DEPARTMENT OF HEALTH

Designing Inclusive Shiny Dashboards: Accessibility Best Practices and Innovations

SHINYCONF 2025

About Our Work

We strive to develop dashboards that are informative, relevant to our audience, easy to navigate, and fully accessible. You can learn about and clone our demonstration products at https://github.com/ajstamm/shiny-a11y-app.

Important Terms

Web terms

- HTML: Hypertext Markup Language, the language of websites.
- div: Useful for grouping UI elements or applying custom styles and layouts. (`div()` creates a generic HTML `<div>` container)
- tags: A list of functions in the `shiny::tabs` object that generate specific HTML elements. (e.g. `tags$h1()`, `tags$p()`, `tags$img()`). Allows for full HTML control within the Shiny UI.

Shiny concepts

- input: User selections from text boxes, menus, and other input elements.
- output: Any element displayed on the page, including text, images, and interactive elements.
- server: A function that contains the logic of the Shiny app. It defines how inputs are used to generate outputs and handles reactivity.
- ui: The user interface layout of the app, defined using functions like `fluidPage()`, `sidebarLayout()`, etc. It specifies what inputs and outputs appear on the screen.
- sidebar: Optional column that appears beside the main panel and often includes input options or details about the app. Often created with `sidebarPanel()` inside `sidebarLayout()`.

Shiny functions

- uiOutput: A function used in the UI to create a placeholder for dynamic UI elements rendered from the server using `renderUI()`.
- shinyApp: Function that runs the app. It takes *ui* and *server* as arguments and starts the application.

Resources Available Online

WCAG Resources

These sites provide information about web content accessibility guidelines (WCAG).

- https://www.w3.org/TR/WCAG21/
- https://www.w3.org/WAI/standards-guidelines/wcag/

R Packages that Address Accessibility

We will use or touch on these packages in our workshop.

- shinya11y (https://github.com/ewenme/shinya11y): creates an overlay to review how accessible your shiny app is
- BrailleR (https://ajrgodfrey.github.io/BrailleR/): provides text descriptions of plots for blind users
- sonify (https://CRAN.R-project.org/package=sonify): creates sounds in the shape of charts
- tactileR (https://github.com/jooyoungseo/tactileR): creates tactile charts
- ggtextures (https://github.com/clauswilke/ggtextures): adds textures to ggplots
- RColorBrewer (https://CRAN.R-project.org/package=RColorBrewer) and viridis (https://github.com/sjmgarnier/viridis/): provides color palettes that address colorblindness
- colorspace (https://CRAN.R-project.org/package=colorspace): maps between color spaces
- htmltools (https://github.com/rstudio/htmltools/): manipulates HTML elements
- shiny (https://shiny.posit.co/): framework for building interactive dashboards, R package to create that framework
- bslib (https://github.com/rstudio/bslib/): custom bootstrap themes for shiny

Tools to Test Accessibility

We will use these tools in our workshop.

- ANDI (https://www.ssa.gov/accessibility/andi/help/install.html)
- WAVE (https://wave.webaim.org/)

The World Wide Web Consortium provides a comprehensive list of accessibility tools of various types.

- Web Accessibility Evaluation Tools List (https://www.w3.org/WAI/test-evaluate/tools/list/)

Minnesota Department of Health
Office of Data Strategy & Interoperability
625 Robert St. N
PO Box 64975
St. Paul, MN 55164-0975
health.dsi-datata@state.mn.us
https://www.health.state.mn.us/dsi

04/04/2025

To obtain this information in a different format, email health.dsi-datata@state.mn.us.