11y-autofocus, a11y-missing-content, a11y-interactive-supports-focus, a11y-label-has-associated-control ...
- **Checks have limits:** only see the current component, can't test if it's actually accessible

Accessibility warnings

- **Take care** when following warnings suggestions, it *might* not apply
- Don't add suggested attributes without **knowing implications!**
- Ignore errors if they *really* don't apply:

```
<!-- svelte-ignore a11y-error-name -->
```

- **Please only ignore when you are sure it's accessible**

Route announcements

- Screen Readers normally announce the page title on navigation
- *SvelteKit uses Client Side Routing, no full page reload on navigation*
- SvelteKit **injects a live region** with the page title to emulate behaviour

```
<div id="svelte-announcer" aria-live="assertive" aria-atomic="true" style="position: absolute; left: 0px; top: 0px; clip: rect(0px, 0px, 0px, 0px); clip-path: inset(50%); overflow: hidden; white-space: nowrap; width: 1px; height: 1px;">Dashboard - Datawrapper</div> == $0
```

- **Make sure you have unique html titles for each page of your site!**

Focus management

- Browsers normally reset focus on navigation
- *SvelteKit uses Client Side Routing, no full page reload on navigation*
- SvelteKit **focuses the body** to emulate behaviour on navigation
- Sometimes you may want to keep focus! (*more on that later...*)

The "lang" attribute

- `<html lang="en">` decides in how Screen Readers pronounce your website
- Multilang Website? `transformPageChunk!`

```
<!-- app.html -->  
<html lang="%lang%">
```

```
// hooks.server.ts  
export function handle({ event, resolve }) {  
  return resolve(event, {  
    transformPageChunk: ({ html }) => html.replace('%lang%', get_lang(event))  
  });  
}
```

Incrementally improving: Keyboard Navigation in our Vis Archive

The screenshot shows the Datawrapper web application interface. The browser address bar displays `https://app.datawrapper.de/archive/recently-edited`. The page title is "Datawrapper". In the top right corner, there are navigation links for "Dashboard", "Create new...", and "Archive".

On the left side, there is a sidebar with the following sections:

- SEARCH**
 - Recently edited by me
 - Recently published by me
- SHARED**

You don't have any shared folders yet. Start collaborating with others and create or join a team.

Team settings
- PRIVATE**
 - My archive (10)

The main content area is titled "Recently edited by me" and is sorted by "My last edit". It displays a grid of visualization cards:

- Münster**: Edited a few seconds ago. Placeholder image.
- Berlin**: Edited 3 minutes ago. Placeholder image.
- Income vs life expectancy**: Edited 9 minutes ago. Placeholder image.
- Gender pay gap**: Edited 9 minutes ago. Placeholder image.
- Rural and urban population**: Edited 9 minutes ago. Placeholder image.
- Tippling in Europe and America**: Edited 13 minutes ago. A horizontal bar chart showing percentages for various countries.
- House prices in the U.K.**: Edited 13 minutes ago. A line chart showing price trends for different regions.
- Berlin**: Edited 14 minutes ago. A map of Berlin with a color-coded overlay.
- The rise of Google Chrome**: Edited 16 minutes ago. A line chart showing market share trends.
- Global CO2 emissions by fuel and in...**: Edited 6 months ago. A stacked area chart showing cumulative emissions over time.

Problems:

1. Header nav has to be navigated through on every route change
2. Elements that appear on hover don't with keyboard focus
3. Extra focusable but non-interactable elements make navigating with keyboard extra annoying
4. Losing focus state when navigating inside the archive and even when changing the ordering of the list view

Problem #1: Header nav has to be navigated through on every route change

Fix: Add a skip to content link!

```
<a href="#main" class="skip-to-content-link">
  Skip to main content
</a>

<main id="main">
  <slot />
</main>
```

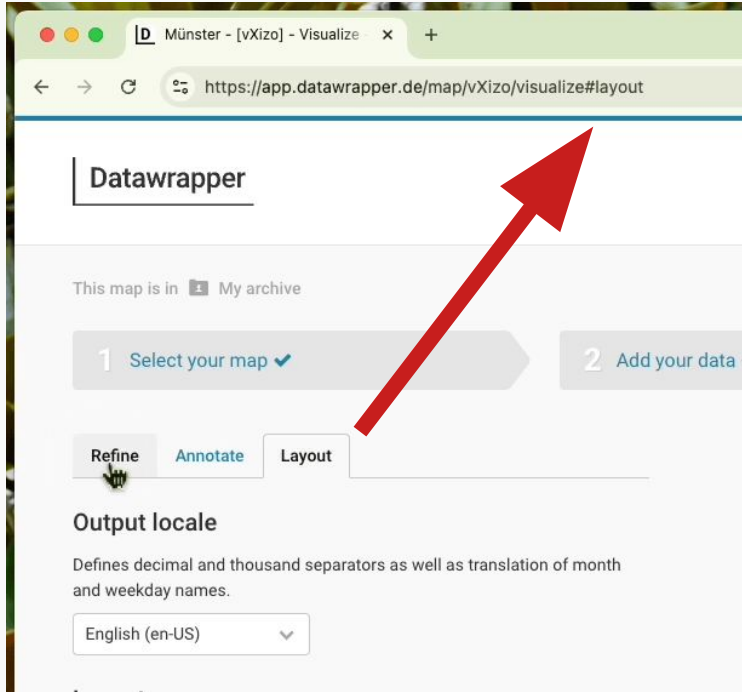
<https://www.sveltelab.dev/y80dhpke0c8syq6>

```
<style>
  .skip-to-content-link {
    position: absolute;
    z-index: 99999;
    padding: 1em;
    left: -90001px;
    opacity: 0;
    margin: 1em;
  }

  .skip-to-content-link:focus {
    left: 0;
    opacity: 1;
  }
</style>
```

Incrementally improving: Keyboard Navigation in our Vis Archive

Problem #1.5: This doesn't play nice with our editor



It's important to take a11y into account from the start!

Incrementally improving: Keyboard Navigation in our Vis Archive

Problem #1.5: This doesn't

Fix: Turn it into a skip to content

```
<button
  class="skip-to-main-content
  on:click={handleSkipToMainC
>
  Skip to main content
</button>
```

- `<a>` only activate on Enter, `<button>` also activate on Spacebar
- Button version can be used multiple times

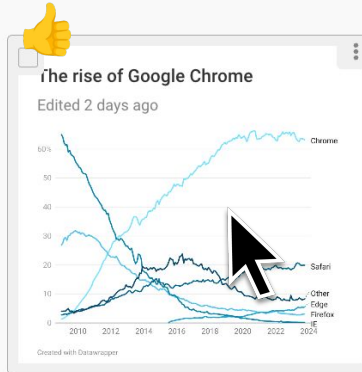
```
function handleSkipToMainContent() {
  const el = document.getElementById('main');
  if (!el) return;
  const tabindex = el.getAttribute('tabindex');
  el.tabIndex = -1;
  el.focus({ preventScroll: true, focusVisible: false });
  el.addEventListener(
    'blur',
    () => {
      if (tabindex !== null) {
        el.setAttribute('tabindex', tabindex);
      } else {
        el.removeAttribute('tabindex');
      }
    },
    { once: true }
  );
}
```

Incrementally improving: Keyboard Navigation in our Vis Archive

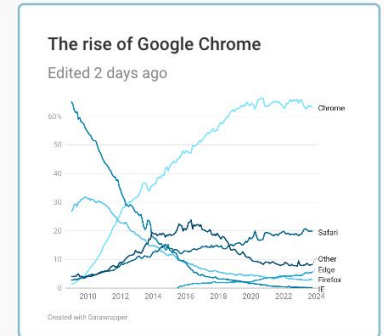
Problem #2: Elements that appear on hover don't with keyboard focus

```
.checkbox-container {  
  opacity: 0;  
}  
  
.vis-container.archive-box {  
  &:hover {  
    .checkbox-container {  
      opacity: 1;  
    }  
  }  
}
```

Using Mouse:



Using Keyboard: !?



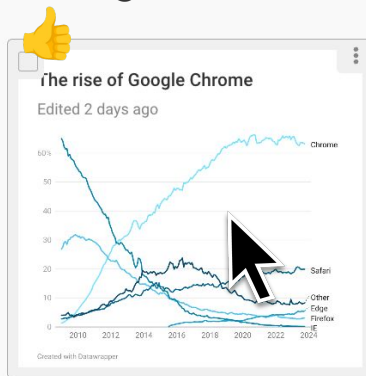
Incrementally improving: Keyboard Navigation in our Vis Archive

Problem #2: Elements that appear on hover don't with keyboard focus

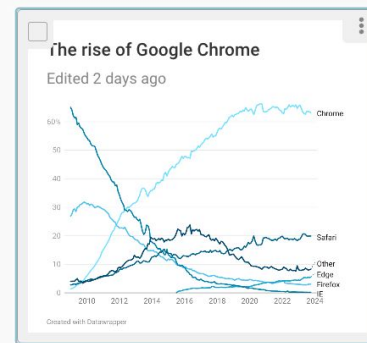
Fix: Consider adding `:focus-within` or `:focus` when using `:hover`

```
.checkbox-container {  
  opacity: 0;  
}  
  
.vis-container.archive-box {  
  &:hover,  
  &:focus-within {  
    .checkbox-container {  
      opacity: 1;  
    }  
  }  
}
```

Using Mouse: 👍



Using Keyboard: 👍



Problem #3: Extra focusable but non-interactable elements

```
<!-- ContentEditable.svelte -->  
<span  
  role="textbox"  
  tabindex="0"  
  contenteditable={editable ? 'true' : false}  
  on:click={onClick}  
  on:keydown={onKeyDown}  
>  
  {value}  
</span>
```

▼ PRIVATE

▼  My archive (2) ✓

 Cool new folder!



Problem #3: Extra focusable but non-interactable elements

A11y: Elements with the 'textbox' interactive role must have a tabindex value. svelte(a11y-interactive-supports-focus)

View Problem (⌘F8) Quick Fix... (⌘.)

```
<span
  role="textbox"

  contenteditable={editable ? 'true' : false}
  on:click={onClick}
  on:keydown={onKeyDown}
>
  {value}
</span>
```


Incrementally improving: Keyboard Navigation in our Vis Archive

Problem #3: Extra focusable but non-interactable elements

Using Keyboard: **!?**

▼ PRIVATE

▼  My archive (2) ✓

 Cooler folder name!

▶ SHARED

- Element: focusable
- Pressing enter: does nothing
- Pressing space: scrolls page



Incrementally improving: Keyboard Navigation in our Vis Archive


Problem #3: Extra focusable but non-interactable elements

Fix: Take care when following Svelte a11y warnings

Do not just add stuff to make them shut up, but test what you are doing

```
<!-- ContentEditable.svelte -->
<span
  role="textbox"
  tabindex={editable ? 0 : undefined}
  contenteditable={editable ? 'true' : false}
  on:click={onClick}
  on:keydown={onKeyDown}
>
  {value}
</span>
```

▼  My archive (10) ✓

 Cooler folder name!

Incrementally improving: Keyboard Navigation in our Vis Archive

Problem #4: Losing focus state when navigating inside the archive

(1) Focus

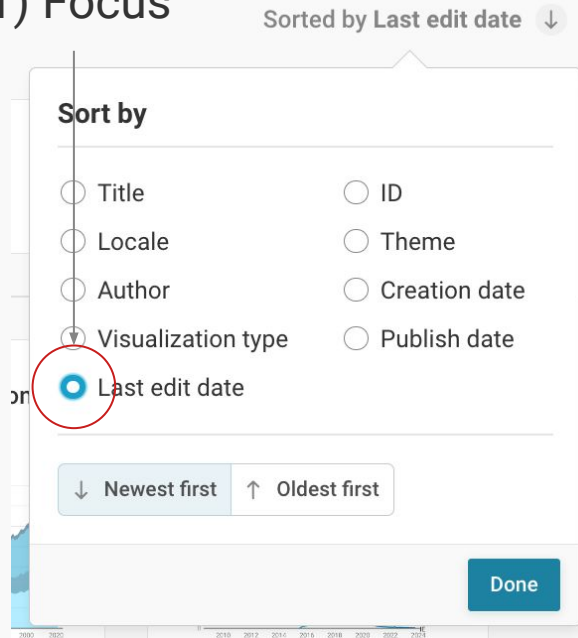
Sorted by Last edit date ↓

Sort by

<input type="radio"/> Title	<input type="radio"/> ID
<input type="radio"/> Locale	<input type="radio"/> Theme
<input type="radio"/> Author	<input type="radio"/> Creation date
<input type="radio"/> Visualization type	<input type="radio"/> Publish date
<input checked="" type="radio"/> Last edit date	

↓ Newest first ↑ Oldest first

Done



(2) Right Arrow Key to change Sort Order



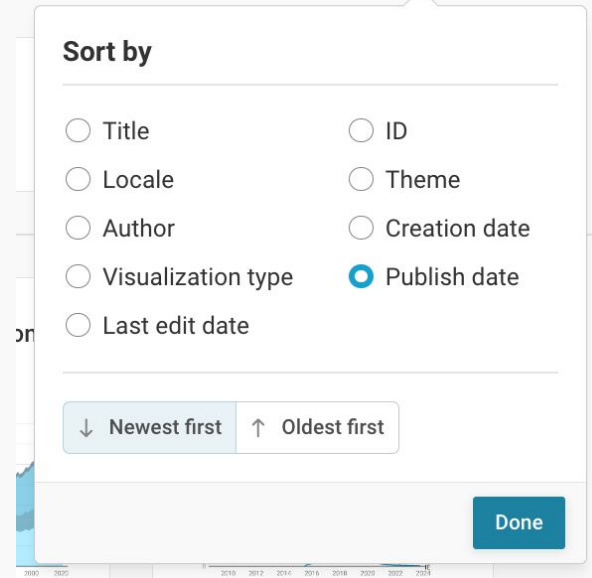
(3) Page Navigates & Focus is reset

Sort by

<input type="radio"/> Title	<input type="radio"/> ID
<input type="radio"/> Locale	<input type="radio"/> Theme
<input type="radio"/> Author	<input type="radio"/> Creation date
<input type="radio"/> Visualization type	<input checked="" type="radio"/> Publish date
<input type="radio"/> Last edit date	

↓ Newest first ↑ Oldest first

Done



Problem #4: Losing focus state when navigating inside the archive

```
$: if (currentOrderOption.id !== selectedId) {  
  currentOrderOption = sortOptions.find(o => o.id === selectedId);  
  goto(selectedOrderOption?.query, {  
    replaceState: true,  
    noScroll: true  
  });  
}
```

Incrementally improving: Keyboard Navigation in our Vis Archive

Problem #4: Losing focus state when navigating inside the archive

The screenshot illustrates a user interface for navigating through a code archive. On the left, a code editor displays a JavaScript snippet:

```
$: if (current) {
  goto(someRepoName);
}

});
}
```

The navigation menu on the right shows a 'PRIVATE' section with a dropdown for 'My archive (10)'. A folder named 'Coolest folder name ever!' is highlighted. A 'Sort by' dialog is open, showing options for sorting by Title, Locale, Author, Visualization type, ID, Theme, Creation date, Publish date, and Last edit date. The 'Last edit date' option is selected. Below the dialog, there are buttons for 'Newest first' and 'Oldest first', and a 'Done' button at the bottom right.

Incrementally improving: Keyboard Navigation in our Vis Archive

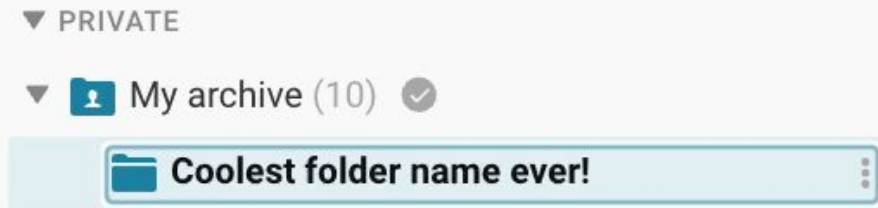
Problem #4: Losing focus state when navigating inside the archive

Fix: Use `keepFocus` option of `goto` when appropriate.

For links and forms: `data-sveltekit-keepfocus`

```
<a href={folder.path + orderParams} data-sveltekit-keepfocus>
  <FolderIcon {folder} />
  <ContentEditable bind:this={editableFolderName} value={folder.name} />
</a>
```

You should also **only** use this when the **focused element still exist after navigation!**



Let's try it again!

The screenshot shows the Datawrapper web application interface. The browser address bar displays `https://app.datawrapper.de/archive/recently-edited`. The page title is "Datawrapper". In the top right corner, there are navigation links for "Dashboard", "Create new...", and "Archive".

The main content area is titled "Recently edited by me" and is sorted by "My last edit". It displays a grid of visualization cards:

- Münster**: Edited a few seconds ago. Placeholder visualization.
- Berlin**: Edited 3 minutes ago. Placeholder visualization.
- Income vs life expectancy**: Edited 9 minutes ago. Placeholder visualization.
- Gender pay gap**: Edited 9 minutes ago. Placeholder visualization.
- Rural and urban population**: Edited 9 minutes ago. Placeholder visualization.
- Tippling in Europe and America**: Edited 9 minutes ago. A horizontal bar chart showing percentages for various countries.
- House prices in the U.K.**: Edited 13 minutes ago. A line chart showing price trends over time for different regions.
- Berlin**: Edited 14 minutes ago. A choropleth map of Berlin districts.
- The rise of Google Chrome**: Edited 16 minutes ago. A line chart showing market share trends from 2008 to 2014.
- Global CO2 emissions by fuel and in...**: Edited 6 months ago. A stacked area chart showing CO2 emissions from 1980 to 2010, categorized by fuel type.

The left sidebar contains navigation options: "SEARCH" (Recently edited by me, Recently published by me), "SHARED" (You don't have any shared folders yet...), and "PRIVATE" (My archive (10)).

Key Takeaways

- A11y levels the playing field and benefits everyone (and everything)
- Svelte and SvelteKit have useful features that help with a11y
- You don't need to memorize all WCAG rules by heart to make meaningful improvements.
- Don't just please the Svelte a11y warnings
 - test your changes via keyboard navigation and screen readers
 - research stuff when you are unsure



Thank You

Antonio Sarcevic

antonio@datawrapper.de
@sarcevic@mastodon.online

datawrapper.de

support@datawrapper.de
@datawrapper