Cornell University

Web accessibility testing guidelines for WCAG 2.2 AA

The following guideline is a comprehensive guide on how to test for each Success Criterion in the Web Content Accessibility Guidelines 2.2, Level AA (WCAG 2.2 AA). It is intended to be used in conjunction with the WCAG 2 Checklist, as the tests are derived from the Checklist descriptions.

Using the guidelines

Within the guide, bulleted list items are all checks that the reviewer is expected to run through. If you come across any situation where a check item is not met, mark a failure in the checklist and fill in details of where the issue was discovered.

You may tag an item as “Warning” on the WCAG 2.2 checklist if you determine that the element you have evaluated technically passes WCAG but offers a degraded user experience that should not exist and is fixable by the author (i.e., don’t tag something that a screen reader announces in an unexpected manner if the screen reader itself is the issue).

Authoring information

Last Update: May 23, 2025

Latest Checklist Version: 3.25.098

Authored by: Sean Gnau (sg667), Hank De Leo (hcd43)

About the checklist

The WCAG 2.2 AA Checklist contains 90 guidelines that have to be met to fully conform to WCAG 2.2 AA. It is important to note that the guidelines in the checklist are NOT the exact guidelines defined in the official W3C resource. They are *interpretations* of WCAG 2.2 which are designed to break down some of the more broadly scoped criteria.

Categories

The checklist also provides categories for each testing guideline. These are meant to help identify relevant criteria as you are evaluating a site, application, or document. They are broken down as follows.

- **Color**: Tests about color contrast and the use of colors on the page.
- **Content**: Tests about how content is written.
- **Forms and Inputs**: Tests that verify the proper use of forms and inputs (buttons, selects, etc.)
- **Images**: Tests where images are verified for appropriate use and accessibility features.
- **Interaction**: Tests about how site functions behave when you interact with them.
- **Keyboard**: Tests that require manual keyboard testing to verify (also covers Screen Readers)
- **Motion**: Tests that must be checked when content moves or changes on its own.
- **Multimedia**: Tests about video and audio.
- **Resizing**: Tests about how the page behaves when resizing, zooming, or rotating the browser contents.
- **Structure**: Tests about the theme and the appropriate use of HTML markup.

Filling out an issue in the Checklist

To fill out an issue, find the guideline that the site is in violation of and fill in the columns E through N for that row.

The checklist is intended to be filled out as "one issue per line." If you have multiple failures for one criterion, add a new row, and copy at least the contents from Columns A through D into the new row. Add your second issue in the newly copied row.

Mark a "Pass" if the criterion is fully met and there are zero failures.
Mark a “Warn” if the criterion is met, but the user experience fails to meet accessibility best practices or the user experience is otherwise compromised.

Mark a "Fail" if the criterion is not met in any way or the user experience is diminished such that a user of assistive technologies would be significantly inconvenienced.

Mark "N/A" if the criterion is not applicable to the website/webapp (e.g., you would mark N/A for any criterion about multimedia if the website/webapp doesn't have any multimedia content).

Cornell Custom Dev WCAG 2.2 Checklist Scripts

The ‘Cornell Custom Dev WCAG 2.2 Checklist Scripts’ is a Google Apps Script that compiles the data in the primary Checklist Tab and outputs a deliverable format. When asked, please grant permission. The Summary Doc script will generate a new Google Doc and automatically save it in your Google Docs Workspace.

Definitions

Image Categorization

Images used on the web have different purposes, and the accessibility requirements of each image vary depending on its purpose. Common accessibility guidance groups images into different categories.

- **Informative:** Images that describe basic information or concepts that can be described in a short phrase or smaller.
- **Decorative:** Images that do not add any new information to the page.
- **Functional:** Linked images or images that can be clicked like a button to start some action.
- **Complex:** Similar to informative images, but explaining the contents would take a significant amount of text. Graphs, charts, and maps are common examples.
- **Grouped:** Multiple images are used to convey one bit of information.

Testing tools

There are multiple different testing tools and mechanisms that a reviewer must become familiar with to perform all of the required accessibility checks on the website or application they are testing. The following list will detail all of the tools this guide will ask for and how to access each.

Automated testing

- **Siteimprove**: One of our primary testing tools. Siteimprove allows us to scan entire websites and perform automated accessibility checks. The enterprise application version only works on publicly accessible websites and web applications. It cannot evaluate anything behind authentication (login prompts).
  You may access Siteimprove via https://it.cornell.edu/siteimprove. Click on "First Time Account Setup" if you have not used the application yet.
- **Siteimprove Plugin**: This is a browser plugin that you can download from your browser's storefront. This plugin is able to run on any page and perform the same accessibility checks that the primary enterprise application is able to. This is generally used only when the page you are evaluating is behind a login, or the thing you are reviewing is an external site/app.
- **WAVE**: This is a browser plugin that activates a sidebar and overlay that lists out accessibility issues identified on the page you are currently looking at.
- **ANDI**: A "bookmarklet" app that the Social Security Administration uses to test accessibility on sites. It activates an overlay and can notably provide a simulated screen reader output for any element that you hover or keyboard navigate to without having to invoke your actual screen reader.

Manual Testing

- **Keyboard**: Keyboard testing is a manual process where you use the TAB, SHIFT+TAB, and arrow keys (primarily) to navigate through the page by moving "focus." Elements with keyboard focus are generally identified by an outline.

- **Screen Reader**: This is a tool that reads out text-based and text-alternative content aloud. Windows and Mac come with a screen reader pre-installed (Narrator and VoiceOver respectively). There are also third-party screen readers for Windows such as JAWS and NVDA.

  If you are on macOS, use VoiceOver when performing screen reader tests. If you are on Windows, use NVDA or JAWS as they have much greater usage than the built-in Narrator.

All guidelines

1

Informative and grouped images must contain alternative text describing the purpose or meaning of the image(s).

WCAG 2 criterion

1.1.1 Non-text Content (Level A)

Tools and requirements

- WAVE extension, screen reader, or browser inspector

Note

This guideline is targeted at “informative images and grouped images.” Please refer to the image categorization guidance in the Definitions section if you are unsure which category the image you are assessing belongs to.

Test Procedure

Determine basic alternative text support.

Inspect any images, and make sure that there is basic alternative text support.

- If the `<img>` tag or `role=”img”` is used
    - There must be an `alt` attribute on the image. (Siteimprove and WAVE detect this.)
    - Every `<img>` tag in a group of images must have the `alt` attribute.
- If the `<svg>` tag is used

    - The attribute and value `role=”img”` must be applied.
- If the `<figure>` tag is used
    - The image(s) contained within use the conventions of the aforementioned `<img>` and `<svg>` tags.
- If there is a `title` attribute present on an image, that does **not** factor into alternative text. This attribute only provides hover-over text and is not available to assistive technology.
- CSS background images must not be used to present informative or grouped images.

Determine if the alternative text value is appropriate.

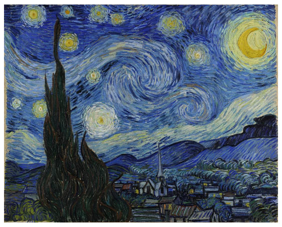
Use WAVE, a Screen Reader, or the browser’s inspector tools (F12) to view the alternative text for each image.

Note If the “informative image” is described adequately by plain text information directly adjacent to the image, it should follow Guideline 2 instead as it should be considered a decorative image.

- Alternative text must adequately describe the visual contents of the image.
- Alternative text may NOT be a filename, or placeholder text like “picture-1.”
- **Best Practice** Alternative text for images should not include text like “image of.” Assistive technology users will already know this. It is okay to state that an image is perhaps a screenshot, a photograph, or a painting, etc., if it is necessary to know this.
- **Best Practice** Alternative text should not be text that would be considered the title, caption, or legend of the image. We are only concerned with the literal visual appearance.

Examples of alternative text for an informative image

Consider an image of Vincent Van Gogh’s *The Starry Night*



Good: *An oil painting with thick individual brush strokes of a dark blue sky with yellow stars, white swirling clouds, and a yellow crescent moon above a town. A large black wavy cyprus tree stands in front of the left third of the painting.*
(You would then have the title and author as a caption)

Passable: *The Starry Night by Vincent Van Gogh*

Bad: *van-gogh-starry-night.jpg*

2

Decorative images must have empty alternative text or be otherwise hidden from assistive technology.

WCAG 2 criterion

1.1.1 Non-text Content (Level A)

Tools and requirements

- WAVE Extension + Manual Evaluation
- ANDI
- Siteimprove

Test procedure

This guideline is targeted at "decorative images." Please refer to the image categorization guidance in the Definitions section if you are unsure which category the image you are assessing belongs to.

Decorative images include images that are fully described in text positioned directly adjacent to the image.

Inspect any decorative images and determine the following.

- If the `<img>` tag or `role="img"` is used
    - The attribute and value `alt=""` must be applied to the decorative image.
- If the `<svg>` tag is used
    - The attribute and value `aria-hidden="true"` must be applied to the decorative image.

- If CSS background images are used, there is no further requirement for decorative images.

3

Functional images must have alternative text describing the input's purpose.

WCAG 2 criterion

1.1.1 Non-text Content (Level A)

Tools and requirements

- WAVE extension
- Manual evaluation required

Note

This guideline is targeted at "functional images." Please refer to the image categorization guidance in the Definitions section if you are unsure which category the image you are assessing belongs to.

Test procedure

Confirm the functionality of the image when you interact with it. (Figure out if it is a link or if it is a button that triggers some action.) After doing so, check for the following properties on the functional image.

- Every image must have an `alt` attribute present.

- Alternative text for functional images must describe the functionality of the image.
    - A linked image must have alt text equivalent to the destination. Additional text like “logo of” or “image of” may not be present.
    - An image that triggers some action needs alt text equivalent to what the label of a normal button that would trigger said action would be.

4

Complex images (graphs, maps, charts) must have a text description of all relevant information.

WCAG 2 criterion

1.1.1 Non-text Content (Level A)

Tools and requirements

- WAVE Extension + Manual Evaluation

Note

This guideline is targeted at "complex images" Please refer to the image categorization guidance in the Definitions section if you are unsure which category the image you are assessing belongs to.

Examples of images that are relevant for this guideline include

- Graphs and charts (line, pie, bar, etc.)
- Maps
- Diagrams
- Academic scientific posters

Test procedure

All complex images require a long description. Ensure the long description meets the following requirements.

- Long descriptions must be available and describe all of the important information contained in the complex image.

    - For a **chart or graph**, a long description would have details of all of the data points in the graph. A common way of doing so would be to link to an accessible data table with all of the data points.
    - For a **map**, you want a long description to describe the details the map is trying to provide. For instance, if it is there for **directions**, a text alternative describing how to navigate to the location should be provided. If the map is designed to **highlight a region**, then a text alternative should describe how far the region extends with as much detail as necessary.
        - *Instructions based on sensory characteristics based on the real world are okay to provide.*
    - For a **diagram or other complex image** (such as a flow chart or scientific poster), the long description must include all the information in the diagram in text format. A flow chart can usually be converted to an unordered list. Other diagrams will need lengthy text descriptions, and these text descriptions should be formatted with headings and semantic markup.
- The location of the long description must be easy to find. One of the following techniques must be present.
    - The location of the long description is included in the `alt` text of the complex image.
    - The long description is in text or in a link directly adjacent to the image.
    - The long description is in the same `role=”group”` as the complex image.
    - **Note** While `longdesc` is meant for this type of application, it does not have good assistive technology support, and must be supplemented with one of the other solutions.

5

CAPTCHAs must be identified with alternative text

WCAG 2 criterion

1.1.1 Non-text Content (Level A)

Tools and requirements

- Screen Reader

Test procedure

Use a screen reader and navigate to the CAPTCHA.

- The CAPTCHA must announce its presence to a screen reader
- The CAPTCHA's accessible name must provide information that a task is required and the type of task.
    - Ex 1 "Type the word in the image"
    - Ex 2 "Type the letters spoken in the audio"

6

CAPTCHAs which require user input offer at least two different modalities (e.g. visual and auditory)

WCAG 2 criterion

1.1.1 Non-text Content (Level A)

Tools and requirements

- Screen Reader
- Keyboard

Test procedure

Use various input mechanisms (cursor, keyboard, screen reader) to navigate to a CAPTCHA and interact with it. Identify what senses (sight, vision) the CAPTCHA requires by default and observe if there is an option for an alternative.

- CAPTCHAs must have two different modalities of interacting with the test.
  - Commonly, you'll have a vision test paired with an auditory test.
- All CAPTCHA modalities must allow the user to achieve the same thing.

7

Adjacent text and images which navigate to the same destination should be presented and announced as one link.

WCAG 2 criterion

1.1.1 Non-text Content (Level A)

Tools and requirements

- Siteimprove (Policy)

Test procedure

The most common page types that run across issues for this test are news pages, people pages, and events pages, as these often have listings with featured images next to a text link.

We have a policy in Siteimprove (enterprise app) to assist with checking this. It is titled "Adjacent Links." Note, the Siteimprove check only alerts that two links are next to each other, it does not verify if the links navigate to the same destination.

Linked images supplemented with a text link where both the image and text link navigate to the same destination must be presented to the user as a singular link. One of the following techniques must apply.

Option 1: The alt text of the image after the links are combined must be empty.

```
<a href="##">
     <img alt="" src="imgsrc.png" />
     <p>My Link Destination</p>
</a>
```

Option 2: The image is given `aria-hidden=”true”` and `tabindex=”-1”` so only the text link is utilized. If this is done, the image must have a text alternative that is consistent with the requirements of a functional image.

```
<a href="##" aria-hidden="true" tabindex="-1">
     <img alt="My Link Destination" src="imgsrc.png">
</a>
<a href="##">
     <p>My link destination</p>
</a>
```

8

Audio-only content must supply a basic text transcript. The location must be referenced in the accessible name of the audio content.

WCAG 2 criterion

1.2.1 Audio-only and Video-only (Prerecorded) (Level A)

Tools and requirements

- Siteimprove helps (QA > Inventory)

Test procedure

Audio-only content can be found with the assistance of the enterprise Siteimprove app. From the Dashboard of the site you are assessing, navigate to QA in the navigation, then the Inventory option. Click on the “Audio” link to get a list of audio content on the site. (If Audio is not clickable, there are no detected audio-only items within the site).

Audio-only content must meet the following checks.

- Ensure that a **basic text transcript** is available adjacent to the audio content, or linked adjacent to the audio content.
    - Any important audio (spoken words, important background/ambient sounds).
    - Speakers must be identified.
    - Timestamps are **NOT** required.

9

Video-only content supplies either a descriptive text transcript or audio description. The location must be referenced in the accessible name of the video content.

WCAG 2 criterion

1.2.1 Audio-only and Video-only (Prerecorded) (Level A)

Tools and requirements

- Siteimprove helps for identifying video content (QA > Inventory)

Test procedure

Video content can be found with the assistance of the enterprise Siteimprove app. From the Dashboard of the site you are assessing, navigate to QA in the navigation, then the Inventory option. Click on the “Video” link to get a list of video content on the site. Please note that this includes silent video (the subject of this criterion) AND multimedia content.

For any content which ONLY includes video and presents vital (non-decorative) information to understanding the content.

- One of the following must be included adjacent to the content or linked to adjacent to the content.
    - Audio Description which includes all the following.
        - Any text that appears on screen
        - The visual appearance of the scene

        - Any visual content which is important for the user to understand what is going on (e.g. “Tom stands up and reaches for his keys”)
        - Separate audio descriptions are not required if everything is already described in the primary audio track (such as a video where commentary describes the action on screen)
    - *Descriptive* text transcript which includes all of the following.
        - Any important audio (spoken words, important background/ambient sounds)
        - Speakers must be identified.
        - Important visual details, such as text on screen, or anything important for understanding the content which is only shown visually.
        - Timestamps are **NOT** required.

10

Multimedia content must have caption support for audio. Captions must be accurate, must include dialogue, the individual speaking, and any relevant audio information.

WCAG 2 criterion

1.2.2 Captions (Prerecorded) (Level A)

Tools and requirements

- Siteimprove helps for discovering video content (QA > Inventory)

Test procedure

Video content can be found with the assistance of the enterprise Siteimprove app. From the Dashboard of the site you are assessing, navigate to QA in the navigation, then the Inventory option. Click on the “Video” link to get a list of video content on the site. Please note that this includes multimedia video (the subject of this criterion) and silent video.

- Captions must be present.
    - Closed or Open captions are both valid.
- The captions MUST be accurate to the audio.
    - Commonly, YouTube automated captioning is inaccurate, and rarely uses punctuation. These will more often than not fail to be sufficient.
- The captions MUST include all relevant non-verbal noises.
- The captions MUST denote a change of speaker.
    - It’s strongly recommended to always denote speaker changes in captions, to support users with cognitive disabilities, but if the speaker

change happens on-screen (as opposed to background audio/narrator voices), it’s a warning not a failure.

11

Multimedia content must supply a text transcript OR an audio description.

WCAG 2 criterion

1.2.3 Audio Description or Media Alternative (Prerecorded) (Level A)

Tools and requirements

- Siteimprove helps for discovering video content (QA > Inventory)

Test procedure

Video content can be found with the assistance of the enterprise Siteimprove app. From the Dashboard of the site you are assessing, navigate to QA in the navigation, then the Inventory option. Click on the "Video" link to get a list of video content on the site. Please note that this includes multimedia video (the subject of this criterion) and silent video.

- One of the following must be present adjacent to the multimedia content.
  - Descriptive text transcript
    - Any important audio (spoken words, important background/ambient sounds).
    - Speakers must be identified.
    - Important visual details, such as text on screen, or anything important for understanding the content which is only shown visually.
    - Timestamps are **NOT** required.
  - Audio description which includes all of the following.
    - Any text that appears on screen.

- The visual appearance of the scene.
- Any visual content that is important for the user to understand what is going on (e.g. “Tom stands up and reaches for his keys”).
- Separate audio descriptions are not required if everything is already described in the primary audio track (such as a video where commentary describes the action on screen).

12

Live multimedia content must be captioned. Captions must be accurate, must include dialogue, the individual speaking, and any relevant audio information.

WCAG 2 criterion

1.2.4 Captions (Live) (Level AA)

Tools and requirements

- Siteimprove helps for discovering video content (QA > Inventory)

Test procedure

Any multimedia content presented live must have caption support.

- Captions are not required for silent live video.
- The captions MUST be accurate to the audio.
- The captions MUST denote a change of speaker.

13

Multimedia content must supply an audio description, which accurately informs the user of any important visual information not already conveyed through audio.

WCAG 2 criterion

1.2.5 Audio Description (Prerecorded) (Level AA)

Tools and requirements

- Siteimprove helps for discovering video content (QA > Inventory)

Test procedure

All multimedia content must include the following checks.

- Audio description must be provided where the original audio track does not describe all relevant information presented visually.
- The audio description must be available as a separate track, or a separate video altogether with the audio description content included.
- **Audio descriptions must include the following information not already described by the default audio track**
    - Any text that appears on screen.
    - The visual appearance of the scene.
    - Any visual content that is important for the user to understand what is going on (e.g., "Tom stands up and reaches for his keys").

Note: This overlaps with Guideline 11, for WCAG 2 SC 1.2.3 Audio Description or Media Alternative (Prerecorded). The reason is that criterion 1.2.3 is required to meet the less *stringent Level A compliance. For Level AA compliance, the audio description is not optional anymore and must be included.*

14

Landmarks present on the page are used to organize the correct type of content.

WCAG 2 criterion

1.3.1 Info and Relationships (Level A)

Tools and requirements

- WAVE
- ANDI
- Screen Reader

Test procedure

Use WAVE or a Screen Reader to identify all of the landmarks on the page. Use the inspector of your browser or a screen reader to identify the labels of any landmarks (the aria-label or aria-labelledby value, if it exists). All landmarks that do exist must meet the following checks.

- The landmark type (banner, navigation, etc.) is consistent with the content inside the landmark.
- If the landmark is titled (using aria-label or aria-labelledby), the title needs to be representative of the content.

Note - Do not mark a failure for missing landmarks. That is not flaggable under SC 1.3.1.

As a reminder, landmarks include:

- **Banner** - Site-level content. Content that will typically repeat on all pages like the logo and site name.
- **Complementary** - Content which is supplementary to main content, like most sidebar content
- **Contentinfo** - Site-level content (typically footer content) which typically repeat on all pages. Copyright info or other important links go here.
- **Form** - Any collection of inputs that constitutes a singular form.
- **Main** - The main content of the page. Page-specific content goes here, including the page title.
- **Navigation** - Any collection of links to different pages or different sections of the same page.
- **Region** - Any section of content important enough that a user could want to navigate directly to it. Used only when no other landmark type is relevant.
- **Search** - An attribute of role=”search” on a container element.

15

Wherever text appearance (i.e., bold, italics, underline, size) is used to convey information, that information must also be presented in text alone.

WCAG 2 criterion

1.3.1 Info and Relationships (Level A)

Tools and requirements

- Siteimprove (policy)

Test procedure

This test is strictly concerned with emphasized text which is used to describe some important detail. Most assistive technologies do not distinguish emphasized text from non-emphasized text to the user. Examples of this are using bold or italic text to denote a new addition to a list of items, or using bold text to indicate a mandatory step in a process.

- For each case of underlined text
  - Refer to Guideline 17.
- For situations where only color is used for emphasis
  - Refer to Guideline 31.
- For each case of bold or italic text
  - Ensure that the use of bold or italics does not imply a special meaning.
    - If it does imply a special meaning, ensure that the meaning is explicitly provided in text.

BAD example for "ensuring the	GOOD example for "ensuring the

meaning is explicitly provided in text” Packing list for summer camp (important items are in bold): ● **Sunscreen** ● **Swim Suit** ● **Five changes of Clothes** ● **Sleeping Bag or other bedding** ● Books ● Flashlight ● Camera ● **Medications**	meaning is explicitly provided in text” Packing list for summer camp (important items are emphasized): ● **Sunscreen (important)** ● **Swim Suit (important)** ● **Five changes of Clothes (important)** ● **Sleeping Bag or other bedding (important)** ● Books ● Flashlight ● Camera ● **Medications (important)**

16

Headings must follow appropriate relative hierarchy to other headings on the same page.

WCAG 2 criterion

1.3.1 Info and Relationships (Level A)

Tools and requirements

- WAVE

Test procedure

Using WAVE, identify the heading structure using the “Structure” tab. For each heading

- Identify the *top-level heading* on the page. It should usually be Heading 1.
- Ensure that the top-level heading content represents the page content (e.g. it should be the title of the page)
- Ensure that subheadings are appropriately assigned levels.
    - All subheadings **MUST** make logical sense as a subheading of the previous higher-level heading. (If a H4 is positioned after a H2, the H4 MUST make sense as a subheading of the H2).
    - Headings that are siblings **MUST** have the same level assignment.
    - Headings **MUST NOT** be used for presentation purposes (you can’t use a H5 simply to have smaller text for example).
    - Headings that are subheadings should be 1 level lower than the parent.
        - It is not a strict failure if this specifically is not adhered to (**do not mark a failure solely because headings are not in sequential order**), but it goes against best practices.

- Ensure that all headings make sense as headings
    - Headings must represent all the paragraph content which follows. Sometimes authors will use subheadings to add additional detail to the heading, but this is not an appropriate practice.
        - Example: An Article title is given H3. The date following is given H4. This is a failure of this criterion because the date is not the item representing the proceeding content, it is the article title. The date should not be a heading.

17

Page structure is implemented with HTML. Visual appearance is implemented with CSS. Meaningful HTML is not used to solely achieve a visual effect or appearance.

WCAG 2 criterion

1.3.1 Info and Relationships (Level A)

Tools and requirements

- Siteimprove (policy)
- Manual Testing

Test procedure

Check for the following

- Ensure that each section of the webpage that is visually identified as a landmark region is implemented as a landmark region.
- Ensure that HTML does not include any presentational attributes in situations where it is deprecated (e.g., height and width are deprecated on tables). You can utilize the Siteimprove policy “HTML used for presentation instead of CSS” to check.
- Ensure that underlining is not used for text emphasis. Underlined text is used to identify a link or other form of interactive text.

18

Form inputs must have labels which are readable (programmatically determinable) by assistive technology.

WCAG 2 criterion

1.3.1 Info and Relationships (Level A)

Tools and requirements

- Siteimprove

Test procedure

For each `<input>` or `<select>` on the page

- Ensure that there is a label, and that label is associated with the input.
- Methods below are acceptable.
    - Using the `for` attribute on the `<label>` tag to reference the `<input>` or `<select>` `id`
    - Using aria-labelledby on the `<input>` or `<select>` to reference the `id` of the `<label>`
    - Nesting the `<input>` or `<select>` tag within the `<label>` tag.
- Methods below are unacceptable.
    - Using placeholder text only as a label
    - Using plain text as a label next to the input (without using an "acceptable" method above)

19

Form input groupings (i.e., related radio buttons, related checkboxes, related text inputs like First/Last name) are grouped semantically.

WCAG 2 criterion

1.3.1 Info and Relationships (Level A)

Tools and requirements

- Siteimprove

Test procedure

This test concerns the use of multiple inputs which are part of the same group. The common input types this concerns are radio buttons and checkboxes, but it applies to any grouping of inputs.

Examples of input groupings are

- A grouping of radio buttons to select an option
- A grouping of checkboxes to select multiple options for the same query
- A first name text input and a last name text input which together create the user's Name

For any situation where inputs should be considered part of the same group

- Ensure that there is a container whose semantic purpose is to group inputs together
    - In HTML, the `<fieldset>` tag is used to group inputs together
    - Using ARIA, you can use `role=”group”` for a generic grouping of inputs

    - Using ARIA, you can use `role=”radiogroup”` for a grouping of radio buttons.
- Ensure that the grouping has a label
    - In HTML, if using the `<fieldset>` tag, a label is supplied by the `<legend>` tag. The `<legend>` tag is positioned nested within the `<fieldset>` tag.
    - Using ARIA, if `role=”group”` or `role=”radiogroup”` is used
        - `aria-label` *or* `aria-labelledby` are used to supply the label.
        - The `aria-label` or `aria-labelledby` is applied to the same tag that the `role` is applied to.

20

Data tables may only be used to present tabular data. Data tables must include table headers, which are associated with the correct table cells. Descriptive text for data tables must be programmatically associated.

WCAG 2 criterion

1.3.1 Info and Relationships (Level A)

Tools and requirements

- Siteimprove (policy)

Test procedure

Tabular data is content organized into rows and columns. Specifically, rows usually represent one item or thing, and columns represent attributes or qualities for each row item. Any tabular data must utilize a data table.

Data tables, as far as accessibility testing is concerned, are tables where table headers are utilized.

Locating data tables can be achieved through Siteimprove, with the “Find Data Tables” policy in the Policy or Accessibility > Accessibility Policies menus.

For each instance of tabular data on the page

- Ensure that the data is organized as a data table.
    - Ideally, this means using a `<table>` tag, with `<th>` tags for headers.

    - Although **HIGHLY NOT** recommended, it is also permissible to utilize ARIA on non table tags, but careful consideration must be made to ensure ALL required aria roles are applied and that assistive technology can interact with the table similarly to any other native HTML table. *(Read: MDN reference)*
- Ensure that table headers exist on at least one axis
- Ensure that table headers where you have multiple axes are properly scoped (using the scope attribute on the `<th>` tag)
- In any scenario where a table has a description available to the user, that description must be associated with the table. (For instance any text that is used as a "title" for the table, or a short description of the table's purpose.)
    - Text used to entitle the table should be a `<caption>` tag.
    - Text used to describe the table's purpose should be a `<summary>` tag.
    - `<caption>` and `<summary>` tags are not mandatory where the table's purpose can be understood from the existing table headers without needing additional context.

21

Layout tables must not include table headers, captions, or summaries. They should be marked with role="presentation".

WCAG 2 criterion

1.3.1 Info and Relationships (Level A)

Tools and requirements

- Siteimprove (policy)

Test procedure

A **layout table** is a table used to organize content visually on a page. A table which is used to present tabular data is a data table, which this guideline does not apply to.

Locating data tables can be achieved through Siteimprove, with the “Find Tables” in the Policy or Accessibility > Accessibility Policies menus.

For any layout table on the page

- There must not be any table headers on the page. Only allowable tags are `<table>`, `<tr>`, and `<td>`.
- The layout tables must be marked as role=”presentation”. That role must be applied to the `<table>` tag.

22

All elements with semantic roles contain all required parent and child elements. (e.g., a "list" must contain "listitem").

WCAG 2 criterion

1.3.1 Info and Relationships (Level A)

Tools and requirements

- Siteimprove (Next-Gen)

Test procedure

Use the Siteimprove issue check labeled "**Role not inside the required context**" and note any findings.

For manual testing, you want to spot check that these common roles have the following properties.

- **List (`<ul>`, `<ol>`, or `role="list"`)**
  - Parent Element: Any element – however a nested list must be within an `<li>` or `role="listitem"`) element from a parent list.
  - Children element(s): List item (`<li>` or `role="listitem"`)
- **Listitem (`<li>` or `role="listitem"`)**
  - Parent Element: List (`<ul>`, `<ol>`, or `role="list"`)
  - Child Element(s): Any element (although it should probably be text of some sort)
- **Radio button or Checkbox**
  - Parent element: Fieldset or Group (`<fieldset>` or `role="group"`)

- **Table (`<table>`)**
  - Parent Element: Any element
  - Child Element(s): Table row `<tr>`, `<thead>` (optional), `<tbody>` (optional), `<tfoot>` (optional)
- **Table Row (`<tr>`)**
  - Parent Element: Table (`<table>`)
  - Child Element(s): Table Header (`<th>`), Table Cell (`<td>`)
- **Table Cell or Table Header (`<td>` or `<th>`)**
  - Parent Element: Table Row (`<tr>`)
  - Child Element(s): Any element (typically just plain text)

23

Element IDs do not repeat more than once per page.

WCAG 2 criterion

1.3.1 Info and Relationships (Level A)

Tools and requirements

- Siteimprove
- WAVE
- aXe

Test procedure

Use Siteimprove or another browser extension with support for checking duplicate IDs and mark any situations where `id` values repeat more than once on a single page. This issue can be fully reliant on automated tools with no manual assessment.

Siteimprove may be the easiest method as it will list out all the pages containing the error. If you find that multiple pages present the same duplicate `id`, make note of that in reporting instead of just saying that all 'x' number of pages have `id` values that repeat.

Note: This check used to be part of 4.1.1 Parsing. The success criterion 4.1.1 Parsing is deprecated in WCAG 2.2.

24

All content is available to (readable by) assistive technology.

WCAG 2 criterion

1.3.1 Info and Relationships (Level A)

Tools and requirements

- Keyboard
- Screen Reader

Test procedure

This test is a “catch-all” requirement for any content that is not readable and/or interactable using assistive technology. Assistive technology includes but is not limited to screen readers, text readers, speech recognition/input, and keyboard input.

If you are using assistive technology, and come across any content that can’t be interacted with that does not fall into any other more specific guideline, mark it under this criterion.

Examples may include

- A CAPTCHA where no modality provided is usable by a specific assistive technology.
- `aria-hidden=”true”` is used on text that a sighted user currently has the ability to read. (e.g., a collapsed accordion should have contents hidden, an expanded accordion should not have contents hidden).

- Screen reader only text is incorrectly hidden with CSS (e.g., using `display: none;` or `visibility: hidden;` instead of using `clip`)

25

The order in which content is presented in DOM must be logical.

WCAG 2 criterion

1.3.2 Meaningful Sequence (Level A)

Tools and requirements

- Screen Reader
- WAVE

Test procedure

For all of the content on the page, the order that content is presented in code order (as opposed to visual order) must be logical. The best way to test this is to use a screen reader to skim through the content on the page to make sure that the order in which content is read aloud makes sense. You can also use WAVE to spot-check heading order, which while that won't give you the full picture of the DOM order, it will help identify if major bits of content are in the correct sequence.

Logical **does not** mean that the reading order must follow the focus order. A screen reader may read content in a different order than content receives keyboard focus. (Generally, this is a bad practice, but it is allowable at Level A in WCAG.)

Logical **does not** mean it must follow a regular Top to Bottom, Left to Right order at all times. Consider a page layout with a banner, a main content area, a sidebar, and a footer where the main content area and sidebar are adjacent horizontally to each other. It can make sense for content in the sidebar to the right to be read aloud first if it is perhaps the location of the table of contents for the page. It could conversely make sense for that content to be read aloud second if there was supplementary content there. This judgment must be made by the reviewer.

26

Whitespace is not utilized to create text spacing within a word. Whitespace is not utilized to create columns or tables visually in plain text.

WCAG 2 criterion

1.3.2 Meaningful Sequence (Level A)

Tools and requirements

- Screen Reader
- Siteimprove (policy)

Test procedure

When browsing content, be sure that there are no cases where whitespace is used for text spacing or formatting.

There is a policy in Siteimprove called “Potential overuse of whitespace.” It searches for situations where there are 3 or more spaces in a row (which would be indicative of using whitespace to create visual columns). It will not be able to search for scenarios where there is space between letters in a word.

This should be among the checks that you perform on the subset of pages that manual testing is performed on.

27

Instructions for operating web-based content and cues for identifying content does not rely exclusively on color, shape, size, position, or sound. (Above/Below references allowed)

WCAG 2 criterion

1.3.3 Sensory Characteristics (Level A)

Tools and requirements

- Manual Evaluation
- Siteimprove (Policy)

Test procedure

Instructions in the context of this criterion references web-based content. It does not apply to instructions relating to performing actions in the "real world."

This test is not concerned with text that is not part of an instruction. It is okay to reference colors or directional relationships as long as that text is not an explanation of how to perform a web-based task. (For instance, you can say that "The field is covered in gold flowers with a brown barn in the middle" since you aren't trying to inform a user about how you interact with web content).

Visual cues are also included in this criterion. A visual cue would for example be like bold text to bring attention to certain text. It is not a failure of this criterion to use visual cues, as long as the visual cue is not attempting to convey important information not otherwise available in text. (For example, you can use bold text to emphasize a term or important bit of text to the user, but you cannot say “items in bold are important” or “bold form fields are required”)

Whenever a web-based instruction or visual cue is found:

- Ensure that color, shape, size, position, or sound are not exclusively used to provide instructions to the user.
    - If the instruction or cue provides an additional descriptor aside from one reliant on specific senses, then that would not count as a failure of this criterion. (Example: “Click the green button” versus “Click the green **submit** button”
    - The lone exception to this rule in regards to instructions is when the language references “above” or “below” which imply before or after in sequence (assuming it references something before or after in DOM order, not visual order)

28

Content is viewable in portrait and landscape device orientations, and the user is not prompted to switch orientation unless a specific orientation is essential.

WCAG 2 criterion

1.3.4 Orientation (Level AA)

Tools and requirements

- Manual Evaluation
- Mobile web browser or a browser that can emulate screen orientation

Test procedure

For this test, you must use a mobile browser (or BrowserStack) on a native device with orientation controls.

Test to ensure that there is no scenario where content is completely not accessible to the user unless the screen is in a specific orientation when using one of these technologies.

The limited exceptions for essential orientation lock apply when the content would only be understood or presented in a particular orientation (such as a TV or museum display). If you had content that was hosted on the web, but would only for instance ever be presented on a tablet display locked to an orientation, you can lock orientation.

Testing using BrowserStack (or native mobile OS web browser)

If using BrowserStack, turn on one of the Android-based mobile devices, and use Chrome.

Within the mobile web browser, go to the site you are testing and ensure that when swapping between landscape and portrait modes the webpage does not lock to one orientation.

Note on orientation emulation

Some browsers like Chrome have features that allow you to emulate orientation, however this is designed around functionality that utilizes the motion of a mobile device. You should continue to use a native mobile device to test Orientation.

29

The purpose of any form input about the user is identified in code when the purpose is defined in HTML.

WCAG 2 criterion

1.3.5 Identify Input Purpose (Level AA)

Tools and requirements

- Manual Evaluation
- Siteimprove (Policy)
- Web browser inspector tool

Test procedure

This test asks you to evaluate webforms. You can utilize the Siteimprove policy "Find Forms" to find forms to evaluate. When you come across a webform, if any field asks for **data about the user** then that field must autocomplete functionality.

Autocomplete support is achieved with the aptly named "`autocomplete`" attribute in HTML. The `autocomplete` attribute is applied to the tags `<input>`, `<select>`, `<textarea>`, and `<form>`. The value of the autocomplete attribute must match the value referenced in the W3C WCAG 2.2 official document (https://www.w3.org/TR/WCAG22/#input-purposes).

Examples

Example for a text input asking for the user's first name:

```
<input type="text" id="first-name" autocomplete="given-name" />
```

Example for a text input asking for the user’s email address:

```
<input type=”text” id=”user-email” autocomplete=”email” />
```

Example for a select box asking for the user’s state of residence:

```
<select name=”state” id=”address-state”
autocomplete=”address-level1” />
```

Names and Addresses

Names can be implemented in two different ways. The first is asking for the user’s full name in one input field. The second is asking for each name in separate input fields. Either option is completely valid, but it will change the autocomplete values you would expect to see.

Addresses can be a bit more complex when it comes to ensuring that the appropriate autocomplete attributes are used. Normally, addresses in forms are going to be split into multiple different inputs. You’ll have one input for the street name and number, a separate input for the town or city, another separate one for the state/province, and another for the zip code.

You would want to see the following autocomplete values for names and addresses (note: US only, different countries have different rules for addresses). The table below is *not* exhaustive. Reference the official W3C WCAG 2.2 documentation for more details.

Form Field	**Expected Autocomplete Value**
User’s Name (if asking for the full name in one input)	name
User’s First Name	given-name
User’s Middle Name	additional-name
User’s Last Name	family-name

Form Field	**Expected Autocomplete Value**
User’s Suffix (Jr./Sr./II/III/IV etc.)	honorific-suffix
Address Street Name	street-address
Address City/Town Name	address-level2
Address State	address-level1
Address Zip Code	postal-code
Address Country	country-name (full name) or country (code)
Email Address	email

30

Color may not exclusively distinguish between plain text and interactive text or distinguish one type of content from another without a 3:1 color contrast difference.

WCAG 2 criterion

1.4.1 Use of Color (Level A)

Tools and requirements

- Siteimprove
- Manual Evaluation
- Color Contrast Comparison tool

Test procedure

For this criterion, evaluate if there are any situations where Interactive text is visually identified using color alone, or if there are more than one different types of content which are only distinguishable from one another by a difference in color.

Interactive text with only a color difference is defined as having the exact same styling (same font, same font size, same text decorations (if any), and same emphasis or bolding (if any) aside from a color shift.

Below are visual examples of text differences that are **not** evaluated under this criterion.

- The concert starts at 9pm, you still have time to **purchase tickets**!
- The concert starts at 9pm, you still have time to *purchase tickets*!

- The concert starts at 9pm, you still have time to purchase tickets!
- The concert starts at 9pm, you still have time to `purchase tickets`!
- The concert starts at 9pm, you still have time to purchase tickets!

An example where color is used to distinguish between two different states or types of content could be a page where "Cornell alumni are identified in red and Yale alumni are identified in blue." This scenario applies if the only visual distinction is a change of color, regardless of *what* is changing color. (A scenario where two different states/types of content are identified by a background color shift are evaluated no differently than if the color of the text itself were the only difference between the two.)

For any interactive text that is only identified by a color shift, or where one state/type of content is visually identified by a color shift alone:

- Ensure that the color contrast difference of the **interactive text** is a 3:1 color contrast difference from the **surrounding text.**
    - Color contrast requirements against the background still apply, but any failures of this would be marked in Guideline 33 or 34: 1.4.3 Contrast (Minimum)
- Ensure that the interactive text gains a different visual indicator when focused and hovered over.
    - Examples of a different visual indicator would include an outline or underlined text.

Mark a **failure** if either of the above two criteria are not met.

31

Color may not exclusively identify content or distinguish differences in any content (e.g. Red items are invalid, Green items are valid)

WCAG 2 criterion

1.4.1 Use of Color (Level A)

Tools and requirements

- Manual Evaluation

Test procedure

During the course of manual evaluation, take note of situations where you see differently colored text compared to the surrounding text.

When coming across text with a different color

- Ensure the color shift is not used to exclusively compare or call out text as important or different from other text.
    - If color is not exclusively used to call out text as different, it does not fall under this criterion (e.g., if text and color are both used, then it would not be a failure).

32

Auto-playing audio that lasts longer than 3 seconds must be pausable OR have an independent volume control.

WCAG 2 criterion

1.4.2 Audio Control (Level A)

Tools and requirements

- Manual Evaluation
- Siteimprove to help identify audio sources

Test procedure

Whenever a source of audio is found that lasts over 3 seconds

- Ensure there is a control to mute, pause, or disable the audio source.
    - The control must be keyboard and screen reader accessible, and it also must be a **button.**

33

Large-scale (24px or 19px bold) text must have a color contrast ratio of 3:1. Logos, inactive components, and pure decoration excluded.

WCAG 2 criterion

1.4.3 Contrast (Minimum) (Level AA)

Tools and requirements

- A color picker of your choice (where you can select colors with an eyedropper tool)
- Siteimprove
- WAVE

Test procedure

Color contrast should primarily be measured using a color picker tool, and then evaluating the contrast between the foreground and background. This is because automated tools like Siteimprove are unable to check color contrast when the text background is a gradient or image.

For this criterion, any text that is part of a logo, or any text that is part of a disabled or inactive form field, button, or link, and any text that is pure decoration and adds zero information to the site are **NOT** evaluated for color contrast.

Large scale text is defined as text that is at least a font size of 24px (18pt) with a font weight at or above 700, **or** text that is at or above a font size of 19px (14pt) regardless of font weight.

For large-scale text, the color contrast requirement is 3:1. You can use color contrast checker tools like the ones on WebAIM if your color picker tool doesn't already support checking the contrast ratio between the foreground and background.

If you come across **large-scale** text that does not pass the 3:1 contrast ratio test, mark it as a **failure** on the checklist.

34

Non Large-scale text must have a color contrast ratio of 4.5:1. Logos, inactive components, and pure decoration excluded.

WCAG 2 criterion

1.4.3 Contrast (Minimum) (Level AA)

Tools and requirements

- A color picker of your choice (where you can select colors with an eyedropper tool)
- Siteimprove
- WAVE

Test procedure

Color contrast should primarily be measured using a color picker tool, and then evaluating the contrast between the foreground and background. This is because automated tools like Siteimprove are unable to check color contrast when the text background is a gradient or image.

For this criterion, any text that is part of a logo, or any text that is part of a disabled or inactive form field, button, or link, and any text that is pure decoration and adds zero information to the site are **NOT** evaluated for color contrast.

Non large-scale text is defined as text that is below a font size of 24px (18pt) with a font weight under 700, **or** text that is below a font size of 19px (14pt) regardless of font weight.

For non-large-scale text, the color contrast requirement is 4.5:1. You can use color contrast checker tools like the ones on WebAIM if your color picker tool doesn't already support checking the contrast ratio between the foreground and background.
Above: An example of the Color Contrast Analyzer plugin testing for 4.5:1 contrast.

If you come across non-large-scale text that does not pass the 4.5:1 contrast ratio test, mark it as a **failure** on the checklist.

35

Text can be resized up to 200% without page content disappearing or losing functionality

WCAG 2 criterion

1.4.4 Resize Text (Level AA)

Tools and requirements

- Web browser with built-in zoom text only feature (Firefox, Safari)
- Zoom Text Only plugin for Chrome

Test procedure

There are two ways to test zooming text to 200%. Only one of the two methods must meet both verification checks to pass.

Zooming the entire page

Use the standard keyboard shortcuts of ⌘= (Command + Equals) or CTRL= (Control + Equals) to zoom the entire page.

Zoom text only

- **Firefox**
    - In the menu bar, select **View > Zoom > Zoom Text Only**
- **Safari**
    - Use the shortcut ⌥⌘= (Option + Command + Equals)
- **Chrome or Edge**
    - Use an extension such as Zoom Text Only or Text Zoom from the Chrome Web Store

Verification Steps

- **Check Scaling:** The text must appreciably scale when testing. It must not be locked to a fixed size when scaling the page.
- **Check Content Visibility:** No content should disappear or become unusable when zoomed.

36

Text can be resized up to 200% without text clipping through other elements.

WCAG 2 criterion

1.4.4 Resize Text (Level AA)

Tools and requirements

- Web browser with built-in zoom text only feature (Firefox, Safari)
- Zoom Text Only plugin for Chrome

Test procedure

There are two ways to test zooming text to 200%. Only one of the two methods must meet both verification checks to pass.

Zooming the entire page

Use the standard keyboard shortcuts of ⌘= (Command + Equals) or CTRL= (Control + Equals) to zoom the entire page.

Zoom text only

- **Firefox**
  - In the menu bar, select **View > Zoom > Zoom Text Only**
- **Safari**
  - Use the shortcut ⌥⌘= (Option + Command + Equals)
- **Chrome or Edge**
  - Use an extension such as Zoom Text Only or Text Zoom from the Chrome Web Store

Verification Steps

- **Check Scaling:** The text must appreciably scale when testing. It must not be locked to a fixed size when scaling the page.
- **Check for Clipped or Overlapping Text:** No content should clip (be “cut off” by another element), or overlap other elements when zoomed.

37

Images of text are not used when the same presentation can be made with native HTML/CSS. Logos and branding are excluded.

WCAG 2 criterion

1.4.5 Images of Text (Level AA)

Tools and requirements

- Manual Evaluation

Test procedure

When evaluating pages within the scope of manual evaluation, check for cases where images exist that include text. Report images of text where the text in the image can be replicated using HTML and CSS. Possible examples may include:

- Text over a banner image
- Text with transparency
- Text with gradients
- Text made to appear as a button

Examples of images of text that need to be flagged include

- Banner/Carousel images that include a lot of text where there is no alternative text version
- Images of documents that lack a text alternative
- Images of text that serve as buttons or controls

Examples of images of text that should **NOT** be flagged include

- Text in the background of the primary subject of the image (like a STOP sign in the background.
- Logos and branding.
- Handwriting or similar text that you cannot display using HTML and CSS.

38

Content may only scroll in one dimension (horizontal or vertical) at a width and height equivalent of 320x256 pixels or larger. Excluded is content where a two-dimensional layout is necessary (video, data tables, maps, diagrams)

WCAG 2 criterion

1.4.10 Reflow (Level AA)

Tools and requirements

- Manual Evaluation

Test procedure

Using the Chrome web inspector (F12), the easiest way to test for this criterion is to open the Responsive design preview menu. ( ) Within that menu, set the dimensions to a width of 320px and a height of 256px.

After setting those dimensions, ensure the entire page only scrolls in one direction (either horizontally or vertically). Only certain elements that require a two-dimensional layout are allowed to scroll in two directions. Examples include data tables, maps, and diagrams.

39

Active user interface components must meet a 3:1 color contrast ratio. (This includes buttons, inputs, custom focus indicators, dropdowns, checkboxes, and radio buttons.)

WCAG 2 criterion

1.4.11 Non-text Contrast (Level AA)

Tools and requirements

- Color contrast tool

Test procedure

Note: This is a highly complex criterion. Please read through the official documentation for full details.
https://www.w3.org/WAI/WCAG22/Understanding/non-text-contrast.html

When testing color contrast, evaluate buttons, input fields, focus indicators, dropdown (select) boxes, checkboxes, and radio buttons for a 3:1 contrast ratio using a color contrast tool of your choice.

Buttons must have either the text or icon part of the button have a 3:1 contrast ratio against the background.

Input Fields and Dropdowns must have a 3:1 contrast ratio against the surrounding color. (Or there must be an outline that has a 3:1 contrast ratio). Dropdowns usually have an arrow to indicate that the element is a dropdown box, and the arrow or other icon used to tag it as a dropdown box must have a 3:1 contrast ratio.

Focus indicators must have a 3:1 contrast ratio **if they are not the browser default**. You compare the focus indicator to the adjacent color. (Ergo, if the focus indicator is outside the element, you compare it to the page background color. If the focus indicator is inside the element, you compare it to the background color of the element.)

Checkboxes and Radio Buttons must have a 3:1 contrast ratio against the background.

40

Graphical objects that describe important content must meet a 3:1 color contrast ratio; except flags, real-life imagery, branding, reference screencaps, and heatmaps.

WCAG 2 criterion

1.4.11 Non-text Contrast (Level AA)

Tools and requirements

- Color Contrast Analyzer tool

Test procedure

Note: This is a highly complex criterion. Please read through the official documentation for full details.
https://www.w3.org/WAI/WCAG21/Understanding/non-text-contrast.html

When testing color contrast, evaluate graphical objects for 3:1 contrast. Graphical objects that get tested under this guideline include icons (like social media icons, or icons used to describe an interactive element) and charts (like line charts and pie charts).

Icons must have a 3:1 contrast against the background of the icon.

For **line charts**, all lines (including the graph lines marking the gradations) must meet a 3:1 contrast ratio.

For **pie charts**, all pie slices must have a 3:1 contrast ratio (if there is no supplemental information like percentages provided alongside the labels) against whatever color surrounds it (regardless of if the adjacent color is another pie slice or the background color of the chart).

Flags, real-life images, branding, screencaps, and heatmaps are all excluded from this test.

41

No content or functionality may be lost when text is set to: Line spacing of 1.5x font size, Letter Spacing at 0.12x font size, Word spacing at 0.16x font size, and paragraph spacing 2x the font size.

WCAG 2 criterion

1.4.12 Text Spacing (Level AA)

Tools and requirements

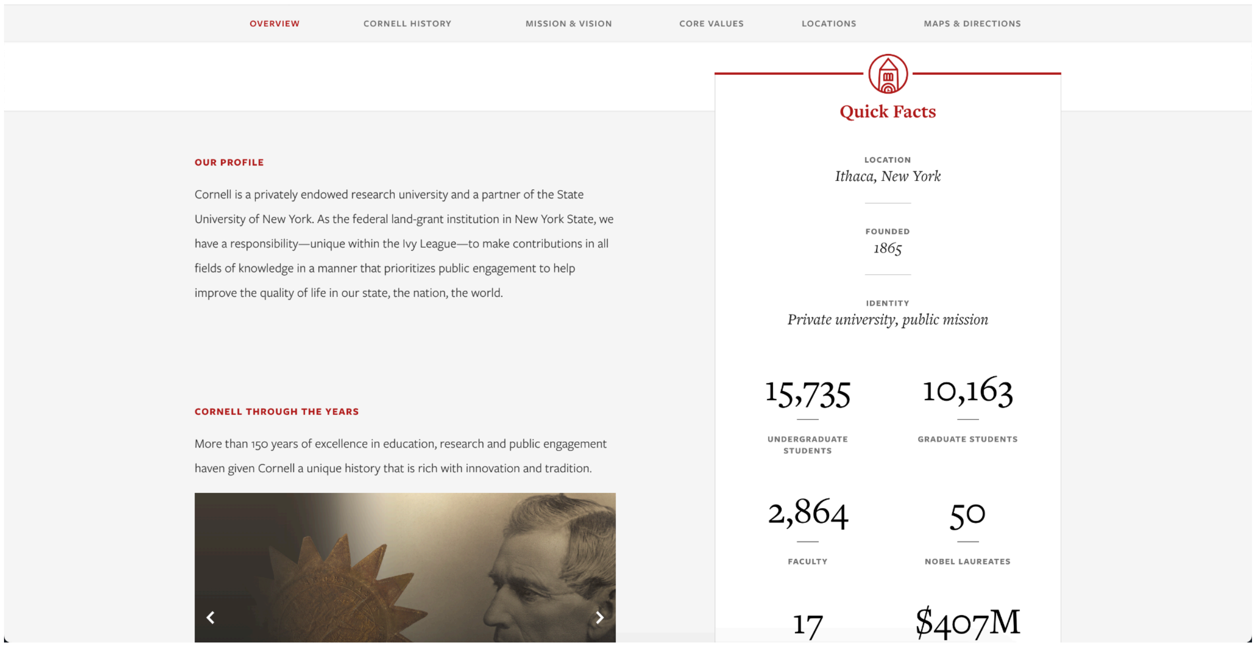
- Text Spacing Bookmarklet https://codepen.io/stevef/full/YLMqbo

Install the above bookmarklet by enabling your bookmarks bar in your browser, and dragging the "Text Spacing" link provided on that Codepen page into your bookmarks bar. Activate and deactivate it by clicking on the bookmark in your bookmarks bar.

Test procedure

Using the Text Spacing bookmarklet, ensure that no text content disappears or clips through other objects/text. The bookmarklet adjusts text to the maximum limits defined by the criterion.

See below for a visual example of the Text Spacing bookmarklet.


OVERVIEW CORNELL HISTORY MISSION & VISION CORE VALUES LOCATIONS MAPS & DIRECTIONS

OUR PROFILE

Cornell is a privately endowed research university and a partner of the State University of New York. As the federal land-grant institution in New York State, we have a responsibility—unique within the Ivy League—to make contributions in all fields of knowledge in a manner that prioritizes public engagement to help improve the quality of life in our state, the nation, the world.

CORNELL THROUGH THE YEARS

More than 150 years of excellence in education, research and public engagement haven given Cornell a unique history that is rich with innovation and tradition.

Quick Facts

LOCATION
Ithaca, New York

FOUNDED
1865

IDENTITY
Private university, public mission

15,735 UNDERGRADUATE STUDENTS

10,163 GRADUATE STUDENTS

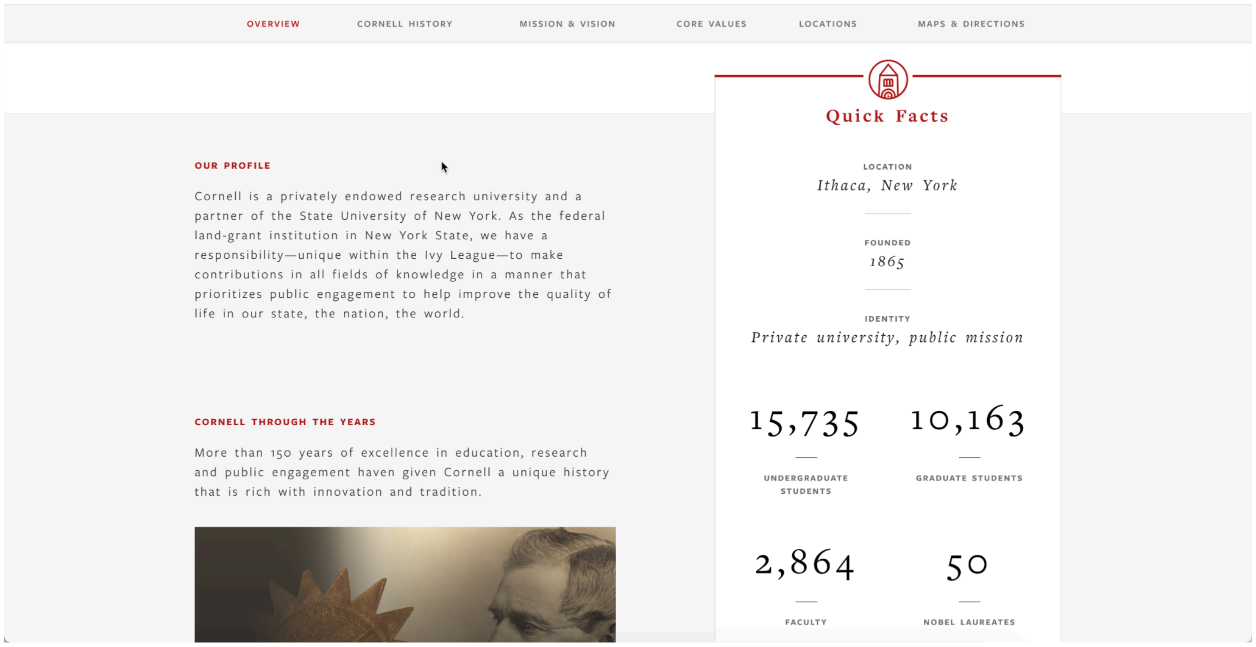
2,864 FACULTY

50 NOBEL LAUREATES

17

$407M

Figure 41.1 - Normal View of https://www.cornell.edu/about/


OVERVIEW CORNELL HISTORY MISSION & VISION CORE VALUES LOCATIONS MAPS & DIRECTIONS

OUR PROFILE

Cornell is a privately endowed research university and a partner of the State University of New York. As the federal land-grant institution in New York State, we have a responsibility—unique within the Ivy League—to make contributions in all fields of knowledge in a manner that prioritizes public engagement to help improve the quality of life in our state, the nation, the world.

CORNELL THROUGH THE YEARS

More than 150 years of excellence in education, research and public engagement haven given Cornell a unique history that is rich with innovation and tradition.

Quick Facts

LOCATION
Ithaca, New York

FOUNDED
1865

IDENTITY
Private university, public mission

15,735 UNDERGRADUATE STUDENTS

10,163 GRADUATE STUDENTS

2,864 FACULTY

50 NOBEL LAUREATES

Figure 41.2 - Appearance of https://www.cornell.edu/about/ when the Text Spacing bookmarklet is used.

42

Content generated by hover or focus can be dismissed without moving hover or focus.

WCAG 2 criterion

1.4.13 Content on Hover or Focus (Level AA)

Tools and requirements

- Keyboard

Test procedure

When interacting with elements on the page, if any content appears on hover or focus **and** disappears when hover or focus leaves (like a dropdown or a tooltip).

- First verify that the content disappears when hover or focus leaves. If the content is persistent on mouseout/focusout events, this criterion is inapplicable.
- Ensure that the item can be dismissed without moving hover or focus. The most common way of making this happen is by making the ESC key get rid of the popup.

43

Content generated by hover or focus of an element can be hovered over without the content disappearing.

WCAG 2 criterion

1.4.13 Content on Hover or Focus (Level AA)

Tools and requirements

- Keyboard

Test procedure

When interacting with elements on the page, if any content appears on hover or focus **and** disappears when hover or focus leaves (like a dropdown or a tooltip).

- First verify that the content disappears when hover or focus leaves. If the content is persistent on mouseout/focusout events, this criterion is inapplicable.
- Ensure that the dropdown or tooltip can be hovered over itself without the content disappearing. Move your cursor from the element that caused the popup (the triggering element) over the dropdown or tooltip.

44

Content generated by hover or focus of an element does not disappear until dismissed, is no longer valid, or hover or focus is removed.

WCAG 2 criterion

1.4.13 Content on Hover or Focus (Level AA)

Tools and requirements

- Keyboard

Test procedure

When interacting with elements on the page, if any content appears on hover or focus **and** disappears when hover or focus leaves (like a dropdown or a tooltip).

- First verify that the content disappears when hover or focus leaves. If the content is persistent on mouseout/focusout events, this criterion is inapplicable.
- Ensure that the dropdown/tooltip does not disappear on its own without you taking an action to explicitly remove it like using ESC or moving away from the triggering element.

45

All interactive elements must be able to be navigated to and interacted with using a Keyboard only.

WCAG 2 criterion

2.1.1 Keyboard (Level A)

Tools and requirements

- Keyboard

Test procedure

Note: On macOS specifically, you may need to enable certain settings for keyboard navigation to work as expected. Go to your Mac Settings, click “Keyboard”, click “Shortcuts” and check the “Use keyboard navigation to move focus between controls”

Starting at the top of the page you are evaluating, continuously hit the TAB key and ensure that you can focus on every element on the page that is interactive. This means, all buttons, all links, all form controls, and any other elements that the developer assigned a tabindex value of 0 or higher to.

Some elements may not be visible when you focus on them. Any of those cases need to be marked as part of 2.4.7 Focus Visible. Sometimes you *can* interact with certain elements even though you are not provided with a visual indication. This guideline is specifically looking for the ability of a keyboard to interact with all interactive elements.

Interacting with elements using a keyboard is dependent on what type of element it is. Below is a list of patterns and how a keyboard is expected to interact with them.

- **Link:** Use the RETURN/ENTER key.
- **Button:** Use either the RETURN/ENTER key or the SPACEBAR. (Both must work to be compliant.)
- **Text Input**: Type once the input has focus.
- **Radio Button:** Use arrow keys to switch between options. (You should NOT use TAB to switch between radio buttons in the same grouping.)
- **Checkbox:** Use RETURN/ENTER or SPACEBAR to tick/untick checkboxes. Checkboxes as part of a group are navigated to using arrow keys.
- **Select Box:** Use RETURN/ENTER or SPACEBAR to open the select box, then use arrow keys to move up and down.
- **Combobox**: Use RETURN/ENTER or SPACEBAR to open the combobox. The text box can be typed in normally, and arrow keys can be used to scroll through the options.
- **Tablist:** Move keyboard focus onto the grouping of tabs. Use Arrow Keys to select the tab you wish to open. Use RETURN/ENTER or SPACEBAR to select the tab. Use TAB to immediately move into the content. (If you have to use the TAB key to navigate through all of the tabs, then it is a failure.)
- **Accordion:** Use RETURN/ENTER or SPACEBAR to expand or collapse the accordion.
- **Carousel**: Use the TAB key to navigate between all of the controls. Use RETURN/ENTER or SPACEBAR to interact with the controls. The use of arrow keys to switch between slides is optional, but if the slide changes, the focus must shift to the active slide instead of being locked to the now-hidden slide.
- **Navigation:** Depending on the implementation, use the TAB key and/or the Arrow keys to select the navigation item, then use the RETURN/ENTER key to click the link. If the navigation offers drop-down functionality, either a button should be available to use RETURN/ENTER or SPACEBAR to open the dropdown, or arrow keys may be used to enter the dropdown. Use the ESC key to return to the parent navigation item, or use the TAB key to navigate out of the dropdown.

- **Modal**: The trigger for a modal should be a button, and therefore triggered using either RETURN/ENTER or SPACEBAR. Within the modal, use TAB to switch between elements, or ESC to immediately close the modal. Modals may also offer an explicit close button which may also be interacted with using RETURN/ENTER or SPACEBAR. If you can use TAB to escape the modal without explicitly closing it, mark a failure under 2.4.3 Focus Order

46

Timing of keystrokes must not be required for interacting with any functionality.

WCAG 2 criterion

2.1.1 Keyboard (Level A)

Tools and requirements

- Keyboard

Test procedure

Note: On macOS specifically, you may need to enable certain settings for keyboard navigation to work as expected. Go to your Mac Settings, click “Keyboard”, click “Shortcuts” and check “Use keyboard navigation to move focus between controls.” On Mac Ventura, go to System Settings > Keyboard Navigation and turn Keyboard navigation on (to use keyboard navigation to move focus between controls).

When using the TAB key to navigate through the page, make note of any situation where you have to hit TAB within a certain window of time to activate. If for example, you can only interact with an element for 5 seconds before the element disappears or can’t be interacted with anymore, that would constitute a failure of this criterion.

This guideline is not looking for situations where certain elements are hidden away until an explicit action is taken by the user. For example, if you activate a modal with the keyboard, new focusable elements may appear that weren’t available before. But, as long as the modal does not disappear on its own, the user has no time constraint in interacting with the modal.

47

Functionality available when hovering with a cursor must be available to keyboard input.

WCAG 2 criterion

2.1.1 Keyboard (Level A)

Tools and requirements

- Keyboard

Test procedure

When manually evaluating a site, take note of situations when functionality becomes available on hover. Any time a menu or popup appears (or any other function) is triggered by hovering over an element, make sure that a keyboard can perform the exact same action.

One notable example is that large navigation menus need to let keyboard users choose which dropdown they want to open and NOT force keyboard users to navigate through every dropdown before coming across the primary dropdown they want to interact with.

48

Focus that enters any element must be able to leave that element. If the method requires more than ESC, Arrow Keys, or Tab, the user must be informed of the method.

WCAG 2 criterion

2.1.2 No Keyboard Trap (Level A)

Tools and requirements

- Keyboard

Test procedure

When using the TAB key to navigate pages while testing for the 2.1.1 Keyboard criterion, take a note if the keyboard focus gets trapped on an element or in a loop that can't be escaped from.

It is required that if the method of leaving the keyboard trap requires the use of other keys aside from the Escape key, Arrow keys, or TAB/SHIFT + TAB, there must be an instruction that is visible and screen reader accessible indicating what is needed to leave the keyboard trap and return the user to where they were beforehand.

49

If any keyboard shortcut only requires letter, number, punctuation, or symbol characters, an option exists to turn it off, remap to include CTRL or ALT modifiers, OR be active on focus only.

WCAG 2 criterion

2.1.4 Character Key Shortcuts (Level A)

Tools and requirements

- Keyboard

Test procedure

When using the keyboard to navigate through a webpage, check to see if any part of the page supports running functionality with any letter, number, punctuation, or symbol key on its own. (Arrow keys are NOT part of this criteria)

If there are functions that work using a letter, number, punctuation, or symbol key, mark a **failure** *unless* one of the following conditions are met

- The function also requires the use of the CTRL or ALT (for Windows) or COMMAND or OPTION (for Mac).
- The function only works when the user is focused on the element.
- There is a way to disable the function.

You may come across this on certain virtual 3D room pages, or on certain maps.

50

Any time limit may be disabled, extended (with a 20 second warning), or adjusted; unless it is part of a current real-life event, it is essential, or has a time limit of 20 hours or more.

WCAG 2 criterion

2.2.1 Timing Adjustable (Level A)

Tools and requirements

- Manual Assessment

Test procedure

If any functionality on the site is time sensitive (you only have so long to complete it before you are kicked out or miss out on the ability to proceed), mark a **failure** *unless* one of the following conditions are met.

- The time-sensitive content is part of a current live event (such as an auction).
- The time-sensitive content allows you to extend or disable the timer.
- The time-sensitive content has a time limit at or above 20 hours.
- The time-sensitive content is absolutely essential (such as a time limit on an exam).

51

Any moving, blinking, or scrolling information that starts automatically, lasts over 5 seconds, and is part of other content must include a pause, stop, or hide mechanism.

WCAG 2 criterion

2.2.2 Pause, Stop, Hide (Level A)

Tools and requirements

- Manual Assessment

Test procedure

If any moving, flashing, or blinking content is encountered which lasts over 5 seconds, mark a **failure** *unless* one there is a function available to pause, fully stop, or hide away the moving content.

- This function needs to be explained to the user in text.

52

Any automatically updating content that starts automatically and is part of other content must include a pause, stop, or hide mechanism.

WCAG 2 criterion

2.2.2 Pause, Stop, Hide (Level A)

Tools and requirements

- Manual Assessment

Test procedure

If there is any content that has live updates (for instance, using AJAX), mark a **failure** *unless there* is a function available to pause, fully stop, or hide away the live updates.

- This function needs to be explained to the user in text.

53

No content may flash more than 3 times per any 1-second period.

WCAG 2 criterion

2.3.1 Three Flashes or Below Threshold (Level A)

Tools and requirements

- Manual Assessment

Test procedure

If there is any content on the page that flashes or strobes more than 3 times on a page (see the “Below Threshold” provision below for more details), mark a **failure**. There is no provision for pausable content for this criterion.

This applies primarily to multimedia content.

“Below Threshold” provision

The flashing according to WCAG 2 is permissible if the flashing is below a certain threshold, defined by a mathematical equation. For our purposes at Cornell, we simply flag any content that contains tons of flashing of any kind greater than 3 times per second a failure.

54

Content which repeats on multiple webpages has a mechanism of skipping over it.

WCAG 2 criterion

2.4.1 Bypass Blocks (Level A)

Tools and requirements

- Keyboard
- Screen Reader

Test procedure

On a set of web pages within a website or web application, if there is any content that repeats multiple times, ensure that there is a link that can be used to skip over the repeated content. For most cases, it should be the first link you come across when you hit the TAB key once using a keyboard, or the first element a screen reader announces. That link should direct you to the first element after all repeated content.

Repeated content includes logos, navigation elements, search fields, and text that is located in the same place on multiple pages.

55

All pages have a title, which details the topic or purpose of the page. Titles should be organized from most-specific to least-specific.

WCAG 2 criterion

2.4.2 Page Titled (Level A)

Tools and requirements

- Siteimprove
- Manual Evaluation

Test procedure

Siteimprove can test for the absence of a title. Any findings that Siteimprove reports for a lack of title should be reported under this criterion.

When doing manual testing, make note of the title of each page you assess. The title should be in order from most specific to least specific. That is to say, it should be in a <Page title> <Section title(s)> <Site title> order. If it is in reverse order, or the page title is not accurate, mark a **failure** for this criterion.

56

Focus order must follow a logical sequence. Tabindex values must not interfere with the proper tab sequence of the page.

WCAG 2 criterion

2.4.3 Focus Order (Level A)

Tools and requirements

- Keyboard

Test procedure

When using a keyboard to navigate through the web page, ensure that the order that elements receive focus makes sense considering the visual layout. Most elements should be given focus in a left-to-right, top-to-bottom order. If you have columns, focus should navigate through one whole column before moving on to the next one.

Failures should be marked in situations where the focus order jumps around the page without much explanation, or in situations where the keyboard focus doesn't follow an expected keyboard navigation pattern. Certain design patterns have different expected behaviors for keyboard focus. Make sure that the following elements follow these rules.

Tablists

A tablist is a grouping of controls that swaps content in a singular section of the page. (An accordion by contrast opens content in its own section, which can be expanded or collapsed.)

A tablist should be treated as one group. When you keyboard navigate into a tablist, the focus should be sent to the entire grouping of tabs, and arrow keys should then be utilized to switch between each different tab. After selecting the tab the user wants to interact with, the next press of the TAB key should send the user into the tab-panel content.

If a tablist forces the user to use the TAB key to navigate through the entire list of tabs before accessing the tab-panel content, or if the user is forced to navigate through each tab's content before moving to the next tab, it is a **failure.**

Accordion

Accordions should not open automatically. Each accordion should only open when explicitly told to by the user. Furthermore, the contents of the accordion should not receive any keyboard focus until opened.

Carousel/Slider

A carousel/slider/gallery should send focus to the active slide's contents first. Then, keyboard focus should be sent to the controls. Usually this would be a pair of previous and next buttons, followed by direct navigation buttons if applicable.

If previous or next buttons are used, the focus should remain on the previous and next buttons respectively (aria-live should announce the slide contents to a screen reader). If a direct navigation button is activated, the focus should be directed back to the slide contents immediately.

Tabindex

When you are using keyboard navigation, take note of any situations where the focus order does not follow an expected path (e.g., the focus is jumping up and down the page or between multiple different areas of the page).

When keyboard focus is not in a logical order, evaluate the HTML markup to see if tabindex values (specifically positive values of 1 or higher) are used. If positive tabindex values are used (which overrides the tab order and causes poor keyboard focus order), then it would be considered a **failure** of this criterion.

57

Dialog content must gain focus immediately or as the next press of TAB once activated. Dismissing dialog content returns focus to the trigger or to the next element in DOM after the trigger.

WCAG 2 criterion

2.4.3 Focus Order (Level A)

Tools and requirements

- Keyboard
- Screen Reader

Test procedure

Using a keyboard, if you encounter a dialog/modal dialog, then use a combination of TAB and SHIFT+TAB to ensure that the keyboard focus remains within the modal until closed with a close button, or the ESC key.

Using a screen reader, ensure that content outside the modal cannot be read until the modal is closed. Furthermore, if there is text which describes the purpose of the dialog, ensure that the text is automatically announced once the dialog is opened. If it is not announced, that static text should be wrapped in a div referenced by `aria-describedby`.

Keyboard focus and reading order with a screen reader should loop through the modal infinitely until closed.

Any modal that allows keyboard focus to escape the modal without closing it would be considered a **failure** of this criterion.

58

Link destination is described by link text on its own or by link text programmatically associated with other text on the page (except where the destination is ambiguous to all users).

WCAG 2 criterion

2.4.4 Link Text in Context (Level A)

Tools and requirements

- Siteimprove
- Manual Assessment

Test procedure

This criterion checks to ensure that link text describes the link destination. There are multiple checks to consider for this criterion.

1. The link text describes the destination

Anytime a link is encountered, take note of where the link navigates the user. Then, observe the link text used for that link. The link text needs to represent the destination of the link, or the link plus the surrounding context must supply the destination of the link.

A link with appropriate context describes the link destination when one (or more) of the following scenarios is met

- The link text on its own describes the link destination

- The link text combined with contents of the same list item and/or the parent list item describes the link destination
- The link text combined with contents of the same paragraph tag describes the link destination. (Note, this context should be ideally **before** the link itself)
- The link text combined with contents of the same table cell, or the table cell plus the contents of the associated table headers.

Examples of appropriate and inappropriate link text are demonstrated below.

Appropriate Example - Link text on its own describes the link destination

Our financial information disclosure can be found on our **2022 Annual Report**. If you have any questions about the financial report, please **email John C. Smith**.

Appropriate Example - Link text combined with the contents of the same or parent list item describes the link destination.

- Annual Report 2022
  - **Web Version**
  - **PDF Version**
  - **Word Doc Version**
- Annual Report 2021
  - **View** the web version
  - **View** the PDF version
  - **View** the Word Doc Version
- Annual Report 2020 can be viewed in the **archive**

Annual Report 2022 examples use link text of "Web version," "PDF version," and "Word Doc Version" which on their own do not adequately describe the link destination, since there is a lack of detail about what these links are a version of. When combined with the parent list item text of "Annual Report 2022" the needed context is provided.

Annual Report 2021 examples use link text of "View" which is not descriptive at all on its own. Combined with both the text in the same list item, and the parent list item, the user is provided with adequate context. **Content editors should be discouraged from authoring links in this manner, and you are encouraged to issue at least a warning whenever this is encountered.**

Annual Report 2020 has a link text of "archive" which is further detailed with the context in the same list item.

Appropriate Example - Link text combined with table cells and table headers describes the destination

In the following example, consider the first row and first column as table headers.

Year	Publish Date	Web Report	PDF Report	Word Report
2022	Feb 2, 2023	Read	Download (30MB)	Download (25MB)
2021	Feb 12, 2022	Read	Download (18MB)	Download (15MB)
2020	Jan 29, 2021	Read	Download (15MB)	Download (14MB)

If the table you are reviewing only has one row or column designated as a table header, then that is the only other cell that a table cell can rely on for context. If there are no table headers, then only the context within the same cell can be used to determine context.

Context of table headers may not be used if the table is marked as a layout or presentation table (where the `<table>` tag has `role=”presentation”` applied to it.)

Minimum Example - Link text combined in the same paragraph tag describes the destination

Read our 2022 Annual Report **here**.

This example utilizes “here” as link text. It would minimally pass WCAG 2’s requirement due to the context provided in the same paragraph, but it is highly discouraged. **It is recommended to mark a warning whenever this is encountered.**

Failing Example - Link text on its own does not describe the link destination.

Our financial information disclosure can be found in our 2022 Annual Report.

Read more here

2. Same link in the same context navigating to different locations

In scenarios when the link text plus context is fully equivalent, the two links must navigate to the same destination.

Fully equivalent link text plus context when comparing two links means that both the link text on its own is the same, and that any additional context the link may have is also the same on the same page. This does not necessarily mean that the context must come from the same source, if you had two links that said “About Us” in different places on the same page, and both links had no additional context, they would be considered two links with fully equivalent link text plus context.

59

Two or more mechanisms of finding a webpage are available, unless the page is accessed as part of a step in a process.

WCAG 2 criterion

2.4.5 Multiple Ways (Level AA)

Tools and requirements

- Manual Assessment

Test procedure

During manual evaluation, observe how many methods of navigating the site or web app exist. Any web pages that **aren't part of a sequence** (e.g. a confirmation page would only be reachable after completing a form) must be reachable by 2 different methods from the list below.

- Links in navigation areas and content areas (that is to say, any page would have a link to it from some other page on the site)
- Site Search
- Site Map
- Table of Contents
- A list of links to all pages on the site
- The ability to access all pages from links on the homepage

Part of a Sequence

Pages that would be considered part of a sequence is any specific page that you cannot access without first navigating to another page. Appropriate examples of pages that are part of a sequence include

- Form pages that are accessed one after another
- Forms and pages that are accessed after a CAPTCHA or similar test
- Confirmation pages after submitting a form

60

Headings that exist describe the topic or purpose of the content following after.

WCAG 2 criterion

2.4.6 Headings and Labels (Level AA)

Tools and requirements

- Manual Assessment
- WAVE, axe Accessibility Checker

Test procedure

During manual evaluation, observe all the headings present on the page. Ensure that the heading name correlates to the content that follows it until the next sibling heading. The heading name also needs to be relatively concise, and not overly wordy.

For example, if you had a heading 2 saying “News,” then all content and subheadings must relate to “News” until the next heading 2 is encountered.

Note that if a heading’s accessible name is descriptive (using aria-label or similar) but the visual heading is not, it is still a failure of this criterion.

This criterion is not applicable to pages where no headings are used at all.

61

Labels describe the purpose of the inputs they are associated with.

WCAG 2 criterion

2.4.6 Headings and Labels (Level AA)

Tools and requirements

- Manual Assessment

Test procedure

During manual evaluation of forms, observe all of the labels present on the page. Ensure that the label name is descriptive and properly describes the purpose of the input field.

Note that if a form field's accessible name is descriptive (using aria-label or similar) but the visual label is not, it is still a failure of this criterion.

62

Every focusable element has a focus indicator present.

WCAG 2 criterion

2.4.7 Focus Visible (Level A)

Tools and requirements

- Manual Assessment
- Keyboard

Test procedure

During manual evaluation of pages using keyboard navigation, observe and ensure that there is a visual focus indicator present.

A visual focus indicator for the purposes of this specific criterion is only concerned that there is *something* visible which clearly defines the element which has keyboard focus. This can take the form of an underline, outline, border, color shift, or background change (among others). The focus indicator for the purpose of this criterion does NOT have a color contrast requirement (that is applicable to 1.4.11 Non-text Contrast, or Guideline 39).

If you are unable to discern a focus indicator, then mark a failure. If you cannot quickly identify where you are on the page (e..g., the focus indicator is very faint or is clipped), then mark a failure.

63

Elements that currently have focus may never be hidden by other elements (such as a sticky header).

WCAG 2 criterion

2.4.11 Focus Not Obscured (Minimum) (Level AA)

Tools and requirements

- Manual Evaluation

Test procedure

For any focusable element on the page, check to ensure that at least some portion of the element's focus indicator is visible when focus is granted to it.

If focus is "visible" behind a semi-opaque overlay, it is not a failure of this particular criterion. Instead, in these situations, you would likely want to consider a failure against 2.4.3 Focus Order (#56) because focus should not leave an overlay without it dismissing or 1.4.11 Non-Text Contrast (#39) because the focus indicator is far less likely to have a 3:1 contrast ratio when hidden behind a semi-opaque background.

If focus is hidden behind an element that the user opened, it will only fail if the user can use a keyboard command to view the focused element without moving focus. (This includes using ESC to close an overlay, using arrow keys to scroll the page without moving focus, or using a command to toggle between two different views.)

This criterion is particularly important to check when there are sticky headers or modals present on the page. These elements are broken away from the normal flow of the page. As such, other page content will flow underneath these elements.

If a component is movable by the user, the visibility of the focus indicator is only evaluated against the initial placement of the movable component. (Dragging a window away from its initial position such that focus may become hidden will not constitute a failure.)

Specific situations that can be flagged under this criterion include

- A sticky header (or other sticky element) fully obscures the focus of an element like a link or button.
- A modal or slide-over (or similar) allows focus to escape the element without disappearing, causing elements to gain focus behind the modal/slide-over.

64

All functionality that uses multipoint, or path-based gestures can be operated with a single pointer without a path-based gesture, unless essential.

WCAG 2 criterion

2.5.1 Pointer Gestures (Level A)

Tools and requirements

- Manual Assessment
- Touchscreen Device

Test procedure

A **path-based gesture** is defined as functionality where an interaction relies on more than just a start and endpoint of a gesture (like a swipe or click and drag). Examples of path-based gestures include the following

- A swipe gesture where direction or speed is factored into the functionality
- A gesture that requires tracing a specific shape or path

A **multipoint gesture** is defined as any gesture which requires two points of contact to interact, like a two-finger pinch or a multi-point tap to interact.

Whenever either a path-based gesture or multipoint gesture is used then there must be an alternate method to interact with the functionality. For instance, a button that requires a simple tap or click, or a click and hold.

Path-based gestures and multipoint gestures are going to be relatively rare, you may encounter them on carousels or maps.

65

Functionality operated with a single pointer must: Not fire on the down event, fire on the up-event along with a way to abort or undo, reverse the function on the up-event, or completing the function on the down event is essential.

WCAG 2 criterion

2.5.2 Pointer Cancellation (Level A)

Tools and requirements

- Manual Assessment
- Mouse or Touchscreen interaction

Test procedure

When testing any functionality with a cursor or tap gesture, ensure that at least one of the following criteria is met.

- The function must not fire **at all** on the "down" event. It would fire only when the mouse click is released or a touch is released. This implies that functions like drag and drop, which start on the down event are not considered.
- A function begins on the down event but finishes on the up event, there needs to be a way to cancel or undo the function before completion. A drag-and-drop function would fall under this category.
- A function that starts on the down event is canceled on the up event.
- A function *must* fire on the down-event, such as when emulating keyboards or keypads. Interacting with most web functions like buttons, links, and form inputs are never cases that must fire on the down event.

66

For user interface components with labels that include text or images of text, the accessible name contains the text that is presented visually.

WCAG 2 criterion

2.5.3 Label in Name (Level A)

Tools and requirements

- Siteimprove

Test procedure

Siteimprove has a full test for this criterion, and it is very accurate. List any findings that Siteimprove flags as failures.

67

Any functionality which is activated by device motion can be performed with a User interface component that does not require motion.

WCAG 2 criterion

2.5.4 Motion Actuation (Level A)

Tools and requirements

- Manual Assessment
- Mobile device with gyroscope interaction

Test procedure

If there is any functionality where gyroscope motion is used to activate it, then there needs to be an alternate way of performing the functionality in a way where motion is not required.

Finding functionality using gyroscope motion is going to be quite rare. Most sites you encounter will likely not include any such functionality.

68

Any functionality which is activated by device motion can be disabled.

WCAG 2 criterion

2.5.4 Motion Actuation (Level A)

Tools and requirements

- Manual Assessment
- Mobile device with gyroscope interaction

Test procedure

If there is any functionality where gyroscope motion is used to activate it, then there must be a way to fully disable the motion portion of the function. It can take the form of a button or other control, and it should be clearly labeled.

Finding functionality using gyroscope motion is going to be quite rare. Most sites you encounter will likely not include any such functionality.

69

Any functionality that can be achieved by dragging (click and hold then move) must be operable without the need for dragging (unless essential).

WCAG 2 criterion

2.5.7 Dragging Movements (Level AA)

Tools and requirements

- Manual Assessment

Test procedure

This criterion does not apply to **essential** functionality.

Essential or out-of-scope functionality includes

- A written signature field (where you drag the cursor to write)
- A canvas where you have the option to draw freehand
- Scrollable areas
- Any scenario where two pointers are used (such as a pinch movement for zoom)

Identify if there is any non-essential functionality on the site that supports click and drag.

Some examples might include

- An interface where the user is presented with a large list of options alongside a list of selected items and moving items from the options list to the selection list (usually presented as two side-by-side columns).

- A map where you can move around by clicking and dragging the canvas.
- A color picker where you can select a specific color by dragging the picker.
- A slide control (such as for volume control).
- A function where you can drag files into a target area to upload.

Any identified examples must have an option to achieve the same goal without requiring a click and drag. It may be achieved using buttons or keyboard shortcuts.

70

All elements must have a clickable target size of at least 24x24 pixels unless the element is inline, controlled by the browser, or the "target offset" to all adjacent clickable elements is at least 24px.

WCAG 2 criterion

2.5.8 Target Size (Minimum) (Level AA)

Tools and requirements

- Manual Assessment (for now).
- Note: This can (and should) theoretically be automatically tested for, but no testing agents support this feature yet.

Test procedure

Note: This is a complex criterion. Please refer to the official WCAG 2.2 Documentation for details.
https://www.w3.org/WAI/WCAG22/Understanding/target-size-minimum.html

The ***minimum target size*** (the minimum size of the clickable area) of all interactive elements must be at least 24 x 24 pixels with **exceptions.**

Test in the following order, and if an element passes any of the tests (labeled as 1, 2, and 3 below), the element passes this success criterion. If the element fails *all* exceptions and requirements, the element fails the success criterion.

1. If the element's target area can fit a 24x24 square of pixels within it, it passes.

    a. Pass example 1 - The target elements are already at least 24x24px in size.

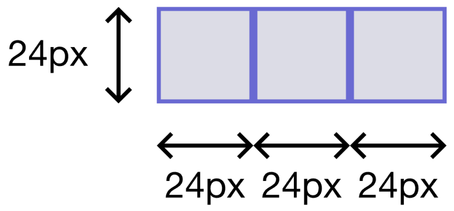


Figure 70.1 - All elements meet the minimum 24x24 size.

    b. Pass example 2 - A target area fits a 24x24px square wholly within it.

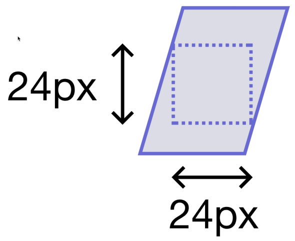


Figure 70.2 - A uniform 24x24 square that is completely contained within the target area.

2. ***Minimum spacing*** - if the element is sized such that you cannot wholly fit a 24x24 square within it, if you can draw a 24px diameter circle in the center of the element and **not intersect** another interactive element's minimum target size, it passes.

   The following examples are not exhaustive. **Read the WCAG 2.2 Documentation for all details.**
   https://www.w3.org/WAI/WCAG22/Understanding/target-size-minimum.html

    a. **W3C Example of pass and fail scenarios of adjacent icon elements.**

       Top-row elements pass because they are already 24x24 on their own.

       Middle-row elements pass because the 20x20px elements are spaced far enough apart that a 24px circle centered on the elements will not intersect another 24px circle on an adjacent element.

Bottom-row elements fail because the 20x20px elements are too close and a 24px circle centered on each element will intersect an adjacent element’s 24px circle.

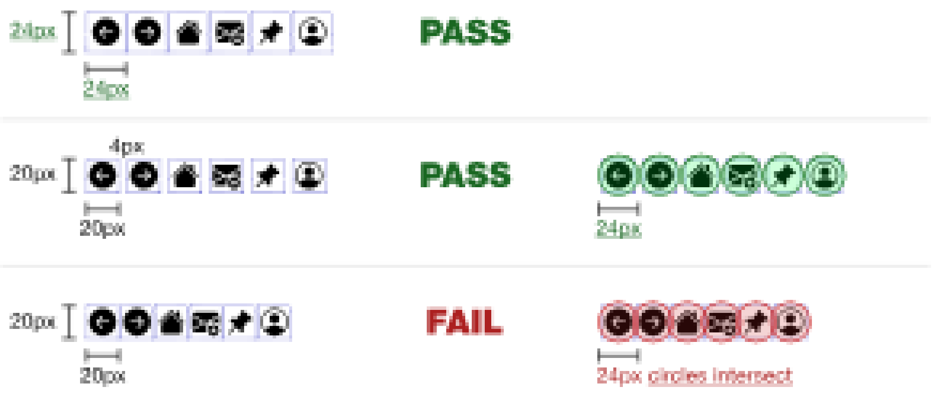


Figure 70.3 - Passing and Failing examples showcasing how to evaluate adjacent links using a circle’s diameter if the target links are smaller than the minimum 24x24px.

b. **W3C Example of passing and failing elements with adjacent buttons.**

The top row passes because there are no elements that intersect a centered 24px circle on each target.

The bottom row fails because elements DO intersect the centered 24px circle

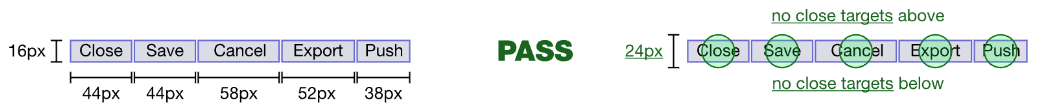
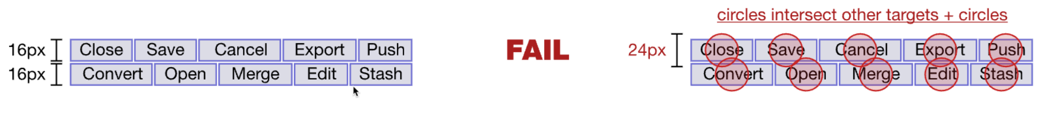


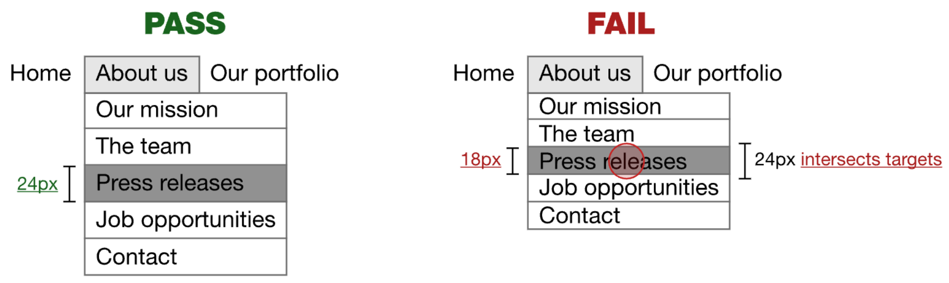
Figure 70.4 -



c. **W3C Example of passing and failing dropdown navigation.**

The passing option features a minimum 24px of height for each element.

The failing option has other navigation elements within the circle diameter.



3. If the target element fails both the **minimum target size** and **minimum spacing** checks - the following exceptions may also allow the element to pass.
   a. The functionality of the failing element can be achieved via an alternative control that passes.
   b. The element is inline (such as an inline link).
   c. The size of the target is fully controlled by the user agent and NOT user modified with CSS or otherwise.
   d. The presentation of an element is essential (the element MUST be presented the way it is, and there is no other way to perform the functionality).

71

The default language of each page must be defined in the html tag.

WCAG 2 criterion

3.1.1 Language of Page (Level A)

Tools and requirements

- Siteimprove
- Screen Reader

Test procedure

Siteimprove has a check for this requirement. It looks to ensure that the lang attribute is applied to the <html> tag at the start of the document. For most Cornell based sites, the lang attribute here should read "en" or "en-US" for American English.

If you do run across a site that is predominantly in a different language other than English, then the lang attribute needs to have the value which correlates to the predominant language of the page.

When testing pages in an alternate language than your default (probably English if you are reading this), you need to ensure that your screen reader is set up to have voices enabled for the language you are testing. On Mac, you can set this up in VoiceOver by going to your VoiceOver Utility, select "Speech" and click the "Voices" tab. From there, press the plus (+) button to assign voices to different languages.

An easy-to-read reference to language codes is provided by W3Schools (https://www.w3schools.com/tags/ref_language_codes.asp)

72

Text in a different language than the page default must be identified in HTML; aside from proper names, technical terms, words without a defined language, and words that are part of the vernacular of the immediately surrounding text.

WCAG 2 criterion

3.1.2 Language of Parts (Level AA)

Tools and requirements

- Manual Evaluation

Test procedure

Anytime a word is encountered which is not part of the page language (usually "English" for Cornell sites) needs to have its language declared independently of the rest of the page. (**Except** proper nouns, technical terms, scientific terms, words without a language, or words that are part of the vernacular of the nearby text).

When testing pages in an alternate language than your default (probably English if you are reading this), you need to ensure that your screen reader is set up to have voices enabled for the language you are testing. On Mac, you can set this up in VoiceOver by going to your VoiceOver Utility, select "Speech" and click the "Voices" tab. From there, press the plus (+) button to assign voices to different languages.

This is done in HTML using the lang attribute within the idiomatic text element <i>.. For instance, if the French phrase "s'il-vous-plait" was used in text in a predominantly English language page, you would want to see an `<i>` tag with the lang attribute applied to it.

```
<p>Carefully read all of the instructions <i
lang=”fr”>s'il-vous-plait</i></p>
```

73

When a user interface component gains focus, it may not trigger a change of context. (Submission of a form, new window launch, immediate change of focus, changing content on the page where such that the purpose is different)

WCAG 2 criterion

3.2.1 On Focus (Level A)

Tools and requirements

- Manual Evaluation
- Keyboard

Test procedure

During manual evaluation, observe the interactions that occur when moving focus to new elements. The observation for this criterion is any situation when adding or removing focus from an element causes a major change of context.

A change of context is considered any situation where the user's focus or window changes such that the user no longer can reasonably understand where they are in relation to where they just were. Examples of changes of context include

- Auto-submitting a form on focus (as opposed to clicking on a button or hitting the Return key).
- Opening a new window or modal on focus.

- Changing the contents of the page, such that the purpose of the content is altered. (It is okay to say close out of a popup in a navigation on a focus change, but not to dramatically change the page content on focus.)

74

When the setting or value of a User Interface component is changed, that does not trigger a change of context unless the user is informed of the behavior beforehand.

WCAG 2 criterion

3.2.2 On Input (Level A)

Tools and requirements

- Manual Evaluation
- Keyboard

Test procedure

This criterion evaluates when you change the value of a checkbox, radio button, select box, or any other user interface component which offers different values or states, there is no change of context *unless* there is an instruction in text *before* the interaction stating that changing values will direct you to a different location. This type of instruction may NOT be screen reader only (invisible to users except those using screen readers).

Common scenarios when a user interface component changing values changes context include

- Filters for large chunks of content.
- Changing the value of a select box which immediately navigates the user elsewhere.
- Auto-submission of a form.

If the change of context is triggered by a button, then that interaction is not applicable to this criterion.

75

Navigations which are utilized on multiple pages keep the same relative order on all pages, unless the user initiates the change. (Items can be removed or added, but they must maintain the same order relative to each other)

WCAG 2 criterion

3.2.3 Consistent Navigation (Level AA)

Tools and requirements

- Manual Evaluation

Test procedure

When observing navigations which appear on multiple pages, ensure that the relative order of the navigation items remains in the same sequence compared to other navigation elements.

It is okay (albeit not recommended) to have the contents of navigation areas change depending on the page. This criterion is only concerned that whatever navigation elements are visible on whatever page, they are in the same relative order that they appear in on other pages.

For example, a main navigation structured as so on the homepage...

- Homepage

- About
- Projects
- Contact

…compared to that same main navigation on an “About” subpage (where the “About” link is replaced by other links)…

- Homepage
- Leadership
- People
- Projects
- Contact

…would be considered a passing implementation because the relative order of the repeating “Homepage,” “Projects,” and “Contact” links are the same.

76

Any components with similar functionality used on multiple pages must be labeled identically and function identically. (e.g. a header Search field must be labeled the same on all pages)

WCAG 2 criterion

3.2.4 Consistent Identification (Level AA)

Tools and requirements

- Manual Evaluation
- Screen Reader

Test procedure

During manual evaluation, identify elements that repeat on multiple pages. Any element such as a search bar or a navigation must have the same accessible name and functionality on all pages it appears on.

Elements repeating on multiple pages are usually found in the header and footer, but can also be in sidebars. Multiple pages in the context of this criterion is not limited to only elements that appear on every page. If an element repeats itself on all “News” pages for example, it would also need to be identified the same on all the pages it appears on.

77

Help mechanisms such as contact details, messaging, chat, or self-help options must be in the same relative order on all pages where the information is present.

WCAG 2 criterion

3.2.6 Consistent Help (Level A)

Tools and requirements

- Manual Evaluation

Test procedure

Help mechanisms are a broad scope of mechanisms. These must be in the same relative order on every page wherever the mechanism is repeated. **It is not required to provide these mechanisms, only to put them in the same relative order if they do exist.**

Help mechanisms in scope include

- Human contact information (phone number, fax number, email addresses, hours of operation, physical address)
- Direct human contact mechanisms (messaging client, live chat, contact forms, social media links)
- Self-help mechanisms (FAQs, Support information)
- Automated help mechanisms (chat bots)

78

Whenever an input error is detected through validation, the user is informed of the error, and what was incorrect in the input.

WCAG 2 criterion

3.3.1 Error Identification (Level A)

Tools and requirements

- Manual Evaluation
- Screen Reader

Test procedure

During manual evaluation, take note of any forms that include validation. Validation is any functionality where the site checks input data to ensure that it is filled out correctly. Examples include verifying that a required field has data filled out in it or For this criterion, you are looking to ensure that there is an alert whenever there is an invalid input (such as leaving a required field blank, or not using a proper email format).

The notification that an error has occurred must be **immediately** perceived by users using any assistive technology (such as a screen reader). Furthermore, the notification must inform the user of both the field that was incorrectly input, and the reason it was incorrectly input.

There are examples below indicating valid methods of Error Identification. They are split into “client-side” and “server-side” validation categories. Client-side validation is any form of validation where the form does NOT submit and immediately informs the user of the errors without a page refresh. Server-side validation does submit the form, but refreshes the page and displays information about what fields were incorrectly filled out.

Client-side validation

- Upon hitting the “Submit” button, an alert dialog pops up, detailing all the fields which were incorrectly filled out. The incorrectly filled out fields have the **label** change to identify the error. A link is provided after the submit button (or the user is redirected to a spot where links are provided) to each invalid input.
- Upon hitting the “Submit” button, the user’s focus is directed to text stating which fields are invalid. The invalid fields are identified visually and in HTML using aria-invalid.
- During the course of filling out the form, if a required input or input with invalid data loses focus, the form automatically changes the label of the form field to indicate that there was an error.

Server-side validation

- After being redirected, the invalid form fields are identified at the top of the page. The individual form fields which were incorrectly filled out have their labels changed to identify the error.
- After being redirected, the invalid form fields are identified at the top of the page, and each incorrectly filled out form field has aria-invalid applied to it.

79

Visible labels or instructions are available for all inputs and input groupings.

WCAG 2 criterion

3.3.2 Labels or Instructions (Level A)

Tools and requirements

- Manual Evaluation

Test procedure

During manual evaluation, on all forms, ensure that each input has a descriptive label which describes its purpose. Fieldsets also require labels if they are required to understand the relationship of multiple inputs.

80

Labels describe any required fields or required formatting requirements. (e.g. If a MM/DD/YYYY format is required)

WCAG 2 criterion

3.3.2 Labels or Instructions (Level A)

Tools and requirements

- Manual Evaluation

Test procedure

During manual evaluation, if a form field has a required format, that format must be detailed in the label. Specific formatting details are not needed for fields where a format is implied (e.g. an “Email” field would expect an ‘@’ in it with a URL at the end).

Common fields that may have formatting requirements that need to be explained in the label include

- Dates
- Needing NetID@cornell.edu or just NetID
- Phone Number

81

If an input error was detected due to a blank input on a required field, the user is informed of the fields which were left blank in the error message.

WCAG 2 criterion

3.3.3 Error Suggestion (Level AA)

Tools and requirements

- Manual Evaluation

Test procedure

During manual evaluation, when testing webforms, leave the entire form blank and submit. Any required fields that were left blank need to be detailed within the error message that is received.

(For details on valid error messages, refer to Guideline 78: Whenever an input error is detected through validation, the user is informed of the error, and what was incorrect in the input.)

82

If an input error was detected due to an input that did not follow required formatting, the user is informed of the proper formatting in the error message.

WCAG 2 criterion

3.3.3 Error Suggestion (Level AA)

Tools and requirements

- Manual Evaluation

Test procedure

During manual evaluation, when testing any form inputs that have formatting requirements (like an email address space), purposely fill out the field improperly (for instance, leave the '@' symbol out of the email field). The error message for any field that has formatting requirements must describe how to properly format the data.

Inputs that may have formatting requirements may include

- Dates (mm/dd/yyyy, mm/dd/yy etc.)
- Email
- Needing NetID@cornell.edu or just NetID
- Phone Number

(For details on valid error messages, refer to Guideline 78: Whenever an input error is detected through validation, the user is informed of the error, and what was incorrect in the input.)

83

If an input error was detected that was outside of the allowable range of values, the user is informed of the proper range in the error message.

WCAG 2 criterion

3.3.3 Error Suggestion (Level AA)

Tools and requirements

- Manual Evaluation

Test procedure

During manual evaluation, when testing form inputs that have limits on acceptable values (if any), purposely fill out the field improperly. The error message for any field with a limited range of acceptable values should describe how to correctly fill out the field.

Inputs that have limits on acceptable values may include

- Dates (only after x date and before y date)
- Quantities (such as for tickets or spaces to reserve for a conference)
- Comboboxes

84

Any legal or financial data input must be reversible, validated for input errors, or include a mechanism for reviewing, confirming, or correcting information before submission.

WCAG 2 criterion

3.3.4 Error Prevention (Legal, Financial, Data) (Level AA)

Tools and requirements

- Manual Evaluation

Test procedure

This criterion is specific to forms that request payment. If one of these forms is encountered, attempt to fill out as much of the form as possible. **Do not submit actual payment information** unless you are provided sample data and are instructed to use it as part of the site or product you are evaluating..

Any sort of form that accepts legal or financial data must meet **one** of the following requirements:

- The form submission is reversible. There exists an accessible method of canceling or editing the submission so that any errors can be corrected.
- The form submission is checked and verified for accuracy. Upon submission, there is a confirmation screen that is presented to the user before full submission which includes a review of all of the form fields that were filled out.
- The form submission has a method of reviewing, confirming, or correcting information before submission. This can be a checkbox that must be ticked

affirming that the user has checked their inputs and is fully confident. It could also mean that the form offers up a way of going back to a previous spot in the form to edit responses that the user did not intend to input.

85

Users are not required to refill the same information in the same process unless it is available as a selection, is essential, is ensuring security, or the original information is no longer valid.

WCAG 2 criterion

3.3.7 Redundant Entry (Level A)

Tools and requirements

- Manual Evaluation

Test procedure

This criterion applies to forms where one "**process**" is being achieved. This means that it will apply to a singular webform or a series of webforms that perform the same ultimate objective. For example, a form asking for information when signing up for a newsletter will not have any influence on a form asking for information about registering for a class. However, a series of forms (perhaps a 2 or 3 step form on 2 or 3 different pages) asking for information about registering for a class would be treated as one process, and each form in the process would apply to this rule.

A process will "reset" if a user were to leave the session and return – there is no requirement for information previously entered in a canceled process to be preserved or made available to the user. However, a single process may involve multiple domains and not have the requirements of this success criterion "reset." (notably if a user is temporarily sent off-site for payment processing, and returned afterward).

Within a single process, the user can not be prompted for the same information more than once. (e.g., a process may not ask for the user's email address more than once.) If a process does ask for the same information more than once, that information must be automatically filled in by the website or be "available to select". (Available to select in this context means that there has to be a mechanism such as having the option available in a drop-down, having a checkbox that will auto-populate information, or having text on the page that can be copied into the form. All of these must be on the same page.)

There are **important limited exceptions** to the rule. These include

- Asking for a password more than once for security purposes.
- Preventing a password field from being copyable (for a "re-enter your password" field for example) for security purposes.
- Asking for the same information again if the previous information is no longer valid.

86

Login processes must not solely rely on cognitive tests. All steps in a login process must support some method that does not rely on memory or knowledge.

WCAG 2 criterion

3.3.8 Accessible Authentication (Minimum) (Level AA)

Tools and requirements

- Manual Evaluation

Test procedure

Logins must support some method of entry that does not solely rely on memory. Most often, you will encounter logins that are password/username combos. This criterion is not saying those are forbidden, rather that there must be some way of providing authentication that doesn't require the user to memorize anything.

This criterion also applies to two-factor authentication/multi-factor authentication (2FA/MFA). Any 2FA/MFA solution must also meet this criterion. (Duo Mobile and Beyond Identity - the Cornell 2FA mechanisms as of 2023 - support this criterion)

CAPTCHA that is object recognition based is allowed (this does NOT mean identify the scrambled text, an object is something tangible in this context). CAPTCHA that relies on solving a math problem or a puzzle is not.

This can be achieved by ensuring at least one of the following is supported.

- Allowing browsers to automatically insert passwords and usernames for the user.
- Allowing users to paste passwords and usernames in their respective fields.

- Allowing password managers (like LastPass) to insert username/password information.
- Using a non-password based authentication system (such as a biometric)
- Allowing a third-party provider to sign-in on behalf of the user
- Using a QR code scan to sign in.

87

All elements have appropriate accessible names and roles assigned.

WCAG 2 criterion

4.1.2 Name, Role, Value (Level A)

Tools and requirements

- Screen Reader
- Manual Assessment

Test procedure

During manual evaluation using a screen reader/ANDI, observe all of the elements on the page and ensure that every element has an accessible name and role which accurately describes its purpose.

A list of things to be on the lookout for include

- Elements that behave like links (elements that take you to a different place) must be announced as links.
- Elements that behave like buttons (elements that trigger functionality) must be announced as buttons.
- Elements organized as a table must announce as a table
- Related content items displayed together in a group (with or without bullets) must announce as a list (ordered or unordered depending on the context)
- Any text input must be announced as an input.

"Announced as" in this case refers to how a screen reader will say for example "Link" or "Button" when the user navigates to them.

88

Iframes which present user-readable content require a title or description.

WCAG 2 criterion

4.1.2 Name, Role, Value (Level A)

Tools and requirements

- Siteimprove helps (Policy)

Test procedure

Identify all `<iframe>` content on the page. There is a policy set up in Siteimprove to help with this called “Find iframes.”

- If the `<iframe>` contains readable or viewable content (i.e., is not a container for analytics scripts that the user won’t interact with), the `<iframe>` must have a `title` attribute with a short description of the contents.
    - If the content is video or audio content, the title should represent the title of the media.
    - Otherwise, the title should represent the embedded content. (The page title of whatever is embedded for example)

89

All custom functionality utilizes the appropriate ARIA features for states, properties, and values. (e.g. aria-expanded, aria-haspopup)

WCAG 2 criterion

4.1.2 Name, Role, Value (Level A)

Tools and requirements

- Screen Reader
- Manual Assessment

Test procedure

During manual evaluation, observe the interactions of all elements and functionality on the page using a screen reader. Each of the following types of interactions must be announced as follows.

“Custom functionality” for the purposes of this criterion is focusing on non-native HTML functions, like accordions, modals, and dropdown menus. Not buttons, links, or form fields which have native HTML tags.

- Accordion
    - Announced as “Button” on the element which triggers the accordion. (<button> **OR** role=”button”)
    - The accordion state must be announced as expanded or collapsed (aria-expanded=”true/false”)
- Comboboxes

    - Announced as “Combobox” on the element which triggers the combobox open (role=”combobox”)
    - The state of the combobox must always be announced as either expanded or collapsed (aria-expanded=”true/false”)
- Tablist
    - Announced as “Tablist” on the grouping of tabs (role=”tablist”)
    - If the tab list has a visible label, set aria-labelledby to the id of the labeling element. If not, use aria-label to provide a label.
    - Each individual tab is announced as “Tab” (role=”tab”)
    - The selected tab must announce as selected (aria-selected=”true”)
    - Each panel which contains individual tab content is announced as “tabpanel” (role=”tabpanel)
    - Each tabpanel which is not visible must be hidden from the screen reader (aria-hidden=”true” and/or CSS display: none;)
- Navigation (including Pagination)
    - Announced as “list”
    - Any parent navigation item with a drop down must announce expandable content (aria-haspopup=”true”)
    - Any triggers to open drop downs must announce as “button”
    - If there is a visual indicator of the current item in the navigation list, then the element must announce that the user is on the current item (aria-current=”page”)

If the item you are currently assessing is not listed above, refer to the W3C WAI-ARIA 1.1 reference for details on how it should properly work. (https://www.w3.org/TR/wai-aria-practices-1.1/#aria_ex)

90

Status messages which change without a page reload notify users of assistive technologies without requiring the user to send focus to the message.

WCAG 2 criterion

4.1.3 Status Messages (Level AA)

Tools and requirements

- Screen Reader
- Manual Assessment

Test procedure

A **status message** is any bit of information which informs the user about a state change on the page that does not require a page refresh.

A popular example is a results page for a search if filters are applied. If a message saying “1234 results found” automatically updated without refreshing the entire page (i.e. it was implemented with AJAX), then that sort of status message would apply to this criterion.

The expected behavior is that when one of these messages appears or changes, a screen reader user and other assistive technology users must be immediately notified of the change. It could be as disruptive as a Javascript alert, but more commonly, aria-live is utilized to immediately announce the change to the user.

When using a screen reader, if you come across any sort of status message, interact with the functionality on the page to change that status message, and verify that when the message changes, the screen reader immediately reads out the change to the user, regardless of where the user is on the page.

Best Practice

Visually grouped blocks of content should be organized by appropriate landmarks.

WCAG 2 criterion

1.3.1 Info and Relationships (Level A)

Tools and requirements

- WAVE

Test procedure

For all pages on the site, observe how content is organized visually. Groupings of content should be organized together in appropriate landmarks. It is not a failure of WCAG if this is not the case.

Landmark elements to focus on include the following

- **Banner** - All of the content which pertains to the site as a whole. (the header content of the site)
    - Usually includes the site logo, site searches, site navigation.
    - In HTML5, a <header> tag automatically identifies a banner landmark.
    - Using ARIA, role=”banner” identifies a banner landmark.
- **Complementary** - Any content which supports the *main content* but is relevant as its own section of content.
    - Examples include a list of related articles or a section navigation.
    - In HTML5, an <aside> tag automatically identifies a complementary landmark.

    - Using ARIA, role=”complementary” identifies a complementary landmark.
- **Contentinfo** - All of the content pertaining to the site at the bottom of the page (the footer of the site)
    - Often includes quick links, copyright statements, accessibility statements, and contact information.
    - In HTML5, a <footer> tag automatically identifies a contentinfo landmark.
    - Using ARIA, role=”contentinfo” identifies a contentinfo landmark.
- **Main** - The main content of the page.
    - The main content should include any content exclusive to the page the user is on. It should avoid including content repeated on more than one page (aside content, banner content, footer content etc..). We will not consider it a strict failure if not abided to.
    - In HTML5, a <main> tag automatically identifies a main landmark
    - Using ARIA, role=”main” identifies a main landmark.
- **Navigation** - Any grouping of links used to navigate to different parts of the page or site.
    - In HTML5, a <nav> tag automatically identifies a navigation landmark.
    - Using ARIA role=”navigation” identifies a navigation landmark.
- **Region** - A distinct section of the content which is important enough that a user may want to directly navigate to it
    - Examples include blocks of Upcoming events, Social Media feeds, or Featured Articles
    - In HTML5, a <section> tag automatically identifies a region landmark.
    - Using ARIA, role=”region” identifies a region landmark.
- **Search** - Any search form
    - There is no native HTML5 tag to identify a search tag. You must use ARIA
    - Using ARIA, role=”search” identifies a search landmark.

Best Practice

Landmark types which are used more than once on a page should be distinguishable from each other using ARIA.

WCAG 2 criterion

1.3.1 Info and Relationships (Level A)

Tools and requirements

- WAVE
- Siteimprove

Test procedure

Use WAVE to get a list of all of the landmarks on a specific page. You should repeat this procedure on each unique page type on the site or application.

Landmark types which duplicate more than once should be uniquely labeled. This is generally achieved with aria-label or aria-labelledby. Use the Inspector of your web browser to determine if landmarks have labels.

As a reminder, landmark elements are defined as

- **Banner** - <header> tag or an attribute of role="banner" on a container element.
- **Complementary** - <aside> tag or an attribute of role="complementary" on a container element.
- **Contentinfo** - <footer> tag or an attribute of role="contentinfo" on a container element.

- **Form** - <form> tag **with** an ARIA label or an attribute of role=”form” on a container element.
- **Main** - <main> tag or an attribute of role=”main” on a container element.
- **Navigation** - <nav> tag or an attribute of role=”navigation” on a container element.
- **Region** - <section> tag **with** an ARIA label or an attribute of role=”region” on a container element **with** an ARIA label.
- **Search** - An attribute of role=”search” on a container element.