County of Fairfax, Virginia

WEB ACCESSIBILITY STANDARDS

Department of Information Technology
Date: January 2022

Fairfax County is committed to nondiscrimination in all county programs, services, and activities. Website content will be made accessible as much as possible to users with physical, language, learning, or cognitive disabilities. The following Web Accessibility Standards apply to all content published on the county's Internet and Intranet websites, mobile applications, and any official internal electronic communication provided by an agency. All content hosted externally on behalf of the county must also follow these standards.

Web content must meet federally mandated accessibility requirements established by Section 508 of the Rehabilitation Act and conform to W3C's Web Content Accessibility Guidelines 2.1 (WCAG 2.1). WCAG 2.1 is the most recent and relevant globally recognized accessibility guidelines. It incorporates all the success criteria of WCAG 2.0 required by Section 508 'Refresh' and includes additional criteria to address mobile accessibility, low vision, and cognitive limitations. There are three levels of accessibility compliance: A (minimum), AA (legal standard), and AAA (advanced). To the greatest extent feasible, **Fairfax County conforms** to WCAG 2.1 Level AA guidelines**.**

Department of Information Technology

Web Accessibility Standards

To meet federally mandated accessibility requirements and conform to WCAG 2.1 Level AA guidelines, Web content should follow the criteria as listed below:

SECTIONS

WEB

1) An alternate version of a non-compliant web page, with equivalent information, language, and functionality as the non-compliant page, shall be provided to make a web site compliant when compliance cannot be accomplished in any other way. The alternate version must be as up-to-date as the non-compliant page and must be reached from a compliant web page
2) Identify components that have the same functionality consistently throughout a site
3) Present navigation components consistently throughout a site
4) Provide multiple ways of locating content
5) Avoid disabling the browser’s back button
6) Ensure that your website does not restrict orientation to only portrait or landscape, unless a specific orientation is necessary

CODE & CONTENT

7) Use valid HTML code to avoid parsing errors with browsers and assistive technologies
    a) To ensure you have valid code, run your code through a code validator like W3C Code Validator
8) Use the HTML language attribute "lang" to identify the default language of the text
9) Identify any changes in the language of the text
10) Use headings to begin sections and use them in the correct order
11) Use lists to identify series of related items
12) Provide descriptive page titles
13) If the content has a time limit, provide users enough time to read and use content by allowing time limits to be adjustable by the user
14) The reading order of the content must be logical, intuitive, and available to assistive technology
15) All page functionality is available from the keyboard
    a) Users can navigate through all pages using only the keyboard
    b) Keyboard is never locked on a particular element on the web page
    c) Ensure the keyboard focus indicator (i.e., cursor) is visible
    d) If you have a one key keyboard shortcut, make sure a user can either
    e) 1) turn it off, 2) there's a way to add another key in the shortcut, and/or 3) have the shortcut only active while focusing on a specific component
16) Do not change focus or content unexpectedly
17) Do not refer to elements on the page by their color, size, shape or location, or sound.
    a) For example, do not use "click the circle to continue" or "use the menu on lefthand side, or "wait for the beep to continue"
18) Avoid pop-up windows unless necessary. For non-modal pop-up content (i.e., tooltips) that displays on hover or focus, ensure it can also be hovered, easily dismissed without moving hover or focus, and stays visible until losing focus or dismissal by user
19) If an important status message is presented and focus is not set to that message, the message must be announced to screen reader users, typically via an ARIA alert or live region

FILES

20) For digital presentations (i.e., PowerPoint), every slide has a unique title, reading order is properly set, and all images and charts have alternative text

21) PDFs must be accessible and adhere to PDF standards

    a) If an automated accessibility checker has been provided by the program used to create the PDF, the file should pass the accessibility check without any substantial errors

22) Spreadsheets include labels for rows and columns, detailed labels for any charts and are accompanied by a textual description of the spreadsheet

FORMS

23) Provide labels for all form input fields and associate each label to their input field

24) Clearly indicate required fields by using either a symbol, text, or color but not color alone

25) Provide legends for groups of related form fields (i.e., radio buttons or checkboxes)

26) Ensure the tab order between form elements is logical

27) For web pages that cause legal commitments or financial transactions for the user, allow the user to check and confirm the action before it is completed or provide a way to reverse the action

28) Clearly identify user input errors in plain text and provide suggestions to fix the issue

29) Provide instructions that will help the user avoid input errors to include required fields, restrictions, and data format

30) Ensure input fields that collect certain types of user information have an appropriate autocomplete attribute defined

31) Ensure visible labels start with the same text as their corresponding accessible names (i.e., title or aria-label should begin with the text of button)

32) Use accessible version of CAPTCHAs when necessary

    a) Do not use legacy version requiring users to identify letters or pictures containing certain imagery

FRAMES

33) Ensure frames are appropriately titled
34) Provide equivalent content in an HTML <noframes> tag for browsers that do not support frames

IMAGES

35) Provide appropriate alternative text (“alt”) for all images
36) Provide full descriptions for complex images such as graphs and diagrams
37) Use client-side image maps instead of server-side image maps except where the regions cannot be defined with an available geometric shape.
38) If server-side image maps must-be used for legitimate technical reasons, each link in the image map must be repeated elsewhere in the page in a non-graphical format (i.e., list of links)
39) Provide alternate text for each area in client-side image maps
40) Use actual text instead of images of text
    a) Images of text are allowed for logos only and must include the same words used in the image for the alternate text

LINKS

41) Provide skip navigation to bypass repetitive content or navigation links
42) Ensure that links are understandable and clearly indicate the purpose by the text of hyperlink
43) To distinguish hyperlinks from surrounding text, the contrast ratio must be at least 3:1 and a different color used to hover over the link

MULTIMEDIA

44) Do not design content in a way that is known to cause seizures
    a) Flashing objects should be limited to less than 3 flashes per second
45) Do not automatically play audio, slideshows, or animation

46) Provide a way to stop, start, pause, or control the volume for audio that plays on a page for more than 3 seconds

47) Provide a way for users to pause, stop, and hide any moving, scrolling, or auto updating that lasts longer than 5 seconds (i.e., AJAX updates, news tickers, animated graphics)

48) Provide alternatives for any time-based audio-only, video-only, or multimedia content

a) For pre-recorded web-based content:
- Provide a transcript or audio description for video-only files
- Provide a transcript for audio-only files (i.e., MP3 files, podcasts)
- Provide synchronized captions for audio content in synchronized media

b) For live web-based content:
- Provide synchronized captions for all live multimedia that contain audio content (i.e., video conferences, webcasts, and animations).

SCRIPTS AND APPLETS

49) Ensure that scripted functions are usable with assistive technologies

50) Each script should be accompanied by a <noscript> element that contains a

text equivalent for the script

51) Provide links to Section 508 compliant plug-ins that are required to interpret page content

STYLES

52) Ensure that the page is readable and functional when the text size is doubled

53) Try using *em* or *rem* font sizing when available instead of *px* to allow the text to stay proportionally sized when under significant zoom

54) Ensure appropriate contrast between the text and background

a) Text must have a contrast ratio of at least 4.5:1

b) Large text (i.e.,18pt) must have a contrast ratio of at least 3:1

55) Ensure an appropriate contrast between non-text content (i.e., active controls, icons, buttons, images) and background. A minimum contrast ratio of 3:1 is required.

56) Allow end users to override text spacing with no loss of content or functionality

57) Do not convey information by color alone

a) Whenever color is used as an indicator, also use a non-color-based indicator

b) To distinguish hyperlinks from surrounding text, the contrast ratio must be at least 3:1 and a different color used to hover over the link

TABLES

58) Associate data cells with their headers by identifying the row and column headers
    a) For example, in HTML use TD to define a table cell and TH to define a table header.
59) Use the summary or caption attribute to convey the information presented in the data table.
60) Do not use data table markup for layout tables. If you use a table for layout, use the role=”presentation” on the <table> tags

61) For data tables that have two or more logical levels of row or column headers, use markup to associate data cells and header cells. For example, use THEAD, TFOOT, and TBODY to group rows, COL and COLGROUP to group columns

MOBILE

In addition to adhering to WCAG 2.1 AA guidelines listed under the Web section (criteria 1-61) above, the following mobile accessibility guidelines are also applicable*:*

62) Minimize scrolling
    a) Content should scroll either vertically or horizontally but not both unless it includes maps, tables, or diagrams

63) Users should be able to seamlessly view the content of your mobile app by rotating their devices between portrait and landscape

64) Provide consistent layout using responsive web design based on device size and screen orientation

65) Embeds are preferable to Iframes, as they tend to perform better on smaller screens

66) Do not restrict the ability to zoom
    a) If you define viewport attributes, do not set user-scalable to '0' or 'no'.
67) Ensure touch targets are appropriately sized.
    a) Use a minimum of 24 x 24 CSS pixels for touch targets on a physical device
    b) Ensure adequate spacing between the targets to minimize risk of mis-selecting

68) Provide simple alternatives (e.g., single tap vs. swipe) to potentially complex finger

motions on touch screens

69) Single-pointer gestures have a higher risk of unintentionally performing an action, so make sure at least one of the following is true:

   a) The down event does not execute the function. A user can swipe away from a button depress, and it does not perform the button action (the behavior completes on the up event)

   b) Provide a mechanism that cancels out any accidental activation of elements during the down event like when an up event occurs, or a confirmation before such action takes place

70) For any functions that are activated by motion (i.e., shaking, tilting, gesturing towards the camera), provide a simpler, alternative means of action. Also, give users the option to turn off motion activation.

71) Ensure "hamburger menus" and other mobile-specific UI elements work with keyboard alone, and do not require touch gestures to accommodate users that use Bluetooth or USB keyboards on their mobile devices

72) Provide easy methods for data entry

   a) Reduce the amount of text entry required by using select menus, radio buttons, checkboxes, or by auto-filling information like date, time, and location

SOCIAL MEDIA

73) Add alt text to all uploaded images

74) Do not use animated GIFs

75) Do not automatically play audio, slideshows, or animation

76) Do not use ASCII art as screen readers announce each character

77) Type hashtags in CamelCase for readability (i.e., #thisishardtoread vs #ThisIsHardToRead)

78) Emojis are accessible and automatically described by screen readers, though adequate spacing is required between emojis when adding a string of emojis

More detailed help and resources can be found on the **Web Accessibility page** on FairfaxNET