THE BUSINESS LEADER'S GUIDE TO ACCESSIBILITY

Why it matters, what's at stake, and the steps to take to create a fully accessible digital product

A comprehensive handbook for executives, marketers, designers, engineers, and digital strategists

mindgrub

ACCESSIBILITY DEFINED

Accessibility ensures that persons with a disability can access commodities in the same way as a person without a disability.

A disability is an impairment that affects how a person interacts with the world around them. These impairments can affect cognitive, visual, auditory, or motor skills, or any combination of the four.

The Americans with Disabilities Act of 1990 (ADA) prohibits discrimination against persons with a disability or disabilities. The ADA often overlaps with the Rehabilitation Act of 1973, which prohibits discrimination on the basis of disability in programs conducted by federal agencies, including financial assistance, employment, and employment practices. Within the Rehabilitation Act of 1973 is Section 508 which requires government agencies to make their electronic information accessible to people with disabilities.

While Section 508 does not apply to the private sector, it's still important that your organization consider the need for a web-accessible application.

Software accessibility provides individuals with impairments and/or disabilities the opportunity to access an application with ease. In fact, software accessibility provides a better user experience for **everyone**. The criteria for an accessible site overlap with best practices across several domains, including design, usability, and search engine optimization (SEO).

The web is meant to work for all people.

The Importance of Accessibility

Today, inaccessible websites exclude **15% of the world's population,**[1] and while the lack of accessibility across all digital platforms is problematic from a human perspective, it's also a tremendous business risk.

If your website isn't accessible, your business is missing out on revenue - the estimated global spending power of people with disabilities is **$6.9 trillion.**[2]

Among consumers with a disability, **82%** would opt to purchase from an accessible website over an inaccessible one.[3]

Companies ignoring accessibility standards are also at risk of an accessibility lawsuit, the number of which **tripled from 2017 to 2018.**[4]

We created this in-depth guide to the Web Content Accessibility Guidelines (WCAG) 2.1 A and AA success criteria, with the goal that everyone can develop and design their applications with accessibility in mind.

How to Use This Guide

We like to think of this guide as required reading for everyone who works in the digital world.

It's equal parts *intro into the world of accessibility* and *actionable guide to building a fully compliant product*. Whether you're the Director of Marketing managing your companies website rebuild project, the designer sketching out wireframes, or the developer responsible for coding and creating the site itself, this guide is for you.

We know building web products is a collaborative process, so we wrote this book will the whole team in mind.

In these pages, we'll explain the criteria needed to meet both A (the lowest level) and AA (intermediate) compliance. We've even included table of contents that doubles as a printable checklist, so you can mark off criteria as you go.

So treat this guide like a mini-textbook. Read it in bits and pieces or all at once, from beginning to end or by jumping from section to section. Hang on to it. Refer back to it. Share it with anyone and everyone, and help make the web a better, more accessible place for all.

1 World Health Organization
2 Forbes
3 The Click-Away Pound Report 2016
4 Bureau of Internet Accessibility

Terminology

To avoid jargon-induced confusion, we've included the glossary at the beginning. Before you dive into the criteria sections, take a moment to review these common accessibility-related terms.

A
The minimum level of conformance for accessibility. For Level A conformance, the application satisfies all Level A Success Criteria, or a conforming alternate version is provided.

AA
The second level of conformance for accessibility. For Level AA conformance, the application satisfies all Level A *and* Level AA Success Criteria, or a Level AA conforming alternate version is provided.

AAA
The third and highest level of conformance for accessibility. For AAA conformance, the application satisfies all Level A, Level AA, and Level AAA criteria. Due to the customized nature of achieving AAA conformance, AAA criteria are not included in this guide. For more information, see *Going beyond A and AA* in the Conclusion section.

ARIA
Accessible Rich Internet Applications (ARIA) is a set of attributes that define ways to make web content and web applications (especially those developed with JavaScript) more accessible to people with disabilities.[5]

ASSISTIVE TECHNOLOGY
Technologies or software that assist individuals with disabilities or impairments in accessing content. Examples include screen readers, screen magnifiers, and speech-input software.

COMPLIANCE / COMPLIANT
Compliance, specifically, "508 Compliance" is the general term used to describe the state of the entire application from an accessibility standpoint. A compliant application adheres to the governmental standards described in Section 508 of the Rehabilitation Act of 1973.

CONFORMANCE
In order to meet the needs of different groups and different situations, three levels of conformance are defined: A (lowest), AA, and AAA (highest). Conformance can be met at the individual level (success criteria) and macro-level (full application). For a site to be fully conformant, all parts of the application must meet the criteria established for that level.

EQUIVALENCE
Individuals with disabilities and/or impairments may not be able to receive information in the same way that an individual without an impairment would. When that is the case, the information can be delivered in a different way but must be equivalent, meaning nothing is omitted.

FAILURE
An aspect of the application does not conform to WCAG success criteria.

SUCCESS
An aspect of the application conforms to WCAG success criteria.

SUCCESS CRITERIA / CRITERION
Criteria refer to the collective standards, while criterion refers to an individual standard, by which an aspect of the application passes or fails for accessibility.

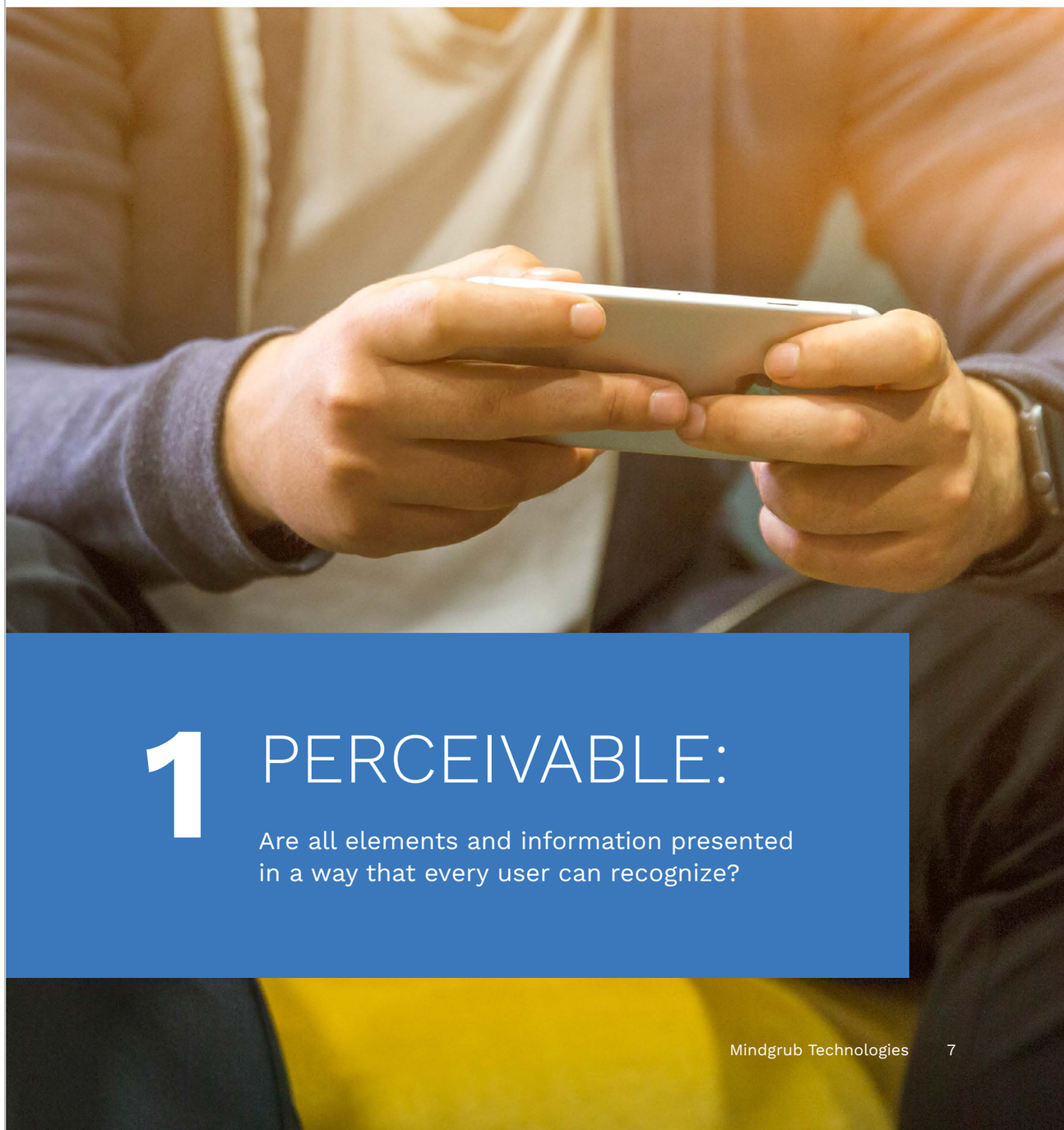
WCAG
WCAG stands for Web Content Accessibility Guidelines, which was created by the World Wide Web Consortium (W3C). These are the standards software applications must meet in order to be accessible to individuals with disabilities. This guide uses criteria from WCAG 2.1, the most current edition of WCAG at this time.

5 ARIA

TABLE OF CONTENTS | Standard Compliance

This table of contents doubles as a checklist. Use it to navigate this guide or print it out and check off criteria on your journey to compliance.

CRITERIA FOR
Standard Compliance (A and AA)



1 PERCEIVABLE:

Are all elements and information presented in a way that every user can recognize?

1.1 & 1.2 Section Preview:
Text Alternatives & Time-Based Media

The following criteria provide text alternatives for non-text content and time-based media.

- 1.1.1 Non-text Content (A)
- 1.2.1 Audio-only and Video-only (Prerecorded) (A)
- 1.2.2 Captions (Prerecorded)
- 1.2.3 Audio Description or Media Alternative (A)
- 1.2.4 Captions (Live) (AA)
- 1.2.5 Audio Description (Prerecorded) (AA)

1.1.1
Non-text Context (A)

This criterion is specifically for users with visual impairments who use a screen reader. It states that all non-text content (images, charts, diagrams, or animations) should have alternative text, commonly referred to as “alt text”.

Alt text provides a textual description of the visual element so that when a user accesses the element with a screen reader, they are able to receive equivalent information.

To the right is an image of the Mindgrub logo.

In order for this success criteria to pass, a screen reader would have to announce “Mindgrub logo” when accessed. If the element was only announced as “Mindgrub” or “image”, then it would fail to meet this success criterion.

(Mindgrub, 2019)

TO MEET THIS CRITERION:

- Ensure <alt> tag is provided on all images
- To test: use a screen-reader and check that when the content is accessed, it announces:
  1. What the non-text content is
  2. A detailed description of the non-text content

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

This criterion is all about equivalency so that all users may access the media. The type of alternative that should be provided is dependent on the type of media.

TO MEET THIS CRITERION:

Provide an alternative way to access recorded media.

- For audio-only media, provide an exact transcript so that a user with a hearing impairment can access the media.
- For video-only (without audio, such as an animation), a transcript *or* an audio description may be provided. This allows a user with a visual impairment to access the media. Additionally, this is beneficial for users with cognitive impairment(s) who may not be able to understand what is happening in the video.

1.2.2
Captions (Prerecorded) (A)

This criterion is specifically for users who have a hearing impairment. It states that captions should be provided for prerecorded audio and video.

TO MEET THIS CRITERION:

- Include captions that are *exactly* equivalent to the audio. You’ll want to include who is speaking, plus any sound effects or noises. Be sure to include all your **claps**, **whistles**, and **stomps** because recording each noise is vital to the success of this criterion.

Simply put, all content and information must be structured in a way that can be rendered programmatically or textually.

1.2.3
Audio Description or Media Alternative (A)

Someone who is blind or has low vision is still able to experience certain aspects of videos because of the audio. However, there are certain aspects of a video that cannot be translated through audio, such as actions, characters, scene changes, and on-screen text that is not spoken. These aspects of the presentation are important, but not a part of the main soundtrack. Thus, to ensure equivalency, it is important to provide a means for users with visual impairments to get this information.

TO MEET THIS CRITERION:

- Provide an audio description to augment the main soundtrack of the video, **or**
- Provide a textual description of the entire media presentation

1.2.4
Captions (Live) (AA)

Captions are required for prerecorded video and audio (per criterion 1.2.2), but they're also required for live video and audio. When there is real-time video or audio, captions must be synchronized with the media. You'll likely see this occur for breaking news.

TO MEET THIS CRITERION:

- Captions may be generated with a real-time translation tool or an on-site transcriber.
- Captions must be synchronized with the content.

1.2.5
Audio Description (Prerecorded) (AA)

Audio descriptions are beneficial for users with visual impairments because it provides an audible version of the video. It's important to note that the audio description has to be synchronized with the video. This means that if there is a pot banging in the background, then the audio description should denote that a pot is banging in the background.

This ensures that users who rely on their hearing to experience what they otherwise would experience visually can receive information in an equivalent manner.

TO MEET THIS CRITERION:

- All stage directions, scenery, and video context that are not verbally announced must be stated in the audio description.

1.3 Section Preview: Adaptable

These criteria ensure that content can be viewed in different ways, without losing any information.

- 1.3.1 Info and Relationships (A)
- 1.3.2 Meaningful Sequence (A)
- 1.3.3 Sensory Characteristics (A)
- 1.3.4 Orientation (AA)
- 1.3.5 Identify Input Purpose (AA)

1.3.1
Info and Relationships (A)

This is a meaty criterion. Per the WCAG, *the intent of this Success Criterion is to ensure that information and relationships that are implied by visual or auditory formatting are preserved when the presentation format changes.*

Simply put, all content and information must be structured in a way that can be rendered programmatically or textually.

This criterion is critical to achieving accessibility because its satisfaction benefits keyboard users, screen reader users, *and* individuals with cognitive impairments. For example, a user without visual impairment can distinguish headings, bold-faced or italicized text, new paragraphs, and more - all of which allow the user to perceive relationships and structure.

A user with a visual or auditory impairment may not perceive such relationships and structure if the formatting is not programmatically determined. If the technology does not permit programmatically determining relationships, a text description must be provided.

TO MEET THIS CRITERION:

- Appropriately tag:
  - Headings
  - Paragraphs
  - Fieldsets
  - Lists
  - Tables
    - Accurately tagged columns, rows, headers
    - No empty cells
    - Captioned tables
  - Emphasized text (**bold**, *italicized*, <u>underlined</u>)

IMPORTANT TO NOTE:

- Someone using a screen reader will use the screen reader's built-in features to access content quickly and efficiently. They may navigate the application by accessing headings so that they are aware of the sections within the web page. In this instance, having all headings tagged as <H1> will be vital for screen reader users to receive information in a hierarchical, organized way.

- Additionally, someone who has a cognitive impairment that affects how they process stimuli (for instance, someone prone to experiencing sensory overload), will benefit from having appropriately and consistently styled headings so that information is not overwhelming and can be processed in digestible pieces.

1.3.2
Meaningful Sequence (A)

Meaningful sequence refers to how the application is designed and created. Successful design and development processes consider how assistive technologies, specifically screen readers, access and programmatically render information.

In other words, if a person is using a screen reader to navigate your site, you want to make sure the information is read to them in the correct order.

Screen reader users use certain keystrokes in conjunction with their assistive technology to access information. Some of these keystrokes include using the tab key to announce interactive elements and the reading keys to read content on the page.

Typically, when a screen reader user first lands on an application, they will use the reading keys to hear what is on the screen. This includes any textual content and interactive elements in the order that they appear visually.

Imagine if the navigation was at the bottom of the page? A person interacting with your site using the keyboard would have to navigate to the bottom of the page, using the tab key, each time they want to interact with a different area of the application.

If your content isn't arranged in a meaningful sequence, the user - whether he or she has an impairment or not - is likely to be confused and disoriented. This is a great example of how designing for accessibility benefits everyone!

TO MEET THIS CRITERION:

- Ensure that it is possible to programmatically determine the correct content sequence so that a screen reader will read the content in an order that makes sense. There may be more than one correct order, but only one is needed to satisfy this criterion.

1.3.3
Sensory Characteristics (A)

Often times applications contain symbols and shapes that a person without an impairment understands intuitively, such as an arrow directing attention to a certain area. However, someone with a cognitive impairment may not understand what the symbol means and someone using a screen reader may need more information to understand the context of the symbol.

It is important that descriptive instruction is provided to accompany the symbol. We cannot rely solely on the symbol's shape, color, size, visual location, orientation, or sound.

The images below are examples of unsuccessful and successful implementations of sensory characteristics.

The first image has two blank blue arrows that represent navigating forward or backward in the application. However, these symbols have no labels or instructions detailing how to interact with them.

The second image has the same blue arrows, but with text detailing the purpose of the arrows. Additionally, below the symbols is a textual instruction describing in further detail how to interact with the functionality. This is important for individuals with visual and cognitive impairments, as it allows them to quickly and efficiently interact with user interface elements - without having to guess their function.

Unsuccessful:

Successful:

(National Enterprise Architecture)

TO MEET THIS CRITERION:

- It is best to provide a label for the symbol in addition to an instruction describing the purpose of the element.
- Ensure that sensory characteristics **do not** rely solely on one characteristic. Notice in the example above, the instruction uses color, shape, and textual information to describe the user interface element.

1.3.4
Orientation (AA)

(Ward, 2013)

Some of you may have auto-rotate disabled on your mobile devices because you like to view content in one way; some, however, are okay with auto-rotate!

Auto-rotate is not an option for a user with a motor impairment who may struggle with hand-related tasks. That user needs to view content in a certain orientation and needs to be able to change the orientation on demand.

TO MEET THIS CRITERION:

- Your application needs to be fully available in *both* orientations: portrait and landscape. If the application is distorted in one view, then this success criteria fails.

1.3.5
Identify Input Purpose (AA)

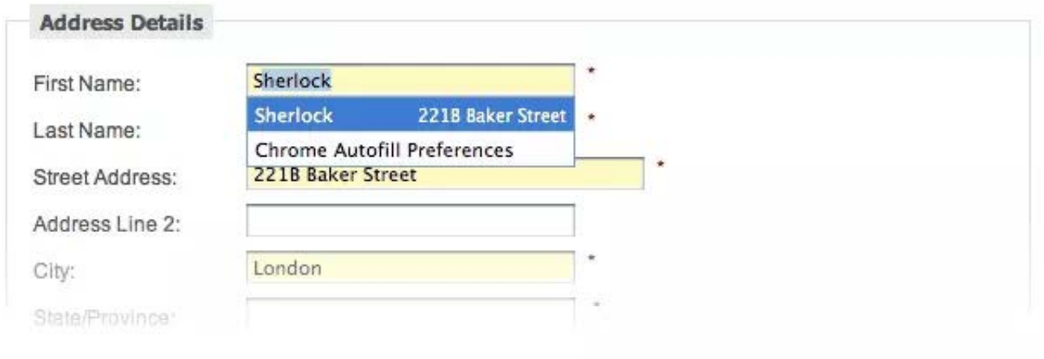
This criterion is not only a game-changer for users with motor or cognitive impairments, but it is also beneficial for non-impaired users. The purpose of the criteria is (1) to simplify the process of entering data into a form, and (2) to ensure that the user enters the *correct* data.

We can achieve these goals by providing detailed labels for input fields so that browsers can accurately auto-fill personal information.

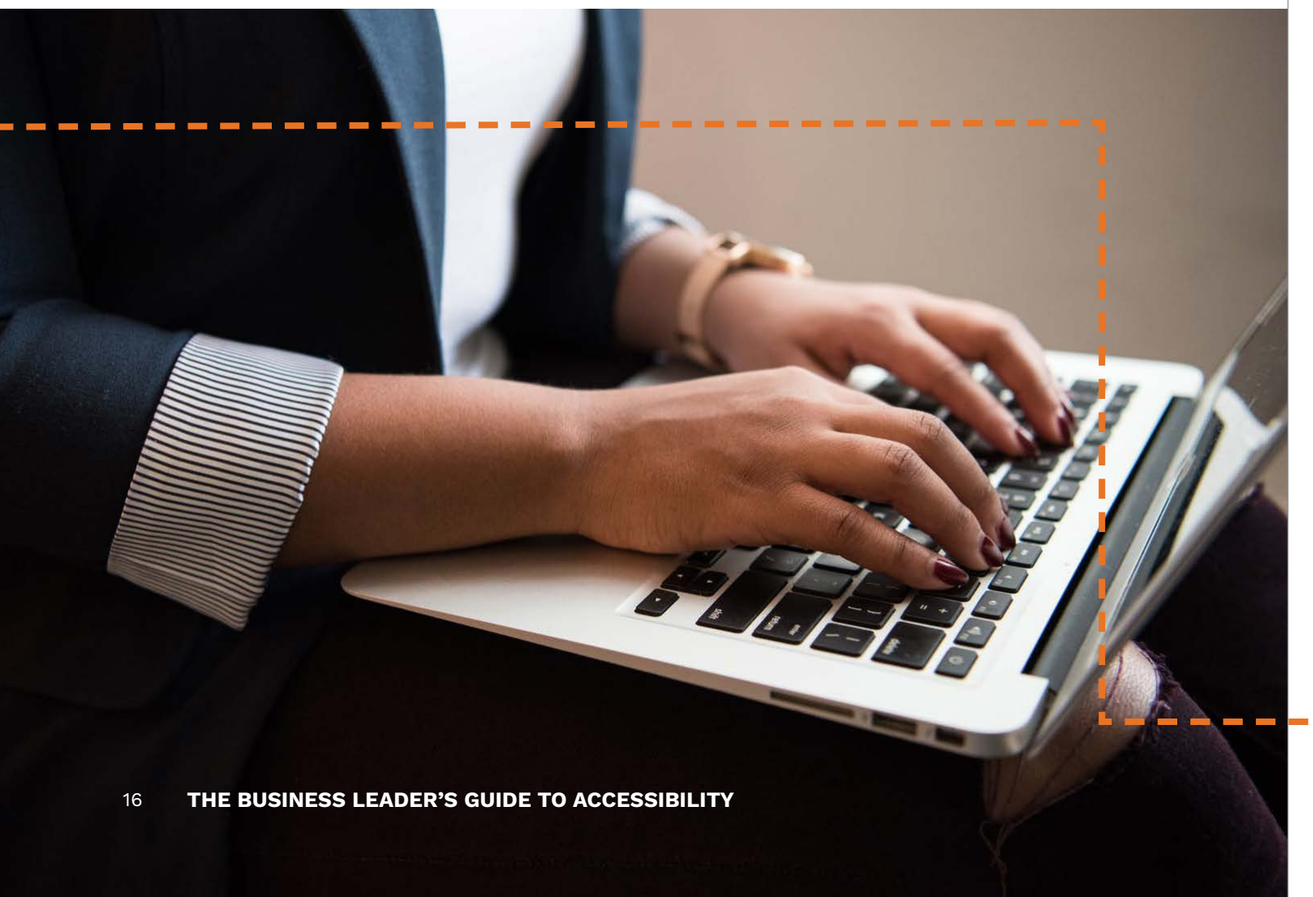
Oftentimes detailed labels and instructions are sufficient to complete forms. However, in some instances, there may be criteria that are not completely specified. For example, a form may have multiple phone number fields; this can be the user's home phone number, mobile phone number, or work phone number.

TO MEET THIS CRITERION:

- Optimally label input elements so that all browsers are able to accurately fill in personal information.



(Shankland, 2012)



1.4 Section Preview: Distinguishable

These criteria ensure that users can easily see and hear content, and distinguish foreground and background.

- 1.4.1 Use of Color (A)
- 1.4.2 Audio Control (A)
- 1.4.3 Contrast (Minimum) (AA)
- 1.4.4 Resize text (AA)
- 1.4.5 Images of Text (AA)
- 1.4.10 Reflow (AA)
- 1.4.11 Non-text Contrast (AA)
- 1.4.12 Text Spacing (AA)
- 1.4.13 Content on Hover or Focus (AA)

1.4.1 Use of Color (A)

Have you ever opened up a calendar program and noticed that today's date is shaded in green? Why is that the case? This is another instance of how designing for accessibility is designing for all.

Color should not be the **only** means of conveying information. This criterion benefits users with partial sight, colorblind users, users who have difficulty with sensory overload (and thus have disabled non-textual information), and users with a cognitive impairment who may not be able to differentiate certain colors and/or hues or immediately associate them with something meaningful.

TO MEET THIS CRITERION:

- Provide a textual cue, in addition to the colored element, to differentiate between similar, but distinct elements. For instance, if you decide to implement a red dot as a warning, make sure it has some associated, helpful text.
- The example to the right successfully passes this criterion because the cells are not simply green or red. They also have textual information so that all users can parse the meaning of the cell.

(Mindgrub, 2019)

1.4.2
Audio Control (A)

Have you ever been at work and had the volume turned up on your laptop? If you visit a site that welcomes you with a video on auto-play, you'll be scrambling to turn off the volume to limit the distraction to your coworkers.

You may be able to quickly click a button to mute the volume on your device, but this can be problematic for a screen reader user. Typically, the volume for their assistive technology is controlled through the hardware so simply turning off the hardware volume will also cancel the screen reader volume.

This also impacts the cognitively impaired user who may have trouble focusing on content while hearing audio in the background. It's important that all users are able to easily control any audio.

TO MEET THIS CRITERION:

- There needs to be a method for turning off any audio that plays for more than 3 seconds.
  - If the audio automatically stops playing after 3 seconds, then this success criteria passes.
  - If the audio plays for more than 3 seconds, a method *within* the application needs to be provided to turn off the volume so that the user can freely navigate the application.

1.4.3
Contrast (Minimum) (AA)

Have you ever encountered an application that you had trouble viewing because the text was white on grey or grey on black? Imagine how someone who has low vision feels, especially those who have a color vision deficit.

This criterion establishes a minimum contrast ratio, or a ratio of the foreground (text) to the background. Meeting this minimum contrast ratio ensures your text is legible and encourages all users to stay engaged with your content.

Listed above to the right are several tools you can use to check the contrast ratio.

TO MEET THIS CRITERION:

- The visual presentation of text and any images that contain text should have a contrast ratio of 4.5:1 for any text 12pt or below, and 3:1 for text larger than 18pts.
- In order to test the contrast ratio, you'll need to download a tool. We recommend the Colour Contrast Analyser, which comes with the Web Accessibility Toolbar (WAT).
- Another effective resource is the wave tool, a Google Chrome and Mozilla Firefox browser extension that checks for the contrast of textual information on a page. It also checks and validates other aspects of accessibility.
- Additionally, you can inspect the foreground and background colors of textual information and input them on the WebAIM checker to analyze color contrast.

Have you ever encountered an application that you had trouble viewing because the text was white on grey or grey on black? Imagine how someone who has low vision feels, especially those who have a color vision deficit.

1.4.4
Resize Text (AA)

Sometimes you have to zoom in on your browser screen. You shouldn't have to lose content for an enhanced view!

This criterion refers to resizing the browser and ensures that textual information and content are still available when the user zooms in. This maintains the integrity of the web application for those with a visual impairment.

TO MEET THIS CRITERION:

- Increase browser size 200% and ensure that all content and information are still available.
- For AA criteria, content can still be accessible by scrolling. However, 1.4.10 tightens the requirements on accessible content in an enhanced view.

1.4.5
Images of Text (AA)

Sometimes making textual information into an image enhances the aesthetic of the site. However, this can be a major loss for users who have a default setting on their hardware or browsers to display text only.

In this regard, the user who disables images during their experience will not receive this otherwise important information. What's more, someone with low vision using assistive technology cannot manipulate images of text. A screen reader user will receive the alt text for the image, but may not receive the textual information contained within the image unless the long alt description <longdesc> is used.

Other incidental benefits to adhering to this criterion include search engine optimization (search engines will crawl text, helping you rank for keywords included in the copy) and email deliverability (email clients often interpret emails with images only as SPAM and use textual cues to determine whether an email should make it to a user's primary inbox).

TO MEET THIS CRITERION:

- If an image of text is essential, include text directly under the image of text with the same information contained within the image.

**Note: 1.4.6 - 1.4.9 are AAA criteria. For more information on AAA, see the conclusion.*

1.4.10
Reflow (AA)

Responsive design means designing for all platforms - mobile, tablet, and desktop. Responsive websites adjust the way content is displayed based on the user's device, screen size, and orientation.

Have you ever opened a website on your phone to find that it isn't mobile-friendly? To read the homepage, you have to scroll all over - left to right, up and down. It's a big pain, isn't it?

This can be particularly annoying for someone with low vision, who may need to zoom in more on any given site. Scrolling from left to right then top to bottom can negatively impact their experience and cause disorientation. Addressing this criterion will make for a better experience for every user.

TO MEET THIS CRITERION:

- Make your content responsive. View it on different devices and make sure the user does not have to scroll in multiple directions to receive information.
- A user should be able to read the enlarged text in a single column.

1.4.11
Non-text Contrast (AA)

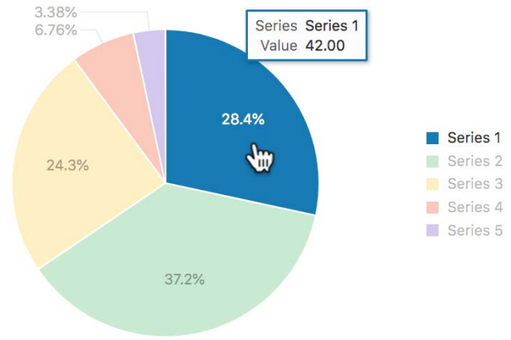
Previously, we discussed contrast ratio for text and images of text. This criterion checks the contrast ratio for *non-text* user interface components such as radio buttons, filled checkboxes, and the outline of form fields against its adjacent colors. This criterion also checks the contrast for graphical objects such as icons, charts, or dynamic infographics.

Why are these criteria separate, you may ask?

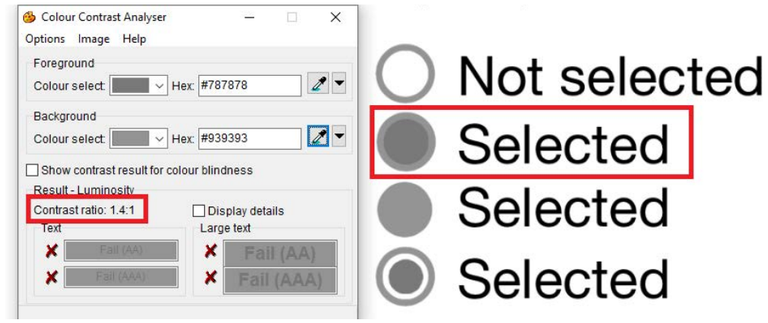
Well, when someone interacts with a particular user interface element, the color contrast of that element may change to indicate a change of state. As a result, the contrast ratio for non-text content is 3:1. It's important to note that this is only for *active controls.*

Just like 1.4.3, this does not take disabled or inactive controls into consideration. Inactive controls are purposely styled with *lower* contrast to indicate the current *inactive* state. Once the element is *activated,* the contrast **must** meet the established 3:1 ratio.

Additionally, graphical objects need to meet the contrast ratio of 3:1. The example to the right is a dynamically changing pie chart. "Series 1" is the active data element; the blue section of the pie chart and the white background have a contrast ratio of 4.7:1. The other colored sections of the pie chart do not meet the 3:1 ratio, which is acceptable as they are not the active element. Thus, this graphical element successfully meets this criterion.



(W3C, 2018)



(W3C, 2018)

Above is an example of a failure for a radio button, tested using the Colour Contrast Analyser. The contrast of the filled radio button to its outline is 1.4:1. This means it would be challenging for an individual with an impairment to decipher the focus of the element, and as a result the element fails to meet this criterion. The fourth radio button is the best implementation for a selected radio button. With a contrast ratio of 4.4:1, this element satisfies this criterion.

TO MEET THIS CRITERION:

- Ensure that the visual presentation of **active** user interface elements and graphical objects has a contrast ratio of 3:1.
- Test the contrast ratio using the WebAIM checker or the Colour Contrast Analyser that comes with the Web Accessibility Toolbar (WAT).

1.4.12
Text Spacing (AA)

Users need to have the ability to alter how they view textual content.

A user with low vision or a cognitive limitation such as dyslexia needs to be able to manipulate their textual landscape to fit how they process information.

The text does not have to default to a certain display, but the software and/or the application needs to be able to accommodate the user if they alter the textual presentation within an application - without distorting the content.

Applications or software need to be able to accommodate the following text styles:

- Line height (line spacing) to at least 1.5 times the font size
- Spacing following paragraphs to at least 2 times the font size
- Letter spacing (tracking) to at least 0.12 times the font size
- Word spacing to at least 0.16 times the font size

TO MEET THIS CRITERION:

- Ensure text fits within the respective container without being cut off.
- Ensure text fits within the container without overlapping other containers.

1.4.13
Content on Hover or Focus (AA)

Have you ever been on a website and triggered a tooltip (or a hint) by hovering over something with the mouse or accidentally clicking on something?

Your first thought is usually, "Where did that come from?" Imagine how the following users might feel:

1. The keyboard-only user who did not intend to trigger the action
2. A user who is blind who may not know that an action has been triggered
3. A user with a cognitive impairment who doesn't understand this new information, where it came from, or how to get rid of it

It can disrupt, and even derail, a user's experience with your application.

TO MEET THIS CRITERION:

When additional content appears, it needs to be:

- **Dismissible:** The user needs to be able to **dismiss** the content. This can be implemented by activating an acknowledgment button of sorts, such as a close button or escape key. See more on the Pointer Cancellation criterion (2.5.2).
- **Hoverable:** The content also needs to be **hoverable**. Sometimes, content appears by selecting a related element, however, when the additional content appears, and the user attempts to hover over it, it disappears. This success criterion ensures that the additional information can be hovered over so that the user can receive the full information.
- **Persistent**: Finally, additional information must be **persistent**. A user without an impairment who receives additional content will say "where did that come from?" Sometimes this content disappears after a few seconds, and you end up wondering where it went. The content needs to be available until the user initiates a trigger to dismiss the information.

Responsive websites adjust the way content is displayed based on the user's device, screen size, and orientation.

2 OPERABLE:

Are all components of the site capable of being used by all users?

2.1 Section Preview: Keyboard Accessible

These criteria ensure that users can access all content via the keyboard.

- 2.1.1 Keyboard (A)
- 2.1.2 No Keyboard Trap (A)
- 2.1.4 Character Key Shortcuts (A)

2.1.1 Keyboard (A)

This may be the *most critical* success criterion.

If this criterion fails, two user groups are automatically affected: users with motor impairments *and* users with visual impairments.

Those with motor impairments rely primarily on the keyboard to navigate an application. If elements on the page are not keyboard accessible, then the user cannot access certain information or initiate certain actions.

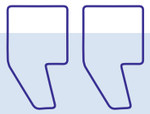
A screen reader user utilizes the keyboard in conjunction with their assistive technology to access elements. If elements are not accessible via the keyboard, then the screen reader user cannot navigate the application either.

TO MEET THIS CRITERION:

Check for keyboard accessibility using the following common keystrokes.

- Tab - navigates forward to interactive elements (keyboard & screen reader)
- Shift + Tab - navigates backward to interactive elements (keyboard & screen reader)
- Ctrl + Tab + navigates between browser tabs within a single browser window (keyboard & screen reader)
- Alt + Ctrl + Tab - navigates between open applications (keyboard & screen reader)
- Alt + Arrow Keys - reads content within the application from top to bottom (screen reader)
- F5 - refreshes application (keyboard & screen reader)
- Ctrl + R - refreshes application (keyboard & screen reader)

This criterion goes hand in hand with the focus visible criteria (2.4.7). An element can be accessible, but if the focus is not indicated or it is barely visible, then a keyboard user may not know that it is accessible.



For a person who doesn't rely on the keyboard to navigate an application, they can simply use the mouse to get to where they need to in the application. This isn't the case for keyboard users, so it's our job to make sure that all content is keyboard-navigable.

2.1.2
No Keyboard Trap (A)

If a user's primary means for navigating is with a mouse, it's unlikely they will ever be stuck within an experience without an intuitive way to exit. However, it *is* possible to get trapped in an application as a keyboard user.

A keyboard trap is when a user is unable to navigate beyond a certain area in the application or is "trapped" within certain elements. The thought of being trapped is actually pretty terrifying. For a person who doesn't rely on the keyboard to navigate an application, they can simply use the mouse to get to where they need to in the application.

This isn't the case for keyboard users, so it's our job to make sure that all content is keyboard-navigable. Some common areas of an application in which keyboard users are likely to get trapped include:

- The address bar
- Complex tables
- Calendar widgets

TO MEET THIS CRITERION:

- This may be one of the first things to test from a keyboard perspective. Ensure that you can freely navigate throughout the entire application using the keyboard shortcuts mentioned in 2.1.1, including initiating and exiting modals and popups.

**Note: 2.1.3 is part of AAA criteria. For more information on AAA, see the conclusion.*

2.1.4
Character Key Shortcuts (A)

Certain applications allow for keyboard shortcuts, which allow the user to select certain keys to quickly access different content. This criterion states that if an application allows for keyboard shortcuts, there is a method to:

1. Turn off the shortcut
2. Remap the shortcut to include one or more non-printable keyboard keys (e.g., Ctrl, Alt)
3. The keyboard shortcut is only active when that component has focus.

Not having the option to modify a keyboard shortcut can negatively impact the user's experience with the application. Keyboard users may initiate a certain change in context due to accidentally activating a keyboard shortcut.

In other instances, the user may have to use that key for other tasks, such as typing into an editable field. Additionally, speech input users need to be able to dictate commands without the initiation of an action.

TO MEET THIS CRITERION:

- Ensure that the user has the means to turn off the shortcut.
- Ensure that the default shortcuts can be remapped the shortcut to include one or more non-printable keyboard keys (e.g., Ctrl, Alt).
- Ensure that the keyboard shortcut is only active when that component has focus.

2.2 & 2.3 Section Preview:
Enough Time & Seizures and Physical Reactions

These criteria ensure that users have enough time to read and use the content, and that content is not designed in a way known to induce seizures or physical reactions.

- 2.2.1 Timing Adjustable (A)
- 2.2.2 Pause, Stop, Hide (A)
- 2.3.1 Three Flashes or Below Threshold (A)

2.2.1
Timing Adjustable (A)

I need more time! Filling out time-sensitive forms can make anyone feel anxious. Users with impairments often need additional time to complete tasks or forms.

For instance, a user with a motor impairment relies on the keyboard to access form fields and does not have the convenience of using the mouse to quickly navigate to specific parts of the form. They have to use the tab key to access every element, which will require additional time.

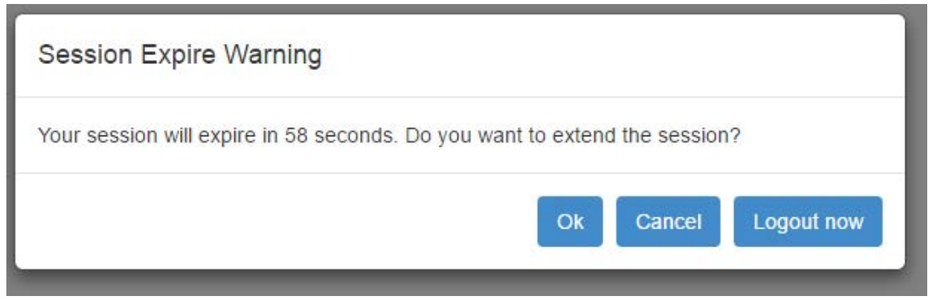
A user who is blind will need to listen to all the content on the page and use assistive technology to input data. A user with a cognitive impairment may need a little more time to read and understand the content.

Allowing time extensions doesn't just benefit users with impairments; it benefits all users!

Have you ever been filling out a form with sensitive information, and clicked over to another tab to look something up or walked away from your computer to grab your credit card?

You'd be pretty frustrated if you came back to the form only to find out that your session expired, all progress was lost, and you have to start all over again.

It is vital that users be able to extend their time if needed.



(Buddha, 2017)

TO MEET THIS CRITERION:

The user needs to be able to do one of the following:

- Turn off the time limit
- Adjust the time limit before it starts, and extend it up to at least 10 times the default limit
- Extend the time limit at least 10 times, and be prompted with a warning message that lasts at least 20 seconds when time is nearing expiration

There are three exceptions:

- If the time limit is part of a real-time event and there is no possible alternative
- If the time limit is absolutely essential to the activity
- The time limit is longer than 20 hours

2.2.2
Pause, Stop, Hide (A)

This success criterion states that for any moving, blinking, scrolling, or auto-updating information that:

1. Starts automatically
2. Lasts more than five seconds, and
3. Is displayed in parallel with other content

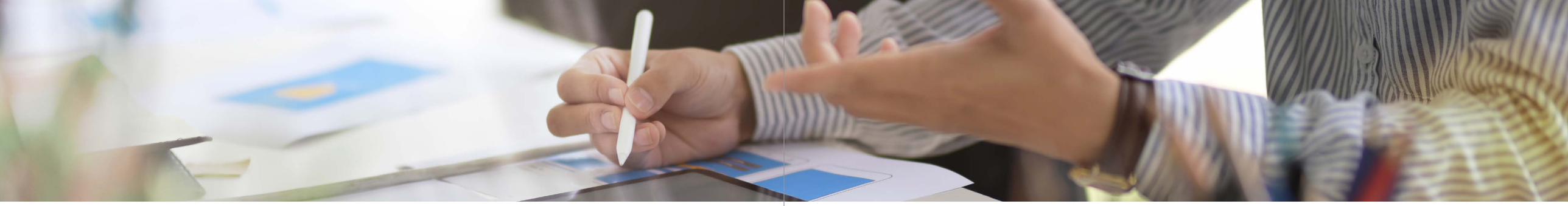
There must be a method for the user to pause, stop, or hide it *unless* its movement is essential to activity.

Moving, blinking, scrolling, or auto-updating content can be disorienting to users with low vision or cognitively impaired users.

If there is auto-updating information that starts automatically and is presented in parallel with other content, there must be a method for the user to pause, stop, hide, *or* control the frequency - *unless* the auto-updating is essential to the application.

TO MEET THIS CRITERION:

- Ensure that moving content is not presented in parallel with other content such as text or images.
- Ensure that if moving content *is* presented in parallel with other content, then there is a method to pause, stop, or hide the moving content.



2.3.1
Three Flashes or Below Threshold (A)

The success of this criterion can actually prevent a medical emergency.

Some websites get fancy and include flashy calls to attention. This is acceptable, but it's important to note that the element cannot flash more than three times in a second. More than that and the element can trigger epileptic seizures in users with epilepsy.

This can generally be checked manually by eyeing the content and using a stopwatch for precision. Our recommendation, in fact, is to not add anything flashy on a site, as it can be both distracting and unattractive for all users.

TO MEET THIS CRITERION:

- Ensure that flashing content does not flash more than three times in a second.

2.4 Section Preview: Navigable

These criteria ensure that the user is offered help navigating your site, finding content, and determining where they are within your site.

- 2.4.1 Bypass Blocks (A)
- 2.4.2 Page Titled (A)
- 2.4.3 Focus Order (A)
- 2.4.4 Link Purpose (In Context) (A)
- 2.4.5 Multiple Ways (AA)
- 2.4.6 Headings and Labels (AA)
- 2.4.7 Focus Visible (AA)

2.4.1
Bypass Blocks (A)

Menus on menus, link on links, and image after image.

A heavy application can be navigated with ease by a user without impairments by simply using the mouse to click on whatever area of the application they want to access. However, this proves challenging for the keyboard-only user who uses the tab key to access content.

This criterion states that there must be a method for skipping repetitive content so that the user can skip directly into the main content area.

Without this, the user is forced to tab through all the menu links to get to their desired menu link. This is simply frustrating and time-consuming. A "skip to main link" solves this problem; this link will typically be invisible until interacted with by tabbing to it.

To check for a "skip to main" or "skip to content" link, ensure that one appears after the first couple of tab presses and **before** the navigation menu. It may appear at times as a "skip navigation" link.

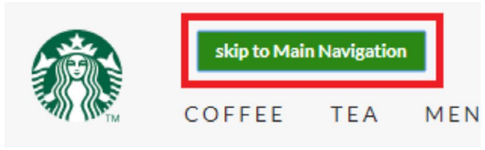
Best practice is to hide the link until a user tabs to it. However, the "skip to content" functionality also has to do what it is intended to do: skip navigation links, be available and functional for screen reader users, and meet color contrast.

If any of the following are true, the "skip to content" functionality may still be inaccessible, and this criterion will not pass.

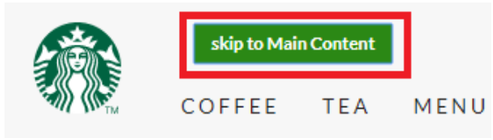
1. At times, the link is **not** selectable with standard keystrokes (Enter key or spacebar).
2. The link **does not** go to the main content area or skip the main navigation.
3. The link is **not** selectable while assistive technologies are enabled.
4. The link **does not** meet the minimum color contrast ratio for text.

As helpful as "skip to content" links can be, it is important to use them wisely. For instance, "skip to content" links on flash applications serve no purpose.

See the example below. Starbucks has a “skip to Main Navigation” link, which skips the “Find a Store”, “Sign In”, and “Join Now” links, known as the secondary navigation.



They also have a “skip to Main Content” link which skips both sets of navigation links completely, and places the user on the first interactive element within the main content area.



TO MEET THIS CRITERION:

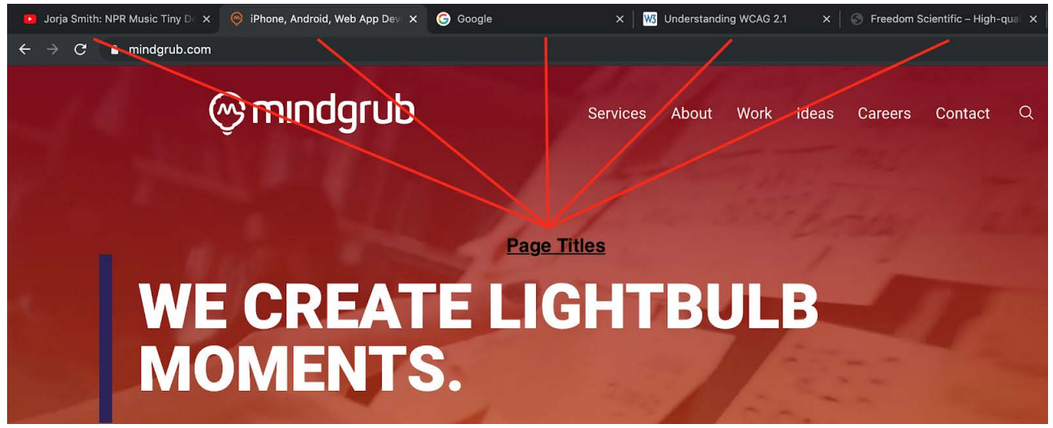
- Choose wisely. If the links are not repetitive and excessive, there may not be a need to provide a “skip to main” navigation link.
- Hide the link and only make it visible upon tabbing to it.
- Ensure that the verbiage matches the functionality. For example, “skip main navigation”, skips the primary navigation links, but will access the secondary navigation. A “skip to main content” link places the user on the first interactive element within the main content area.
- Ensure that the link is selectable via the keyboard *with* and *without* assistive technologies enabled.
- Ensure that the link meets the minimum contrast ratio for interactive elements.

2.4.2
Page Titled (A)

This criterion requires every page within a web application to have a distinct title that describes the purpose of the page.

This makes it easier for keyboard users, who may navigate back and forth between different browser pages, to quickly identify the purpose of the page by reading or hearing the title.

This is another example of accessibility criteria overlapping with best practices from other disciplines. In addition to being necessary for accessibility compliance, titles are also important for search engine optimization (SEO). A good title increases your chances of ranking higher on search engines and attracts more relevant traffic to your website.



(Harsel, 2017)

TO MEET THIS CRITERION:

- Add <title> attribute to HTML code (or using a Content Management System, or CMS) in each individual page.
- Ensure titles are differentiated amongst the set of web pages.

2.4.3
Focus Order (A)

Focus order affects users with cognitive, motor, and visual impairments. This success criterion states that when the order of content is important or meaningful, the user’s focus should proceed in a logical manner. Whether the individual is using a keyboard, screen reader, or other assistive technology, they should be able to move through the components on your site in a logical order. This reduces confusion and helps the user form a mental model of your content.

An example of successfully meeting this criterion is ensuring the user can navigate to the email address field, then the password field, and lastly the submit button when logging in.

- This allows the keyboard user to access and populate the necessary fields before submitting login information.
- This ensures that the screen reader user is not confused by the order in which elements are announced.
- Focus order also ensures that cognitively impaired users who may also use the keyboard do not become disoriented while navigating the application.

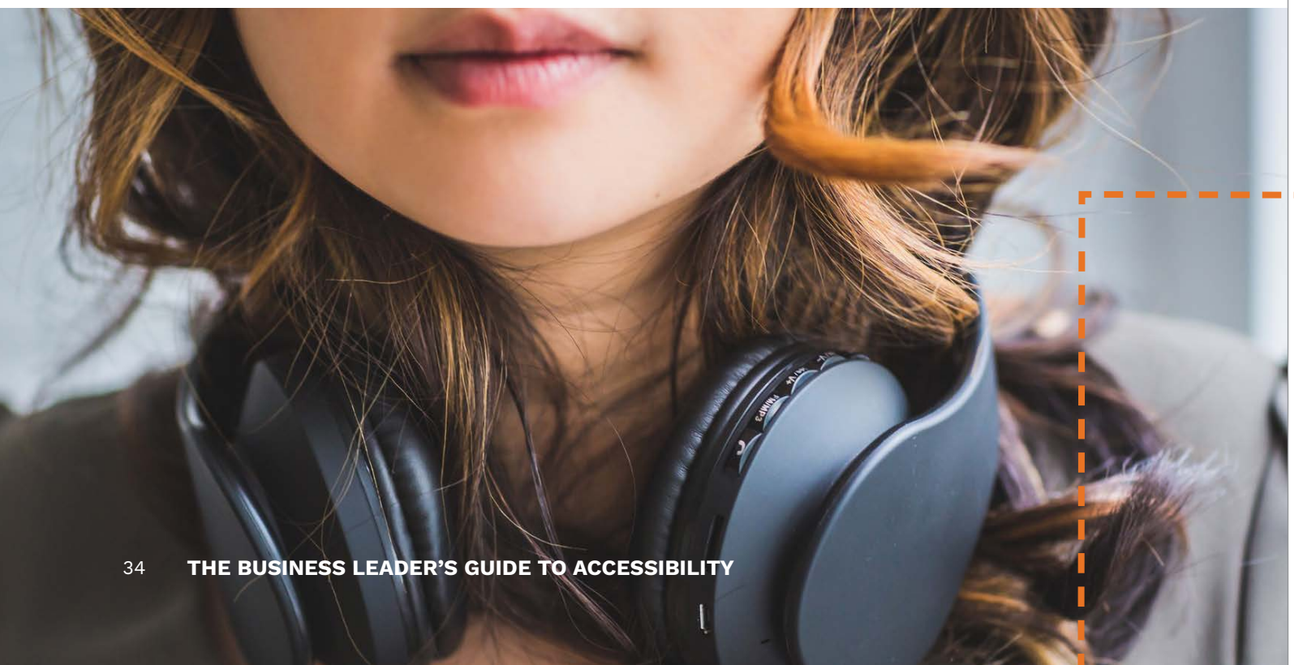
It's important to note that there may be more than one correct navigation sequence. When this is the case, only one order needs to be provided to successfully pass this criterion.

This success criterion also affects how users interact with modals and popups.

If the user selects an element that initiates a modal or popup, the next tab press should provide an indication of visual focus on the first interactive element within the popup/modal.

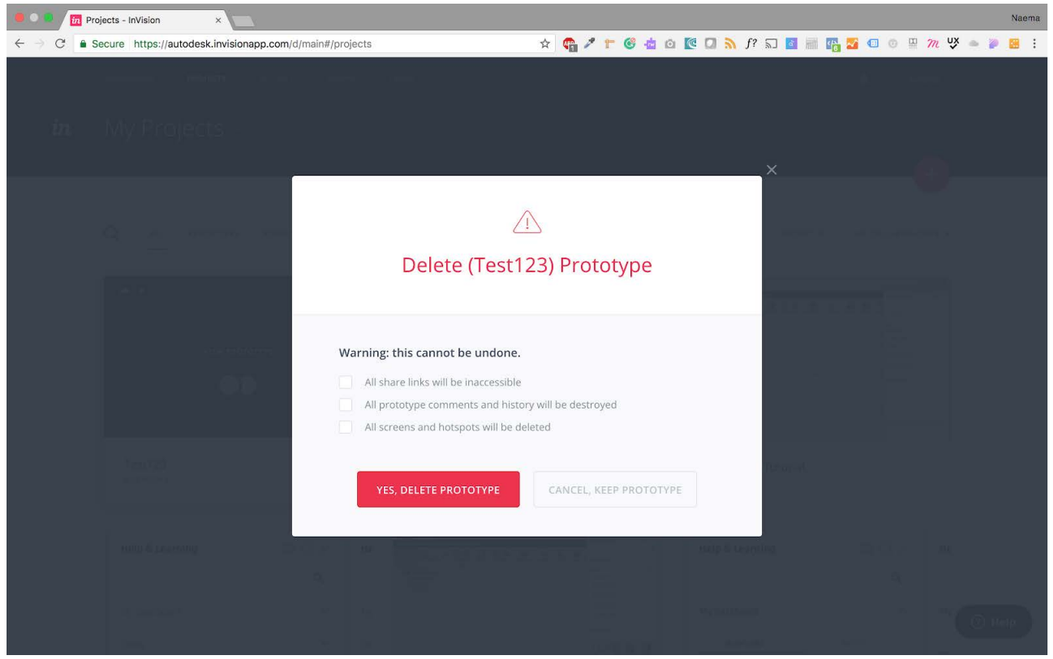
If a user presses the tab key and visual focus remains behind the popup, we violate another accessibility requirement! This would impact the focus visible success criterion (2.4.7) because the user cannot see where they are in the application.

On the other hand, let's say that focus expectedly goes to the first element within the popup or modal, and after tabbing to the last element within the popup/modal, the user accidentally presses the tab key again. The focus should go *back to the first element* within the popup/modal and should not leave the popup/modal until the user initiates an action to close the popup/modal. If pressing the tab key navigates the user outside of the popup/modal, then this also affects the focus visible success criterion and fails this criterion.



The picture below shows an example of a modal that appears after the user selects the delete button.

Upon opening the modal or the first tab press, the user is expected to land on the "All share links will be inaccessible" checkbox. The last tab stop within the modal is expected to be the "Cancel, keep prototype" button (per the meaningful sequence criteria). The next tab press after that should take the user back to the "All share links will be inaccessible" checkbox. If instead, the next tab stop goes to the URL, then the application fails for this success criteria.



(Baskanderi, 2017)

TO MEET THIS CRITERION:

- Ensure elements within the tab index flow in a logical manner (consult with a UX professional to validate your assumptions).
- Ensure focus is immediately available in the popup/modal or is available after the first tab press upon initiation.
- Ensure that focus does not leave popup/modal until the user initiates action to do so.

2.4.4
Link Purpose (In Context) (A)

"To read more blog posts from Mindgrub, click here."

The preceding sentence provides a link to Mindgrub's blog (which you should definitely check out). Visually, this success criterion is met because the user receives information about the purpose of the link from the surrounding context. If this was a random link that simply read, "link" or "here" or "read more" without providing context, the application would fail this requirement.

This success criterion impacts users with motor, visual, and cognitive impairments. By providing context around the link, we eliminate any confusion about where a link may lead, and we let the user determine the link's purpose and skip links that aren't of interest.

It's important to note that when tabbing throughout the application, only interactable elements are accessed. In this instance, a screen reader user will tab to the element and hear "here". This is where ARIA comes into to save the day. By adding an ARIA-label attribute to the link, the screen reader will announce the link as "To read more blog posts from Mindgrub, click here link"

TO MEET THIS CRITERION:

- Ensure that hyperlinks have distinct surrounding context.
- Ensure that hyperlinks are not vague.
- Ensure ARIA attributes are added to elements that need additional information or context.

2.4.5
Multiple Ways (AA)

This success criterion ensures that users can navigate a site in more than one way. Providing users with options allows them to pick the method of navigating that best meets their needs.

One example of navigation is menu links, but if a menu is extensive, it can overwhelm users with too many irrelevant options. In this instance, adding a search feature benefits users by allowing them to type in a specific query and receive a filtered list of results. This is helpful for all users, especially those with cognitive impairments who may have difficulty understanding the way a site's navigation is structured.

Additionally, having to select the back button within the browser or navigating through menu links multiple times can inconvenience keyboard users. Providing links between pages, such as "Next" and "Back" buttons or breadcrumbs can help the user navigate forward and backward in the application.

An exception to this criteria is when a web page is part of a process or series of steps. When this is the case, a secondary way to reach a particular page is not required.

TO MEET THIS CRITERION:

- Ensure that users have another method of navigating the site aside from menu links such as a search field, a table of contents, or breadcrumbs.
- Ensure that this functionality is easily identifiable and accessible.

Providing users with options allows them to pick the method of navigating that best meets their needs.

2.4.6
Headings and Labels (AA)

The Headings and Labels success criterion is essential to the success of an accessible application. To meet this criterion, all headings and labels must be descriptive.

Clear headings and labels allow *all* users to easily navigate and comprehend your content. Headings divide lengthy content into chunks, making it more scannable and thus helping users with a disability such as limited short-term memory.

The implementation of descriptive headings also impacts visually impaired users who use a screen reader. The screen reader has certain built-in keystrokes that invoke actions such as accessing headings, lists, links, and tables.

As an example, JAWS screen reader users can initialize the Virtual HTML Features, which allows the user to access HTML pages by element type.

The JAWS user can press “Insert + F3” and select “Headings” or simply press the “H” key on their Windows computer. If headings are correctly tagged within the application, the screen reader will render these sections as headings, making it easy for the user to access the content that they desire. That’s why descriptive headings are so important!

Note that this criterion does not require the use of headers and labels, only that if they are used, they are descriptive.

TO MEET THIS CRITERION:

- Ensure headings are provided for lengthy applications.
- Ensure headings are not overused.
- Ensure headings are descriptive.
- Ensure headings are tagged as headings in HTML code.

2.4.7
Focus Visible (AA)

“Whoa, where did it go?” This is what you’ll hear a keyboard user say if this success criterion is not met.

Focus Visible ensures that there is **always** a *visible* indication of focus present on the screen. Users who rely on the keyboard to access elements on the application need to know where they are on the page so that they can decide whether to interact with the element or not.

It is important to note the emphasis on *visible*. While the focus indicator does not have to take a specific form, it should be significant. At times, there is a faint indication of visual focus, either implemented by the developer or defaulted by the browser. If the outline is a browser default, it’s possible to enhance the size and color of the outline.

This criterion is only for users that rely on the keyboard because screen readers will announce where the user is.

TO MEET THIS CRITERION:

- Ensure interactive elements receive keyboard focus by tabbing through the application.
- Ensure that the outline around interactive elements is significant, so the user can discern his or her location within the application.

(Wild, n.d)

2.5 Section Preview:
Input Modalities

These criteria ensure that users can operate elements on your website with inputs other than the keyboard.

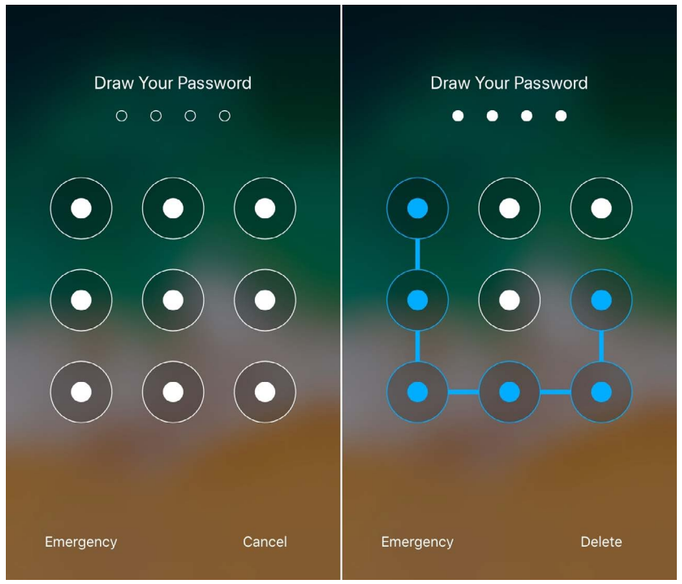
- 2.5.1 Pointer Gestures (A)
- 2.5.2 Pointer Cancellation (A)
- 2.5.3 Label in Name (A)
- 2.5.4 Motion Actuation (A)

2.5.1
Pointer Gestures (A)

Essentially, all multipoint or path-based gestures (such as pinch to zoom, a three-finger swipe, sliding, or dragging) must be able to be performed with a single point of activation, similar to activating a button.

The goal of this success criterion is that users who are unable to perform complex or precise gestures, or users who rely on specialized or adaptive input devices are still able to interact with and control the content on your website. This benefits people who have motor, cognitive, and/or visual impairments.

For example, an individual who has frequent hand tremors as a result of a stroke may struggle with multi-point gestures, like pinch to zoom. This user may also struggle with a path-based gesture, such as providing a passcode that requires swiping in multiple directions. Swiping to connect the dots and unlock your phone is an example of a path-based gesture.



(Bouchard, 2019)

Additionally, some users may utilize devices like a head pointer, an eye gaze system, or a speech-activated device. This doesn't mean you have to avoid complex gestures altogether; there just needs to be an alternative, single-point method for those who cannot execute complex gestures.

TO MEET THIS CRITERION:

- Ensure there is a single-point based gesture for multipath-based gestures.
- Ensure that if a multipoint or path-based gesture is essential to the application, there is an alternative for the disabled user who cannot execute the gesture.

2.5.2
Pointer Cancellation (A)

I think it's safe to say we've all clicked on something unintentionally at one time or another.

For a user with an impairment, this can happen easily and often. This criteria provides all users the opportunity to *cancel* an unintentional selection by performing an "up-event" that triggers one of the following:

Up-Event Activation or Completion

- This is the simplest, most accessible way to implement pointer cancellation.
- Up-event activation occurs when the pointer is released. For a touchscreen interaction, this is when the finger is lifted from the target area, and for a mouse interaction, this is when the button is released from the target area.

Up-Event Abort or Undo

- Up-event activations have a built-in ability to cancel.
- Because activation occurs only as the touch or click within the target area is *released*, users have the ability to abort or cancel the activation by moving their touch or mouse cursor *out of the target* area before releasing.
- This is particularly useful for users who have difficulty using a mouse or touchscreen accurately. When they press an item, there is usually a visual feedback cue letting them know the item has been pressed. If the cue tells them they have selected the wrong item, they can simply cancel by moving out of the target area.
- You may see this on a "push to pay" or "hold to sign in" experience.

Up Reversal

- In instances where the down-event triggers activation, the up-event can reverse the activation. Press-and-hold actions are an example of this. For instance, pressing and holding on a button may cause a popup to display, and releasing the button will cause the popup to disappear again.

Down-Event

- A down-event completion is only allowed when using the up-event would alter the desired outcome of a behavior. For example, when using a keyboard to type, the letters appear when each key is pressed, not released. Time-sensitive activations are another instance where the down-event completion is essential.

TO MEET THIS CRITERION:

- Ensure one of the following functions can be utilized to cancel unintended actions:
  - Up-event activation or completion
  - Up-event abort or undo
  - Up reversal
- Ensure that the down-event is not used for cancellation.
- Ensure that the cancellation action is documented.

2.5.3
Label in Name (A)

This criterion ensures that if an interactive element has text, the accessible name used by assistive technology is visually present in the text.

This is especially important for speech-input users. Let's say that a user is utilizing the Android or iPhone accessibility features to access the 'username' field within the login screen. If the user sees "username" and speaks that command, but the accessible name is "login username", the phone will not focus on that field because the *accessible name* was not spoken.

If there is a mismatch in labels, assistive technology may not recognize what the user is asking, making it challenging for the user to land on the appropriate field.

To check for this criterion, ensure that the visual label matches what is spoken by assistive technology, such as a screen reader.

TO MEET THIS CRITERION:

- Ensure accessible names match visual names.
- Check HTML code for role/id name and visual name.

2.5.4
Motion Actuation (A)

There are some really cool apps out there nowadays, but some of the functionality within these apps can negatively impact a user's experience.

For instance, some functionality is triggered by moving the device or by waving at a screen. When this is the case, there must be a method to disable the action. This ensures that a user who may not have control of their movements does not accidentally initiate an action.

There must also be a way for the user to trigger the same action through a standard control, like a button.

TO MEET THIS CRITERION:

- Ensure that all functionality requiring special movement can be disabled.
- Ensure there is a standard keyboard alternative that provides the same result as the special gesture.

3 UNDERSTANDABLE:

Are all elements and information comprehensible?

3.1 Section Preview: Readable

These criteria ensure that the user can read and understand all content.

- 3.1.1 Language of Page (A)
- 3.1.2 Language of Parts (AA)

3.1.1 Language of Page (A)

In a globalized world, the language in which content is presented is crucially important.

This is simply defined in the HTML code as e.g., <lang="en"> . This allows the page to render in the appropriate language and enables assistive technologies to effectively render information to the user.

Defining the language allows screen readers to use the correct pronunciation rules, and helps browsers appropriately render special characters, different alphabets, and languages that read right to left versus left to right.

TO MEET THIS CRITERION:

- Ensure language of page is provided in HTML code on each page within the application.

3.1.2 Language of Parts (AA)

Sometimes certain text within a page is presented in a different language than the majority of the copy. It is important for this to be specified in the HTML code so that assistive technologies can render it accordingly. For example, an online Italian cookbook may have recipes in English followed by a quote in Italian.

Exceptions to this rule:

- Proper names
- Technical terms
- Words or phrases that are common vernacular, such as the French *rendezvous*

TO MEET THIS CRITERION:

- Ensure the textual parts of application that are different from the default language have the specified language tag.
- Ensure the specified text is tagged in the HTML code in its respective language e.g., <blockquote lang="es">.

3.2 Section Preview: Predictable

These criteria ensure that pages on your website work in consistent and predictable ways.

- 3.2.1 On Focus (A)
- 3.2.2 On Input (A)
- 3.2.3 Consistent Navigation (AA)
- 3.2.4 Consistent Identification (AA)

3.2.1 On Focus (A)

This criterion ensures that focus does not initiate a change in context on the page. This is important for **all** users. Common examples of a context change on focus are forms that automatically submit or a new window that launches when a component receives focus.

If a keyboard user tabs to an element but does activate it, it should not change the state of the element or the page. This also impacts the screen reader user who relies on the keyboard to navigate the application. Furthermore, this will severely disorient a cognitively impaired user.

This criterion can be checked in conjunction with the no keyboard trap (2.1.2) requirement while tabbing throughout the site to ensure that no element that receives focus initiates a change in the page.

TO MEET THIS CRITERION:

- Slowly tab throughout the application to ensure that interactive elements do not change the state of the application upon landing on an element.

> **In a globalized world, the language in which content is presented is crucially important.**

3.2.2
On Input (A)

Similar to the On Focus success criterion (3.2.1), this requirement ensures that inputting certain content does not change the context of the page until some sort of submit button is activated.

This is true for all instances, *unless* the user is notified that changing the state of the element will change the context of the page.

This success criterion is important because if a change of context happens it can negatively impact users with motor, visual, and cognitive impairments. A user with a visual impairment wouldn't be able to see the change in context, and a user with a cognitive impairment will be confused by the unexpected change.

This criterion can be checked while populating a form to ensure that all form fields do not initiate a change in context. An example of this can be a search field where, while typing, search results auto-populate the page without a search button being pressed. In this case, the success criterion would fail.

However, let's say that there are two different types of logins on a page, like "student" and "staff," and a radio button changes the login experience between the two options. If there is text describing the expected change of state for the element *before the fieldset*, the success criterion is met.

TO MEET THIS CRITERION:

- Ensure that all input elements do not change the context of an application without finalization from the user ("submit", "continue", and "accept" buttons).

3.2.3
Consistent Navigation (AA)

There's so much we can do with CSS, but it's best to keep things simple as to not disorient users that may suffer from a cognitive impairment or low vision. This criterion ensures that navigational links that are repeated across several pages are presented in the same order throughout the application. Oftentimes a user with an impairment or disability may have to familiarize themselves with the page in order to navigate it. If the navigation order changes unexpectedly, the user will be confused.

TO MEET THIS CRITERION:

- Ensure that navigation for an application containing multiple pages is presented in the same order throughout the application.

3.2.4
Consistent Identification (AA)

Similar to consistent navigation (3.2.3), this criterion ensures that components that have the same functionality are identified consistently throughout the application.

For instance, a failure needs to be presented consistently. Below is an example of two different icons representing an error or failure. If an application uses these interchangeably, this criterion would fail.

Presenting failures in different ways - either with different labels, different accessible names, or different non-text items - can confuse the user.

Common examples of elements that must be consistently identified:

- Success
- Failure
- Proceed
- Return
- Payment
- Submission
- Cancellation

Additionally, this criterion needs to be implemented in conjunction with sensory characteristics (1.3.3), so that the user receives additional, identifiable information about the element, and not just the icon.

TO MEET THIS CRITERION:

- Ensure that elements are consistently identified throughout the application.

3.3 Section Preview: Input Assistance

These criteria help users avoid and fix mistakes while inputting information.

- 3.3.1 Error Identification (A)
- 3.3.2 Labels or Instructions (A)
- 3.3.3 Error Suggestion (AA)
- 3.3.4 Error Prevention (Legal, Financial, Data) (AA)

3.3.1 Error Identification (A)

Have you ever wondered where *exactly* you went wrong during a task? This happens all the time when using software applications, which is why error identification is so important.

This criterion states that errors must be identified and explained to the user so that they know how to fix the mistake. It's important that this is done in text, so users utilizing assistive technologies can hear the error or see the error textually. The message should be as specific as possible so the user knows exactly what to fix and how and to avoid any confusion or ambiguity.

Sometimes an errored field is simply highlighted in a certain color (which disregards users with color blindness) or denoted by an error icon (which impacts users with cognitive impairments who may not know the meaning of the icon intuitively).

This success criterion benefits **all** users by requiring the error information be provided in writing. Isn't it helpful to know *exactly* where you went wrong instead of guessing why you're unable to submit your form?

TO MEET THIS CRITERION:

- Ensure that error messages are displayed inline with the field where the error occurred.
- Ensure errors have a descriptive, textual context to the error that occurred.

3.3.2 Labels or Instructions (A)

Have you ever received an error for entering the right data in the incorrect format? Like typing your phone number without the area code, or with dashes when they only want numbers?

Don't you wish you knew the proper format to begin with?

This criterion takes care of just that. If you're asking a user to input information, you should provide instruction or a label specifying what information you're looking for and the required format of the answer.

This includes date formats and phone number formats.

What about those asterisks (*)? A cognitively-impaired user may not associate the asterisk with anything informative to their experience at all.

Understandably, it's not visually appealing to provide a textual instruction for each input field, but providing instructions for the meaning of this symbol at the *beginning* of the form helps the user understand what information is required to complete the form.

Doing so will avoid confusion and frustration - and it's just good practice when creating forms.

TO MEET THIS CRITERION:

- Ensure input fields have descriptive labels and any additional information needed to complete the field.
  - If there's limited space for additional text, provide a tooltip adjacent to the form.
- Provide instructions for fields that only accept a specified format where more than one format is common (like dates or phone number).
- Ensure that legends are provided for symbols and characters (like asterisks).

3.3.3
Error Suggestion (AA)

This criterion states that if an input error is automatically detected and there are suggestions for how to correct that error, then suggestions should be provided to the user, unless it compromises the security or purpose of the content.

In other words, instead of prompting the user with a generic error message, tell the user exactly how to correct their mistake.

This is particularly useful for users with cognitive disabilities who may need assistance with completing a form.

For example, say an application requires a user to log in. The username is their email address, which requires the prefix and the domain, but the user only inputs the prefix of their email. Upon selecting the submit or login button, they receive an error suggestion to enter the full username as "*Must use required format. Example: user@email.com*".

In this instance, the user receives information on how to correct their error before re-submitting the form. This reduces the volume of form abandonments that result from frustration or confusion.

TO MEET THIS CRITERION:

- Ensure that if the user is unable to submit a form due to an incorrectly-formatted input, they receive **descriptive** information to modify that input.

3.3.4
Error Prevention (Legal, Financial, Data) (A)

This criterion states that users need to be able to reverse, verify, or confirm any submission that is legally or financially binding.

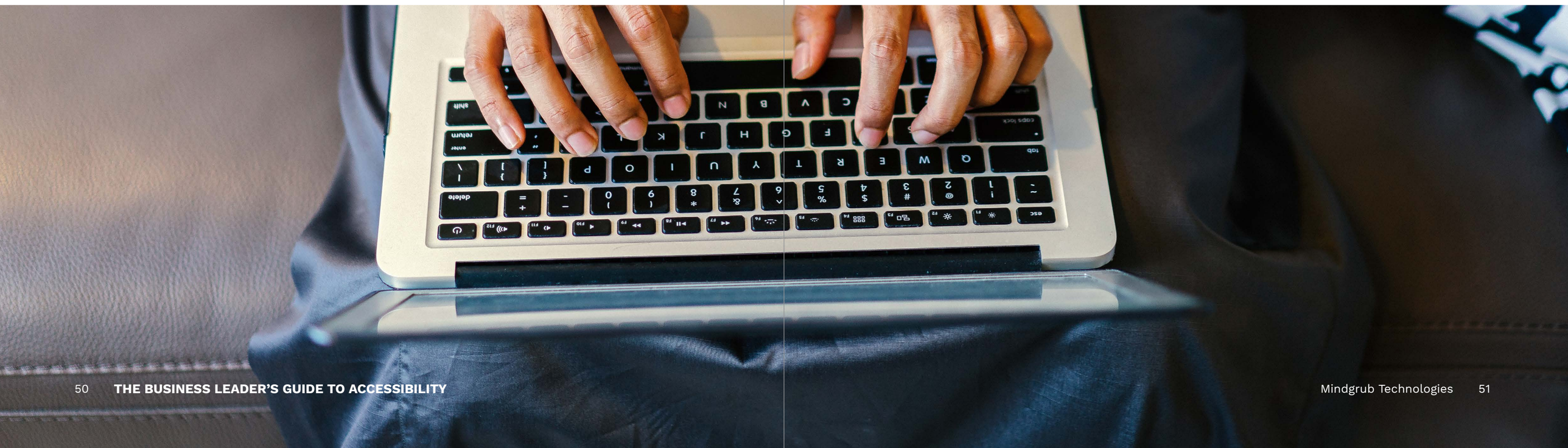
This reigns true for every user.

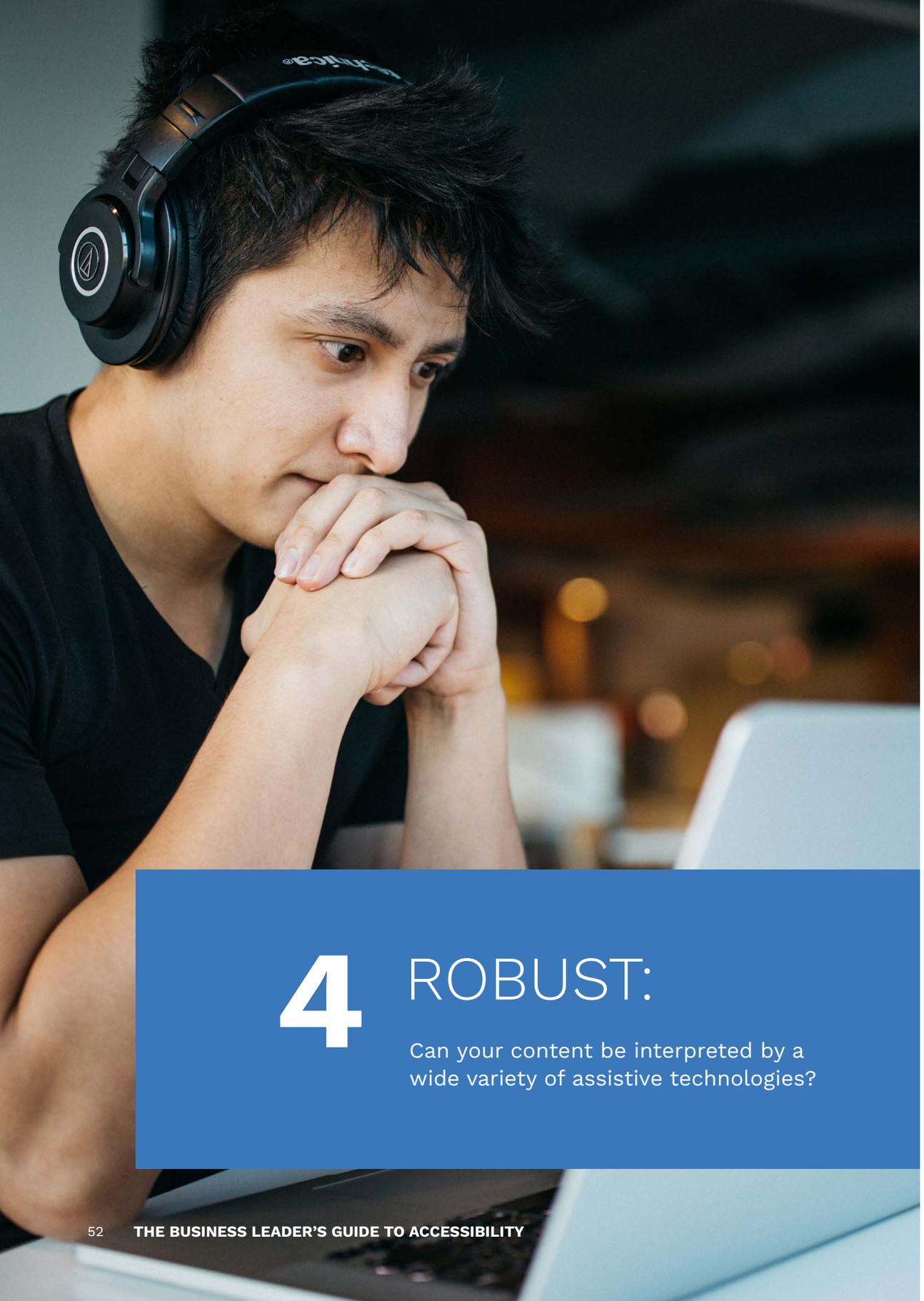
For instance, if you mistyped your address when ordering something, wouldn't you want to be able to review your submission? Or cancel it before it's shipped to the wrong state?

From the busy mom filling out a form with one hand while carrying her child in the other, to a person with a visual impairment using a speech-to-text device, being able to reverse, verify, or confirm data submissions benefits everyone.

TO MEET THIS CRITERION:

- Ensure that the user is able to cancel an order or a submission, or
- Ensure user receives a full page of user-entered content before final submission, or
- Require users to confirm that the populated data is correct.





4 ROBUST:

Can your content be interpreted by a wide variety of assistive technologies?

4.1 Section Preview: Compatible

These criteria ensure that compatibility with current and future assistive technologies is maximized.

- 4.1.1 Parsing (A)
- 4.1.2 Name, Role, Value (A)
- 4.1.3 Status Messages (AA)

4.1.1 Parsing (A)

The goal of this criterion is to ensure that content created using markup languages can be correctly interpreted and parsed by assistive technologies. Exact parsing requirements vary depending on the markup language.

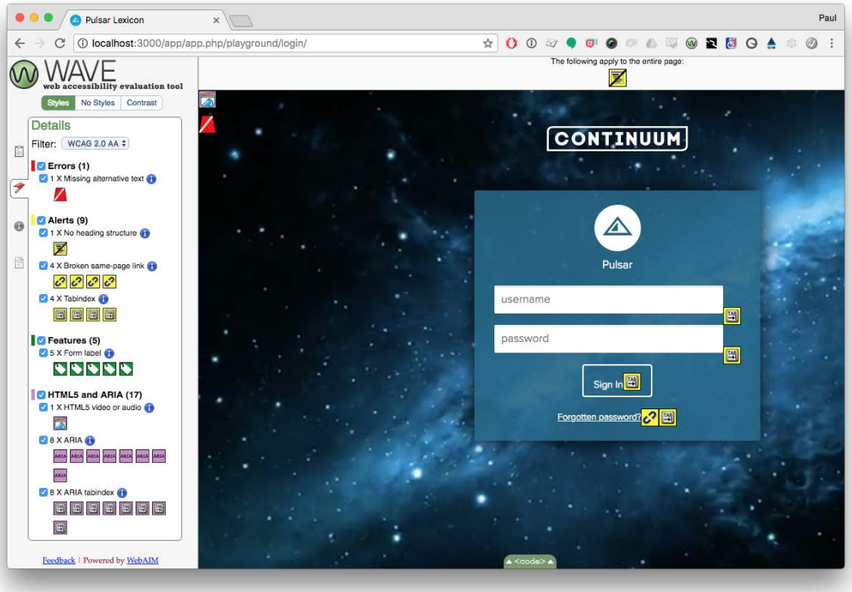
Parsing is something that is done manually while performing accessibility testing on an application. As a tester tabs throughout the application, they are essentially parsing for tab indexes and interactive elements.

As the tester selects either the enter key, spacebar, or arrow keys, they are parsing for how the element is coded. In the event that an element's action is different from what is expected, it is best practice to use the screen reader to hear the instruction for how to interact with the element as well as how the element is coded.

However, the parsing criterion primarily benefits developers.

There is a slew of software tools created specifically to alert developers of areas where their code can be improved in order to achieve accessibility at the highest level. Parsing requires a tool such as W3C Validator, Web Accessibility Toolbar (available in IE11 only), and WAVE extension (available in Chrome and Firefox).

Below is an example of how the WAVE tool parses a web page. By simply clicking on the browser extension icon, the web page will be parsed within a matter of seconds and the results will be displayed on the left pane of the browser window.



(Stanton, 2018)

TO MEET THIS CRITERION:

- Use your preferred parsing tool to review the code and ensure:
  - Pages have complete start and end tags
  - Elements are nested according to their specifications
  - Elements do not have duplicate attributes
  - IDs are unique
- Pay close attention to **errors** and mitigate them; some **warnings** may be ignored.

4.1.2
Name, Role, Value (A)

This success criterion ensures that all non-text content has the appropriate name, role, and value and is therefore able to be parsed by assistive technologies.

Name refers to the name of the element that is visibly provided on the application. Success criterion 2.5.3 states that the visible label and the accessible label should be the same so that assistive technologies can render the appropriate element.

Role refers to the type of element (such as button, checkbox, or toggle).

Value refers to the state or status of the element.

(Choudhary, 2015)

Let's look at the example to the left:

Here is a group of radio buttons. As you can see, "Inclusive" is not selected, and "Exclusive" is selected.

In this instance, when a user utilizing a screen reader accesses the "Inclusive" radio button, they should hear "Inclusive radio button **not** checked". If this is announced when the element is accessed, the success criterion will pass because the **name** ("inclusive"), **role** ("radio button"), and **value** ("not checked") are all read aloud and the user receives all the information that they need to continue with the application.

However, let's say that when the radio button is accessed, the assistive technology only announces "Inclusive radio button", and does not announce its value state as "not checked". This success criterion would fail because the user does not know the current status of the radio button, which can disrupt the user's interaction with the rest of the application.

TO MEET THIS CRITERION:

- Ensure all elements have a name or label.
  - If necessary an accessible name denoted by ARIA
- Ensure all elements are coded correctly.
- Ensure elements with a dynamic status are announced correctly.

4.1.3
Status Messages (AA)

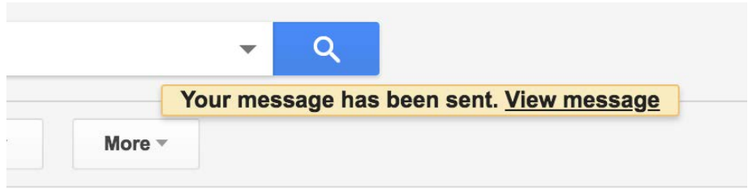
Oftentimes, users are shown a confirmation message after the successful completion of a submission activity. This should be standard practice regardless of accessibility - nothing causes apprehension like a lack of confirmation.

This criterion states that users should be made aware of changes in content that are not given focus in a way that doesn't unnecessarily interrupt their work. These status

messages include messages indicating the results of an action (success vs. failure), waiting states (searching...), progress, or errors.

A status should be concise, clearly visible, and programmatically determinable so that users utilizing assistive technologies can immediately render the message as it appears.

Below is an example of a status message that a Gmail account holder receives after sending an email.



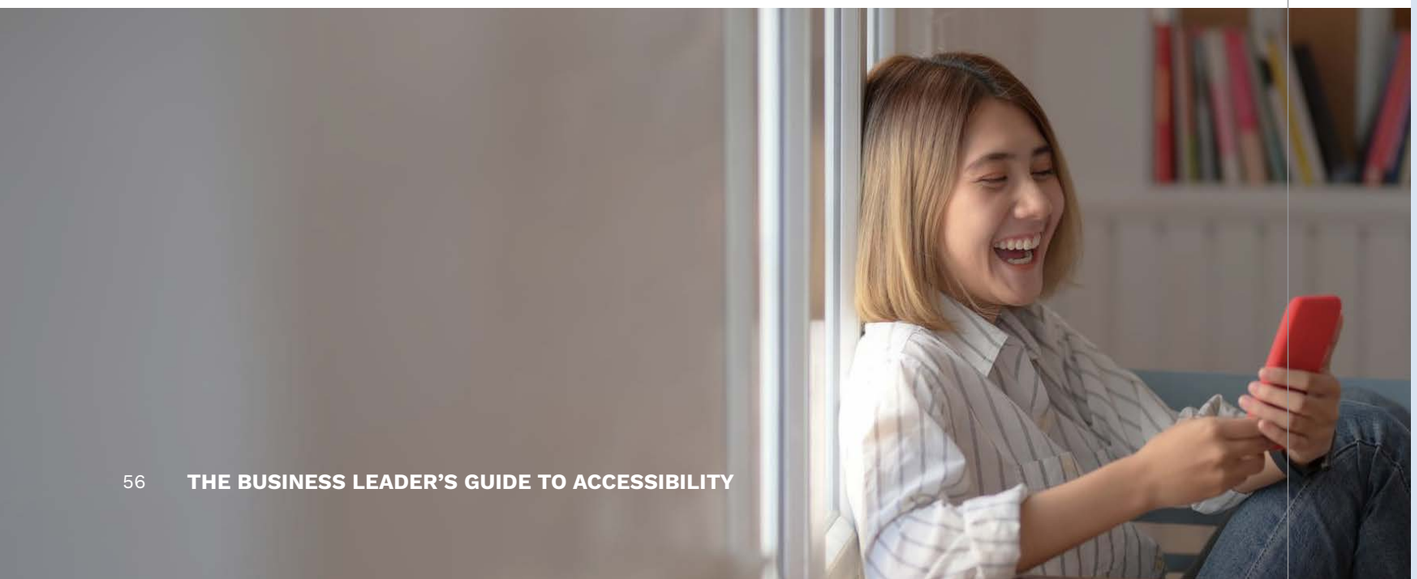
(Urscheler, 2018)

The user receives a status message that, if received while a screen reader is enabled, would be announced by the screen reader. Status messages can appear dynamically at the beginning or end of a page, or even in a modal.

Other examples of status messages include updates to the number of items in a shopping cart, the number of search results returned, and progress bars.

TO MEET THIS CRITERION:

- Ensure actionable functionalities that have a dynamic status receive a status message after taking an action.
- Ensure that confirmation messages are clearly visible, distinguished as status messages, and meet the minimum contrast ratio.
- Ensure the status message is immediately announced by the screen reader after it appears.



IN CONCLUSION

Going beyond A and AA

Now that you've learned how to meet A and AA conformance, you might be thinking, *"What about AAA?"*

To an extent, AAA compliance builds off of A and AA criteria. Where AA is considered acceptable (this is the level required of government sites), AAA is considered the gold standard. It means every single element of your website is 100% accessible to 100% of users.

That being said, the road to AAA compliance is more complex and more customized than the road to A or AA. In some instances, it may not be possible to satisfy all of the Success Criteria outlined for AAA compliance. Because of this, The World Wide Web Consortium (W3C) recommends that AAA compliance never be required by policy.

If you are interested in AAA standards, it's a great idea to consult an agency with experience in crafting compliant sites. A practiced agency team will create and execute a customized plan for bringing your site to AAA compliance.

Wrapping Up

By 2030, an estimated **2 billion or more people will require assistive technology.**[6]

Digital inclusion means providing **equal access** to users of all abilities, and accessible design improves the user experience for everyone.

In the grand scheme of things, what matters most is not the level of compliance you achieve, but **creating the best website possible for all of your users.**

6 Bureau of Internet Accessibility

Are you wondering where your app or website stands when it comes to accessibility?

Are you ready to start the journey towards compliance?

Connect with us today at

mindgrub.com/contact

ABOUT THE AUTHOR

Sandra Koranteng is a Quality Assurance Tester at Mindgrub with a passion for making the web a better, more accessible place. Sandra has extensive experience in accessibility and usability testing and has helped fortune 100 utility companies, state government agencies, and multiple accredited universities achieve compliance.

Sandra has been invited to speak at numerous events including the Virtual Reality and Healthcare Symposium, and she is a member of XR Access, a group of tech industry professionals committed to making virtual, augmented, and mixed reality accessible to individuals with disabilities.

mindgrub

Mindgrub, a Clutch Top 1000 Agency and a member of the Inc. 5000 for seven years running, is a full-service digital agency & consultancy that specializes in crafting award-winning mobile, web, and marketing solutions.

Founded in 2002, Mindgrub has led digital transformations for clients such as Exelon, Under Armour, NASA, GEICO, University of Maryland, Yamaha, Crayola, and Sylvan Learning. Mindgrub's work has been featured at conferences such as SXSW, Adobe Max, and Mobile World Congress, and has been profiled in The Huffington Post, CNN, and Newsweek.

Connect with Mindgrub online at www.mindgrub.com.

REFERENCES

Baskanderi, N. (21 Mar 2017). “Best Practices for Modals / Overlays / Dialog Windows”. *UX Planet*. Retrieved from https://uxplanet.org/best-practices-for-modals-overlays-dialog-windows-c00c66cddd8c

Bhujbal, S. (10 Mar 2017). “Session Expiration Time out Popup using jquery”. *Code Project*. Retrieved from https://www.codeproject.com/Tips/1175658/Session-Expiration-Popup

Bouchard, A. (7 Feb 2019). “LockDroid gives your jailbroken iPhone the classic Android-style matrix passcode screen”. *iDB*. Retrieved from https://www.idownloadblog.com/2019/02/07/lockdroid/

Choudhary, H. (28 Jul 20150. “Radio buttons or Toggle switch for one out of two options”? *UX Stack Overflow*. Retrieved from https://ux.stackexchange.com/questions/82069/radio-buttons-or-toggle-switch-for-one-out-of-two-options

Harsel, L. (1 May 2017). “How to Write Killer Page Titles for SEO [Step-by-Step Guide]”. *SEMRush Blog*. Retrieved from https://www.semrush.com/blog/on-page-seo-basics-page-titles/

Iconsdb. Retrieved from https://www.iconsdb.com/

Mindgrub. (2019).

Shankland, S. (26 jan 2012). “Chrome lets Web pages use smarter autofill technology”. *Cnet*. Retrieved from https://www.cnet.com/news/chrome-lets-web-pages-use-smarter-autofill-technology/

Stanton, P. (17 May 2018). “Which accessibility testing tool should you use”? *Medium*. Retrieved from https://medium.com/pulsar/which-accessibility-testing-tool-should-you-use-e5990e6ef0a

Starbucks (2019). Retrieved from https://www.starbucks.com/

“Web Content Accessibility Guidelines (WCAG) 2.1”. (05 Jun 2018). *W3C*. Retrieved from https://www.w3.org/TR/WCAG21/

Urscheler, R. (11 Apr 2018). “How to add link to sent email message in Gmail Add-On notification”. *StackOverflow*. Retrieved from https://stackoverflow.com/questions/49760370/how-to-add-link-to-sent-email-message-in-gmail-add-on-notification

“What’s New in WCAG 2.1” (8 Apr 2019). *W3C*. Retrieved from https://www.w3.org/WAI/standards-guidelines/wcag/new-in-21/

Ward, C. (10 Jun 2013). “How can I stop my android phone from rotating?”. *Clove*. Retrieved from http://blog.clove.co.uk/2013/06/10/how-can-i-stop-my-android-phone-from-rotating/

Wild, G. (n.d.) “Links and accessibility”. *AccessibilityOz*. Retrieved from https://www.accessibilityoz.com/2014/02/links-and-accessibility/

“1.3.3 Sensory Characteristics”. (2020). *National Enterprise Architecture*. Retrieved from http://www.nea.gov.bh/HTML/WAS/1.3.3%20Sensory%20Characteristics.html