# Github.com Accessibility Opportunities Part 1(unauthenticated user)

Browser: Chrome version 134.0.6998.178

O/S: Windows 11 version 10.0.26100

Screen Reader: NVDA 2023.1.0.27913

## Abbreviations and Conventions used in this document

SR(XYZ): Screen Reader (optional software name)

KB: Keyboard

For SR output:

- Title/lowercase: arbitrary text, e.g. the visible text in a link, the aria-label value supplied for a button, the text in an element referenced using aria-describedby
- ALL CAPS: system/programmatically generated text, e.g. role, state
- +: used to separate the parts of a contiguous announcement, as order may change in different contexts but what matters is that the various parts are communicated
- Examples:
    - “Learn more” + “LINK”
    - “Code” + “TAB” + “SELECTED”

## Issues

**Issue: Announcement - Inappropriate Use of alert role, Link missing context, Close button missing context**

*Where is this happening?*

- GitHub.com, all pages

*Applicable WCAG SCs*

- WCAG 2.0 – 1.3.1 Info and Relationships A
- WCAG 2.0 – 2.4.4 Link Purpose (In Context) A
- WCAG 2.0 – 4.1.2 Name, Role, Value A
- WCAG 2.1 - 4.1.3 Status Messages AA
- WCAG 2.2 – 2.5.8 Target Size (Minimum) AA

*Affected users*

SR users

*Actual behavior*

- For SR users, the alert is unnecessary
- “Learn more” is unhelpful context for SR users in isolation
- “Close” button label provides insufficient context for SR users in isolation
- “Close” button is unnecessarily small at 100% zoom

*Expected behavior*

- No alert at all
- When tabbing into the announcement region, SR users should hear that it’s a region and what it’s purpose is
- When focus moves to the “Learn more” link, SR users should know where it’s going to take them
- When focus moves to the “Close” button, SR users should know what it’s going to close
- The Close button could be larger

*Recommended solution*

- Change role=alert to role=”region”
- Keep aria-label=”announcement”
- Move the announcement container BEFORE the bypass block (“Skip to content”)
- Learn more
  - Use aria-label=”Learn more about GitHub CoPilot” for the link.
    (It’s important that the aria-label includes the visible text for voice command users.)
    OR
  - If the announcement text is consistently short, make the entire string the link, e.g. “GitHub CoPilot is now available for free. Learn more” (you don’t have to style the whole thing like a link, although you should consider showing the underline for all the text on focus or hover)
- Close:
  - SR context
    - if the region guidance above is applied, no change required; the context of the announcement region is fresh in SR user’s minds

        - otherwise, update the aria-label to “Close announcement” or something else precise and concise
    - Increase the size of the icon/button to 24x24
        - While it’s true that the current 16x16 is not an outright failure (since there are no other interactions nearby in the UI, lower-vision users will appreciate a larger more visible button

*Rationale/Explanation*

Alerts should be used sparingly.

Because this content is already at the top of the page/DOM, and contains interactive elements, SR users will find it as part of their normal interaction with the content.

This is an example where design parity (SR experience is 1:1 with visual presentation) is the better than design equity (a different, but in principle equivalent). Because for both sighted users and non-sighted SR users the announcement is “up front,” there’s no need to shout about it for SR users.

CTAs (calls to action) like links and buttons should speak for themselves; it should be clear to SR users what the CTA is going to do or where its going to take the user through its accessible name PLUS additional programmatic context (e.g. aria-describedby) as needed.

*Target behavior/outcomes*

When the page first loads, the tab order and SR announcements starting from the top would be:

1. Announcement link (if present)
   SR(NVDA): “Announcement” + “REGION” + “Learn more” + “LINK”
2. Close button
   SR(NVDA): “Close” + “BUTTON”
3. Bypass link
   SR(NVDA): “Skip to content” + “LINK”

**Issue: Global Navigation – Link behavior not communicated for SR, current page not communicated to SR**

*Where is this happening?*

- GitHub.com, all pages

*Applicable WCAG SCs*

- WCAG 2.0 – 4.1.2 Name, Role, Value A

*Affected users*

- SR users

*Actual behavior*

- Links that visually communicate they will open in a new tab do not communicate this to SR users
- Any link within navigation (global or footer) that points to the current page should communicate this to SR users

*Expected behavior*

- New tab links
  - The new tab behavior must be communicated to SR users, OR
  - Don’t force links to open in new tabs
- Current page
  - Use aria-current=page on appropriate links
  - Do not include the aria-current attribute on links otherwise

*Recommended solution (if opening in a new tab is a must-have)*

- Include .sr-only text at the end of the link text: “opens in new tab”, or
- Use the title attribute within the SVG to communicate the same (remove aria-hidden=true from the SVG)

*Rationale/Explanation*

- New tab links
  - Opening links in a new tab removes agency from all users; prefer letting users decide whether or not they want to open a link in another tab. Without a mouse, it can be challenging to get back to the original tab.
  - If you must use this pattern, ensure that SR users are informed of the behavior up front.
- Current page
  - SR users benefit from explicit confirmation that they are already at a particular destination, even if it’s not visually reinforced

*Target behavior (if opening in a new tab is a must-have)*

- On KB focus while using SR, links that have the icon indicating should announce this behavior
  SR(NVDA): “documentation, opens in new tab” + “LINK”

**Issue: Tab controls improperly implemented for SR users**

*Where is this happening?*

- GitHub.com, homepage, probably anywhere this tab component is used

*Applicable WCAG SCs*

- WCAG 2.0 – 4.1.2 Name, Role, Value A

*Affected users*

- SR users

*Actual behavior*

- All tabs announce for SR as selected whether they are selected or not
- For tab controls, associated content is sometimes above/before the tab control

*Expected behavior*

- Only an active tab control should announce as selected
- Tab content should appear below/after the tab control

*Recommended solution*

- Ensure aria-selected attribute is present on all <button role=tab> elements, and is dynamically set to true or false as appropriate
- [design impact] consistently display/render tab content below/after the tab control

*Rationale/Explanation*

- Tabs are expected to behave consistently for all users

*Target behavior*

- When focused using a KB+SR(NVDA), tabs should announce as:
    - "{tabName}" + "TAB" + ("SELECTED" or "NOT SELECTED")
    - When pressed/activated, the new state of the focused control ("SELECTED" or "NOT SELECTED") should be announced.