eCampusOntario's Digital Accessibility Toolkit

ECAMPUSONTARIO'S DIGITAL ACCESSIBILITY TOOLKIT

CARA WILKIE; JOHN MCNABB; NATALIE ROSE; OSKAR WESTIN; PIERRE DUEZ; AND PIA DIMAYUGA

eCampus Ontario

This project is made possible with funding by the Government of Ontario and through eCampusOntario's support of the Virtual Learning Strategy. To learn more about the Virtual Learning Strategy visit: https://vls.ecampusontario.ca.

CONTENTS

Part I. Definitions

Part II. Standards

Part III. Development and Procurement

Part IV. Testing

Part V. Glossary

WELCOME TO THE TOOLKIT

About eCampusOntario

eCampusOntario is a provincially funded non-profit organization that leads a consortium of the province's publicly funded colleges, universities, and Indigenous institutes. We develop the platforms, tools, and research that advance the use of education technology and digital learning environments in support of lifelong learning.

Accessible Educational Technology

The use of technology has expanded access to education and has given educators new teaching methods and tools. However, the increasing use of technology can create **barriers** for people with disabilities. It is essential to make sure that technology used for education is accessible to all users, including people with disabilities.

Identifying and removing barriers proactively and providing equitable teaching and learning practices ensures that people with a range of abilities and disabilities benefit without judgement. Following these best practices is the right thing to do. Educators and post-secondary institutions also have legal obligations to accessibility under the Ontario Human Rights Code and the Accessibility for Ontarians with Disabilities Act.

eCampusOntario developed this toolkit on accessible educational technology as a first step towards supporting the sector to enhance how accessibility is considered in educational technology.

A high level overview of this toolkit can be found in the Knowledge translation tool

This toolkit covers :

- What is accessible technology, and how can technology create barriers for people with disabilities.

- Legal requirements and international guidance on accessible digital technology.
- Incorporating accessibility considerations into development and procurement processes.
- Testing for accessibility.
- Additional resources.

PART I

DEFINITIONS

Students and educators in Ontario interact with technology in a variety of ways. Because technology can create barriers for some people with disabilities, it is important to make sure that technology is accessible to all users, including people with disabilities. This means identifying and removing barriers for people with disabilities proactively and make sure that teaching and learning practices are accessible and equitable to people with a range of abilities and disabilities.

Covered In This Section:

- Introduction to Disability
- Disability and Technology
- Incorporating Digital Accessibility
- Using Personas

1.

INTRODUCTION TO DISABILITY

People often think of **disability** as it relates to an individual person and their specific impairments. However, the social model of disability helps us understand that the source of barriers for people with disabilities comes from society and the inaccessible environments it creates. The barrier is not that a person uses a wheelchair; the barrier is the building without a ramp. The barrier is not that a person cannot hear audio information; the barrier is information that is only communicated in an audio format. Under the social model of disability, we understand that the responsibility to remove barriers for people with disabilities lies with society, rather than asking people with disabilities to change or adapt to inaccessible environments.

Who experiences barriers to technology?

Many people with and without disabilities experience barriers to technology use. Sometimes those barriers are financial, such as a student who cannot afford to purchase a particular piece of software. Sometimes the barriers are knowledge based, such as a mature student who did not grow up with computer technology.

For people with disabilities, the barriers they face often come from the technology itself, which is rarely designed with a wide variety of users in mind. Most designers build products based around the idea of a "typical" or "average" user. This approach to design does not account for the variety of individuals who may encounter or use the technology in alternative ways. For example, technology may not consider a person who accesses web information by having it read by a screen reader rather than viewed. People with different types of disabilities experience barriers to technology in different ways.

Physical/Dexterity Disabilities: People who have physical disabilities that affect dexterity and fine motor skills may experience barriers trying to interact with a mouse, keyboard, or touch screen in traditional ways. For example, small webpage buttons may cause barriers for people with disabilities affecting their manual dexterity.

Disabilities that Impact Vision: People with visual impairments may have difficulty seeing the equipment they are using (such as a keyboard) and may have difficulty seeing the information that is displayed on a screen. Many people with vision disabilities have some sight, and so they may experience barriers when they encounter small text, crowded graphics, or poor colour contrast. For example, webpages where the headings, column and row headers, and links are not properly coded will create barriers for people who rely on screen readers.

Disabilities that Impact Hearing: People with hearing impairments will encounter barriers whenever information is presented through sound. For example, videos, audio clips, or sound effects that have not been created or designed in accessible ways can pose barriers for people with hearing impairments.

Disabilities that Impact Comprehension: People with disabilities that impact comprehension may experience barriers with understanding information or navigating around websites or apps. Barriers for people can include how information is presented visually, the type of language that is used, or how “intuitive” the app or website is to use. For example, websites with unclear navigation can create barriers for people with some cognitive and learning disabilities.

Other disabilities may impact executive function and social interaction. Each person’s situation is unique. Not every person with the same category of disability will experience barriers in the same way, and some people have more than one disability. Therefore, when planning to make technology accessible, consider a wide variety of users and attempt to remove as many barriers as possible.

2.

DISABILITY AND TECHNOLOGY

You might think of **disability** as it relates to an individual person and their specific impairments. However, the social model of disability helps us understand that the source of barriers for people with disabilities actually comes from society and the inaccessible environments it creates. The responsibility to remove barriers for people with disabilities is ours. Our goal is to proactively identify and remove or reduce barriers for people with disabilities.

Accessibility and technology

Something can be considered accessible when it provides equitable access to allow people with disabilities to complete tasks and make choices without judgment, while retaining their autonomy. Digital accessibility, sometimes referred to as ***a11y*** (where "11" refers to the number of letters omitted), provides equitable access for people who may need Assistive Technology (AT) or personal **accommodation** to complete digital tasks. Accommodations may include modifications to the digital material, such as adjustable screen brightness or font size. ATs can include specialized hardware or software that can assist people with disabilities to perceive and interact with digital content. Examples of ATs are provided below.

Accessibility APIs and assistive technologies

Digital devices, including laptops, desktops, tablets, and other mobile devices, have a built-in Accessibility Application Program Interface (**API**). APIs effectively serve as a translator or guide between different software **components**. Software applications use the rules set out in the accessibility API to communicate to Assistive Technologies (ATs) and ensure that the ATs can interpret the information from the software so that it is understandable for the user.

The interaction can be broken down as follows:

1. Browsers translate web-based applications to an accessibility tree, which speaks to the accessibility API.
2. The assistive technology receives the data and translates it into an alternate user interface.
3. The actions of the assistive technology user are translated back through the Accessibility API to complete desired tasks.

Examples of Accessibility Features and Technology

Screen Readers: Screen readers allow people who have difficulty seeing or reading to access information. The screen reader will identify text in a website, app, or document and read it out loud with an electronic voice to the user. Screen readers are Artificial Intelligence (AI) and so they do not intuitively understand what order text should be read in and rely on features like headers and other clues built into documents, websites, and applications for context.

Switches: A switch is a mechanical aid that allows a person with a physical disability to access the features of a keyboard or mouse in an alternative way. There are various types of switches, from tactile buttons, pedals, levers, or breath-activated tubes. **Switch controls** perform a single action, like the click of a mouse. When a user has access to one switch, software cycles through options on a website, application or virtual keyboard and allows the user to activate the features they want.

Speech-to-Text Technology: **Speech-to-text** technology allows users to bypass traditional typing on a keyboard by speaking. The computer or device recognizes what the user has said and translates the spoken words to commands or text (For example, if the user said: “double-click file name”). People with different types of disabilities, including physical disabilities, vision disabilities, or cognitive disabilities, may use speech-to-text applications. This technology is becoming popular with non-disabled people as well because it is a convenient feature that can save time. Many devices, computers, and applications come with built-in speech-to-text technology.

3.

INCORPORATING DIGITAL ACCESSIBILITY

The need for digital **accessibility** arises with all digital technology that people interact with, including websites, mobile apps, software, audio and video files, documents, and authoring tools. The content and delivery mechanism for these assets can create barriers for people with disabilities if they are not accessible.

The objective of creating an accessible product is to provide an equivalent user experience for everyone. The principle behind **inclusive design** is that by providing accessible content for people who face barriers, you are also creating a better experience for everyone. In other words, creating accessible solutions needed by some will benefit many. Improving accessibility involves identifying barriers that prevent people with disabilities from accessing content and services. Improving digital accessibility reduces barriers and provides equitable access to digital content for persons with disabilities and people who use **Assistive Technology** (AT).

Planning for Digital Accessibility

It is crucial to consider accessibility before you create new content or procure a new technology. Good accessibility does not happen by accident. Making intentional, inclusive decisions during the planning stages will not only benefit your end users; it will also reduce effort, long-term costs, and litigation risk.

Procuring Accessible Digital Technology

Third-party software, digital solutions, or development can create barriers if they are not accessible. Including consistent and clear accessibility requirements during the procurement process will reduce risk for you and remove barriers for your end users. See the procurement section for more guidance.

Software and Website Development

Accessibility is essential throughout the development process, from the **component** level to the overall system design. Planning for accessibility will ensure consistency at the system level, both within and between pages. Using proper landmarks and consistent navigation at the outset will make it easier to design and develop an accessible application.

Digital accessibility does not stop at the planning stage, as with other design criteria. Incorporate testing to ensure that you meet targets before launch. See Testing for more information.

Resources :

You can get more information and resources on digital accessibility through these pages and the following sources:

- The World Wide Web Consortium (W3C) provides many strategies and recommendations that will help you to create robust and accessible experiences on the web, including the Web Content Accessibility Guidelines (WCAG) is an excellent resource
- The Government of Canada has a series of resources on creating digital documents, including in Word, Excel, PowerPoint, Visio and PDF.

More resources are available on our full Resource List.

4.

USING PERSONAS

One way to improve accessibility and the end user experience more broadly is by developing **personas**. Personas are fictional user-types created to represent real people that might use your service, product, site. Creating multiple and diverse personas can help you recognize different needs and expectations of potential users.

In the specific context of improving accessibility, there are two types of personas you can consider developing: personas that represent people with disabilities, and personas without disabilities who may need or benefit from more accessible experiences. Considering disability is an excellent way to help create inclusive and accessible products.

Persons with disabilities as persona examples

Here are some examples of personas that you could use:

- A student who is blind wants to learn from the online course material. They depend on a keyboard and screen reader to review content and complete tasks.
- A student with low vision is watching a slide presentation. They rely on zoom features and good colour contrast to perceive content.
- A student with learning disabilities depends on visual aids, simplified text, and uses **text-to-speech** tools to comprehend online content.
- A student with impaired motor skills needs to complete an online quiz. They rely on their keyboard or **speech-to-text** software to complete the test.

Personas and accessible experiences

You can also add accessible experiences to pre-existing personas. Modifying a persona to include a temporary disability, like a broken arm, will help your designers consider how they might navigate software without using a mouse. Having your persona complete homework outside introduces the need for good contrast. If your persona lives in a noisy household, captions make it easier to consume pre-recorded lecture content.

Accessible considerations

The following list of sample solutions can give you an idea of how to overcome barriers your personas might experience. The Government of Canada also provides detailed personas that can be useful.

Sensory disability (vision)

- Good contrast
- Colour selection
- Readable font size
- Software and hardware with adjustable display settings (e.g., magnification)
- Electronic documents with accessibility features
- Applications that support all browsers and operating system accessibility features
- Reduced motion options
- Time-based media (audio/video) is available in text format
- Ability to control time-based media (e.g., pause, slow down)

Sensory disability (hearing)

- Audio signal/alert with supporting visual cues
- Hardware and software with adjustable audio settings
- Hardware that is compatible with hearing devices
- Time-based media that includes captions and text-based alternatives like transcripts

Physical disability

- Alternative controls for complex interactions requiring two hands
- Keyboard or **switch** compatibility
- Alternate solutions for complex patterns like click and drag
- Large target areas for functional elements (i.e. does not require fine motor control and precise actions)

Cognitive, learning disability, or language barriers

- Use plain language
- Choose readable fonts
- Avoid unnecessary and repetitive animation or blinking text
- Remove persistent sound effects
- Use recognizable design patterns and icons with visible text labels

Resources :

The UK Home Office has a series of informational posters summarizing considerations for accessible design. This content is also available as a series of HTML pages:

- Designing for users with low vision: (image) (html)
- Designing for users of screen readers: (image) (html)
- Designing for users who are D/deaf or hard of hearing: (image) (html)
- Designing for users with physical or motor disabilities: (image) (html)
- Designing for users with dyslexia: (image) (html)
- Designing for users on the autism spectrum: (image) (html)

PART II
STANDARDS

Colleges and universities are committed to **accessibility** for their learners and staff. They also have legal requirements to around accessibility in websites, learning materials, and in purchasing under the Accessibility for Ontarians with Disabilities Act and the Ontario Human Rights Code. Digital accessibility standards help to guide the development of accessible websites and technology and are all based on similar key principles. WCAG, a global standard of website accessibility, is the most common standard. In addition, the United States and European Union have developed accessibility standards that are useful for Canadians to rely on. The Ontario requirements may be changing after the recommendations of two Standards Development Committees and to keep pace with other international standards.

Covered In This Section:

- Ontario Legal Requirements
- International Standards
    - Principles
    - Beyond Websites
- If Standards Can't be Met

5.

ONTARIO LEGAL REQUIREMENTS

The United Nations Convention on the Rights of Persons with Disabilities recognizes accessibility of information and communication technology as a human right. There are two main laws about accessibility in Ontario: the Ontario Human Rights Code OHRC and the Accessibility for Ontarians with Disabilities Act (AODA).

The AODA and OHRC work together. The AODA sets specific accessibility requirements. The OHRC also requiring organizations to respond to individual requests and not discriminate. Keep both laws in mind because if there is a conflict between them, the law that sets the higher level of accessibility will prevail (AODA, s. 38).

Integrated Accessibility Standards Regulation and AODA

In Ontario, the Integrated Accessibility Standards Regulation (**IASR**) of the AODA sets legal requirements for organizations in the "broader public sector" and to "educational or training institutions."

The following AODA requirements relate to the technology of colleges and universities:

- They must think about accessibility when buying goods and services. Section 5 of the AODA says that organizations must "incorporate accessibility design, criteria and features when procuring or acquiring goods, services or facilities, except where it is not practicable to do so."
- Websites that the public can access must conform with WCAG 2.0 Level AA. This is interpreted to include applications and social media accounts.
- When someone asks for it, they must give educational or training resources or materials in an accessible format that meets the needs of the person with a disability. This must be done by getting a **conversion-ready** electronic format or arranging to give comparable resources in an accessible format. Conversion-ready means information in an electronic format that is easily converted into an accessible format (e.g. HTML and structured Word files).
- Organizations that produce educational or training textbooks must make accessible or conversion-ready versions available.
- Libraries of educational institutions must provide conversion-ready formats upon request.
- Report on how they are following AODA requirements by December 31, 2021.

By law, “conversion-ready” means that an electronic or digital format that facilitates conversion into an accessible format. For example, a conversion-ready document might be in Microsoft Word or HTML.

There are some exceptions to these general obligations, such as:

- Where it is not practicable to incorporate accessibility criteria into procurement
- Information that the institution does not control directly or indirectly
- Special collections and rare books

OHRC

The OHRC and human rights codes in other provinces set complementary obligations. Colleges and universities must reasonably accommodate the needs of people with disabilities unless the college or university would experience “undue hardship”. The OHRC often arises when someone asks for an **accommodation** (e.g., a student requests a learning resource in an accessible format). The college or university must consider the request and grant it if reasonable.

The OHRC has also been legally interpreted to require an organization to think about an accessibility need in advance. For example, in the legal case of Lepofsky v. Toronto Transit Commission, the Ontario Human Rights Tribunal found that the TTC had breached the OHRC by failing to implement audible stop announcements.

The challenge in interpreting the OHRC is that it does not say what exact level of accessibility is needed. Analysis of the exact limits of the OHRC obligations would necessitate independent legal advice. However, the OHRC obligations go beyond those that are listed in the AODA.

Future Legal Obligations

In late 2019, Ontario’s Information and Communications Standard Development Committee (I&C SDC) released Interim Recommendations, which includes recommendations for a first and second phase of revisions to the IASR. The Postsecondary Education Standards Development Committee (PE SDC) released initial recommendations in June 2021 for feedback from the public. None of these recommendations have yet been accepted or implemented into regulation.

The recommended changes that are relevant the technology of colleges and universities include:

- Broadening the definition of “website” to include “both mobile applications which run from a website, and those which run as a standalone device but rely on the internet for function.”
- Providing more specifics about how to incorporate accessibility design, criteria, and features in procurement. However, it does not say what accessible design looks like.

- Requiring all materials, assessments, and multimedia created or sourced by faculty to be available in multiple accessible formats.
- Technology used in digital learning must be accessible or a functionally usable alternative option must be provided.
- Developing a plan to vet technology, seamlessly include accessibility in digital learning and consult with stakeholders about this plan.
- Identify in advance and communicate to students the accessibility features of digital technology and learning components needed for each course.
- Having a digital technology accessibility plan.
- Appointing a high-ranking employee as a digital accessibility technology lead.

Resources :

- The Ontario Human Rights Code
- The AODA
- The IASR
- Information about Accessibility Compliance Reports
- Information and Communications Standards Development Committee, Final report
- Post-secondary Education Standard Development Committee, Initial Recommendations Report

6.

INTERNATIONAL STANDARDS

There are a few different international standards of accessibility in technology. WCAG, Section 508, and ETSI 301 549 are the most prominent and work well together.

Web Content Accessibility Guidelines (WCAG)

The World Wide Web Consortium (W3C) and Web Accessibility Initiative (WAI) are responsible for the creation, maintenance, and updates for the Web Content Accessibility Guidelines (WCAG). WCAG provides recommendations and success criteria to help create a more inclusive and accessible internet. The **AODA** currently relies on WCAG 2.0 Level AA conformance for publicly-facing websites. The active version for WCAG is 2.1 (released in 2018). Version 2.2 was in development at the time of writing. Most legislative bodies refer to WCAG 2.0 AA as the current requirement for all website content.

WCAG is informative. However, creating and delivering accessible content goes beyond meeting WCAG criteria. Since WCAG and technology are continually evolving, checking off the list of criteria today may not address the accessibility guidelines of the future. Its best to view WCAG as a helpful resource, instead of a mandatory checklist.

WCAG can also be extrapolated to non-web information and communications technology (see Guidance on Applying WCAG 2.0 to Non-Web Information and Communications Technologies, as an example).

Other International Standards

Other international standards have been developed for public procurement or publicly funded procurement.

The **ETSI** (European Telecommunications Standards Institute) EN 301 549 standards, issued by the European Commission, define functional accessibility requirements that can apply to different information and communications technology (ICT) products and services. It defines accessibility needs for people with different impairments and includes WCAG requirements. The Government of Canada has developed tools to support the definition of procurement specifications **ICT** aligned with ETSI EN 301 549 (for example, see the ICT accessibility requirements wizard).

Similarly, the Section 508 rule legal standards (from the US Rehabilitation Act) are utilized by American purchasers. These standards are required for many publicly-funded procurements in the United States. These standards are broadly harmonized with other standards, including the ETSI EN 301 549 standards and WCAG.

PRINCIPLES

All international accessibility standards are aimed at increasing accessibility in broad and specific contexts. So, it is not surprising that while there are differences among the standards, they all have similar underlying principles. Focusing on the Web Content Accessibility Guidelines (WCAG), there are four underlying principles.

WCAG Principles

The acronym for the four WCAG principles is **POUR**:

- **Perceivable**: Information and user interface **components** must be presented to users in ways they can perceive. Perception, in this case, goes beyond sight and includes considering other ways people can recognize and use your content (e.g., audio cues).
- **Operable:** User interface components and navigation must be operable by a broad audience. Not everyone uses a keyboard and mouse, and many devices require touch technology. **Assistive technologies**, such as screen readers, enable users to interact with your content in a way that works for them.
- **Understandable:** All aspects of your content, from the information itself to the operation of the functional user interface, must be easy to understand. Clear labels and recognizable design patterns reduce the effort in learning how to use and interact with your content.
- **Robust:** Content must be established and reliable enough to be interpreted consistently by as many browsers and assistive technologies as possible. A robust application is stable and future friendly.

Other international standards (such as ETSI 301 549) incorporate WCAG standards and add additional principles and consideration that are applicable to software and hardware, or other information and communications technology.

WCAG Versions and Levels

Three levels prioritize the success criteria, Levels A, AA and AAA:

- Level A is the lowest level required to meet success criteria. Applications meeting Level A **conformance** alone are not accessible. However, these are the most important success criteria to achieve.
- Double A (AA) includes Level A criteria. To be AA conformant, you would need to exceed Level A and meet Double A techniques. These are the second most important criteria.
- Triple A (AAA) A is the highest use of WCAG success criteria. Meeting AAA can be challenging and sometimes impossible for some technologies.

It's common to aim for Level AA conformance, because this provides a good level of accessibility, and it's normally achievable with most content. It is also reflective of **AODA** website standards.

Remember, meeting all Website Accessibility Guideline Success Criteria does not necessarily make your product accessible.

BEYOND WEBSITES

Websites vs Desktop Applications

While the Website Content Accessibility Guidelines (WCAG) were initially developed for websites, the principles of WCAG can be applied to desktop applications too. As an example, the **ETSI** 301 549 standards (the European accessibility standards) build off and reference WCAG. The core principles are similar, and some WCAG techniques can apply to software applications. (See Guidance on Applying WCAG 2.0 to Non-Web Information and Communications Technologies.)

Note that one important difference is that automated testing is limited for desktop accessibility. Please refer to the sections on **Testing** and on **Accessibility Insights.**

Functional Accessibility Requirements (FARs)

FARs are part of ETSI 301 549. Government of Ontario's Information and Communications Standards Development Committee recommended adopting a set of FARs in its review of the Information and Communications Standards. These recommendations have not been adopted into law.

FARs describe the functionality that must be available to ensure information and communications are more accessible to people with disabilities. They do not define the exact method of achieving that functionality.

For example, when visual information is provided, the FARs require that:

- There is a way to perceive and understand the information without vision
- The presentation must be adjustable to support limited vision and/or visual perception or processing (magnification, contrast, spacing, visual emphasis, layout)
- There is a way to perceive and understand the information without relying on colour distinction
- The presentation must avoid characteristics that would trigger photosensitive seizures
- Making alternative formats of the presentation must be possible, such as tactile formats

Similar FARs are written with different disabilities and impairments in mind, detailing requirements for "where speech is required to operate a function" and "where operation has time limits".

The full list of recommended FARs are in the Final Report of the Information and Communications Standards Development Committee

7.

IF STANDARDS CAN'T BE MET

In some cases, it can be challenging, if not impossible, to meet the accessibility standards or regulations. For example, legacy or third-party software may not have the robust functionality to create inclusive experiences. Understanding gaps in accessibility and taking responsibility for providing a more inclusive future reduces your reputational risk and mitigates the risk of litigation.

The first step to address such a situation is to reach out to your internal developers, partners, or vendors. You can identify the accessibility limitations and ask for a solution. Sometimes there is a straight-forward solution that can be implemented. Other times, there may not be an immediate solution. However, a solution could be included in a roadmap and timelines outlined on when their products will become more accessible.

If accessibility gaps cannot be avoided, consider formally documenting the gap, including noting:

- The specific accessibility gap(s) and **barrier(s)**
- Why it was not practicable to incorporate accessibility design, criteria, and features
- Your plan to reduce or eliminate these barriers

The Integrated Accessibility Standards Regulation (**IASR**) of the Accessibility for Ontarians with Disabilities Act requires organizations to, upon request, provide an explanation about why accessibility criteria were not part of procurement. This documentation will help you respond to such a request.

Remember to consider the end user experience over all other decisions, even if your code cannot meet all accessibility specifications. Keeping your audience in mind will help you provide the best experience possible.

PART III

DEVELOPMENT AND PROCUREMENT

All educational technology is either developed or purchased. This section includes guidance about how to set your institution’s accessibility standard and how to incorporate accessibility into development and procurement processes.

Covered In This Section:

- Setting Accessibility Standards
- Accessible Development
    - Intro to HTML and accessibility
    - Intro to ARIA
    - Accessible Templates
    - Testing Throughout Development
    - Deviating From Standards
    - Desktop vs Mobile Content
    - Beyond HTML: Accessible Applications
    - Document Accessibility
    - Accessibility Statements
- Accessible Procurement
    - Organizing the Procurement
    - Determining Specifications
    - Incorporating Specifications Into Solicitation
    - Interpreting Vendor Claims
    - Evaluation and Non-Conformance

- Contract Language

8.

SETTING ACCESSIBILITY STANDARDS

Each post-secondary institution sets its standard of digital **accessibility** independently. You could aim for legal **compliance** (**WCAG** 2.0 Level AA for public websites) or could adopt a more accessible standard. Regardless of what standard of accessibility you adopt, communicate that standard internally. While there are a number of resources and guidelines on digital accessibility, remember that an accessible product is one that your staff and students can use.

Resources to Consider when Setting Specifications

It may seem daunting to set digital accessibility specifications, but don't worry. In short, answer two questions: What is your standard of accessibility? And what technology does that standard apply to (e.g. websites; desktop applications; hardware).

Here are some considerations when setting your specifications:

- It is essential to understand the legislative requirements when establishing your standards. Web Content Accessibility Guidelines (WCAG) 2.0 Level AA, is the minimum standard required in Ontario. While this requirement only applies to publicly available websites, institutions could adopt a more accessible standard or apply the standard to more technology.
- The legal requirements may change in the future. A standards development committee recommended adopting a set of Functional Accessibility Requirements (**FARs**) as the standard of accessibility under Ontario law. These recommendations would apply to all technology that uses the internet to function. See the Legal Requirements page for more information.
- Specifications for web-based solutions typically include WCAG as a resource to meet an expected inclusive experience, complemented by an understanding of **inclusive design** principles, technology **barriers**, and common accessibility **personas**. Consider what level of WCAG you want to strive for. See the International Standards page for more information.
- For non-web content and technology, the WCAG principles or the **ETSI** (European Telecommunications Standards Institute) EN 301 549 standards could be used.

Resources :

- Guidance on Applying WCAG 2.0 to Non-Web Information and Communications Technologies (WCAG2ICT)
- The Government of Canada tool to support the definition of procurement specifications aligned with ETSI EN 301 549.
- Beyond Websites page of this tool.
- Accessibility Statement page of this tool

9.

ACCESSIBLE DEVELOPMENT

How to Build Accessibility Into the Development Process

As with any fundamental design constraint, accessibility is best included – and tested – at every stage of development. Good accessibility does not happen by accident. Building on a foundation of good accessibility will guide design decisions throughout the development process, with resulting impacts on visual presentation (e.g., colour contrast and layout) and overall flow and operation (e.g., navigation and wayfinding).

Accessibility standards

It is important to understand digital accessibility standards and how they apply to the work that you are going to do (e.g., website development, updates, app development). In addition to understanding digital accessibility principles in general, be familiar with your institution's accessibility standards. Being familiar with these will allow you to integrate them into your guidelines and your general development process.

Institutional standards often represent a minimum level of accessibility, sometimes based on legal requirements. Digital accessibility and related guidelines (like WCAG) are continuously updating, reflecting a continuous process of improving accessibility, and in reaction to advancements in technology and web content development. As a result, you may feel that it is practical and forward-thinking to design to a higher level of accessibility than what is represented in your institution's accessibility standards. In any case, it is essential to establish specific success criteria for accessibility design at the earliest stages of development.

Accessibility throughout development

Accessible design is best planned, implemented, and tested at every level of design. We have taken a generally hierarchical design model, with component, integration (or sub-system), and system levels of design. Simpler systems might not have meaningful distinctions between these levels; larger projects might have more levels of hierarchy. The principles and concepts discussed here can broadly be applied to both simpler and larger projects.

Broadly speaking, development has three levels:

- **Component Level** refers to individual **widgets** or controls, or **components** on a single webpage.
- **Integration Level** is an intermediate level of aggregation, looking at an entire webpage, or a single screen in an application.
- **System Level** refers to the entire system being developed – the whole website or application.

Component Level

At the component level, any control presented to the user needs to be accessible to alternative forms of access. For example, a user needs to be able to use their keyboard to switch between controls, as well as to operate them, unless there is a just reason not to allow it (e.g., a resource testing mousing accuracy). Most standard User Interface (UI) libraries offer keyboard-only functionality, with the ability to Tab between fields, buttons, etc. and the use of the keyboard to select from options, enter text and so forth. If your application requires a new type of control, implement keyboard access as well to remain keyboard accessible.

WCAG offers several guidelines for alternative access to controls, both for reading content and interacting with it. Reading content can include reading the current value of a numerical slider, or alternative text descriptions for video, image, or audio content. Most accessibility standards also describe important criteria for interacting with a control (button, interactive graph, toggle, etc.), including consistent keyboard behaviour.

Integration or Sub-System Level

At the integration level (or subsystem level, or webpage level), accessible design planning includes the flow of focus between controls on the page, including tab order and the logical flow of text and controls.

- **Flow of Focus:** One consideration is how screen readers will interpret the flow of the page. Screen readers do their best to interpret screen content logically, but do not always get it right. As a result, test to ensure that the order in which elements are read (including alternative descriptions) are consistent with your design goals.
- **Flow of Controls:** Test flow of control and actions within a single page or subsystem too. Make finality clear to the user if irreversible actions can take place (for example, paying bills in a banking application). Where possible, present the user with a final summary of the transaction, and given the opportunity to

accept, modify, or cancel the action.

At this level of development, consider navigation and waypoints. Make UI elements used to navigate the application (or webpages within a website) identifiable and navigable by alternative means such as keyboard access. Where possible, establish a consistent design so that similar actions function similarly throughout an application. This consistency will create a consistent "look and feel" for end users and increase overall **usability**.

System Level

At the system level, all the design elements and accessibility criteria work together cohesively to create a smooth, consistent user experience. For example, make waypoints implemented at the integration level consistent among all pages.

This level is the final check for success at the lower levels of integration. Navigation blocks should be universal among sub-systems (or webpages within a website) for easy wayfinding. This includes consistent wording, interaction, and placement. Ensure that there are no navigation dead-ends (a page that leads to no other pages) and that the flow between pages is understandable and predictable.

INTRO TO HTML AND ACCESSIBILITY

Hyper**T**ext **M**arkup **L**anguage (HTML) is the markup language (code) used to structure a web page and its content. Since it is a language that helps define a website, it significantly contributes to whether the website is accessible or not. HTML provides expected patterns for reading or navigating content, which the **accessibility** of a user experience is dependent upon. The content structure of HTML includes headings, paragraphs, bulleted lists, images, and data tables. HTML allows you to use tags to markup your elements.

There are different subsets of tags in HTML, one of them being *semantic HTML*. Semantic HTML can significantly contribute to accessibility. In HTML, the term "semantics" refers to the meaning of the code. Semantic HTML includes tags whose purpose is defined by the element. For example, an `<h1>` tag represents the top-level heading on your page (heading one) and an `<li>` tag identifies an item in a list. In contrast, the `<div>` tag does not have any semantic meaning and therefore does not provide meaningful information from an accessibility perspective. For example, it's more effective to style a `<button>` to look a certain way than to create a button with a `<div>` tag with qualifiers and scripts. Buttons have inherent traits and expected behaviours for mouse clicks, keyboard and tap interactions. Therefore, use semantic HTML wherever possible to contribute to good accessibility. It's always better to use semantic HTML code rather than building something new.

Overall, developing a basic understanding of HTML can go a long way towards building accessible products. Wherever possible, follow HTML best practices. Additionally, you can style HTML with Cascading Style Sheets (CSS) or add JavaScript (JS) for functionality. In general, standard semantic HTML elements are **compliant** with current accessibility standards such as **WCAG**, provided you do not extend or change their functionality too much.

Resources :

- W3C HTML Living Standard
- To learn more about HTML, review Mozilla's Getting Started with HTML

INTRO TO ARIA

HTML (HyperText Markup Language) was initially designed to create static text pages. JavaScript provides dynamic **components** which don't exist in native HTML. The **A**ccessible **R**ich **I**nternet **A**pplications (ARIA) standard provides developers with accessibility features not available in HTML alone.

ARIA is a set of attributes (properties of HTML tags) that makes web content and applications more accessible. When it is not practical to use semantic HTML, ARIA can communicate semantic information and interactions to **assistive technologies**. ARIA communicates semantic information about **widgets** (User Interface (UI) elements), structures (relationships), and behaviours to assistive technology. A proper accessible experience depends on the appropriate use of this tool: misuse of ARIA can lead to poor or unusable experiences.

ARIA attributes can make non-semantic HTML like `<div>`s more accessible when native code or underlying technology cannot. For example, suppose you are using an external library or framework that uses a `<div>` as a button. It may be easier to add an ARIA tag such as '`role=" button"`' to the component, rather than rewriting existing styles and functions. Consider this approach carefully, since you will also need to confirm that there is proper JavaScript in place to make the element accessible (e.g., ensure that it is focusable and define event handlers for click and key-down events).

If you intend to enhance the accessibility of HTML with ARIA tags, please consult the authoring practices and the detailed examples below. To summarize the first rule of ARIA: If you *can* use HTML or semantic attributes, then do so. Another common expression is that good ARIA is better than no ARIA, but no ARIA is better than bad ARIA.

Examples of when to use ARIA

Landmarks

- Landmarks define sections of a web page, which helps **screen reader** users navigate the page. Landmarks can be defined either as HTML elements or with ARIA roles: A `<header>` landmark can be replaced, if necessary, on any element (like a `<div>`) by adding `role="banner"`.
- An `<aside>` landmark can be replaced with `role="complementary"`.
- A `<footer>` landmark can be replaced with `role="contentinfo"`.

ARIA Roles and JavaScript

Some JavaScript frameworks generate web pages with non-semantic code, for example, a `<div>` element instead of a `<button>`. Adding the ARIA `role="button"` will improve provide semantic meaning to the `<div>`.

When JavaScript is used to create special components that don't exist natively in HTML, they may need to be given an ARIA role to define what they are. For example:

- An accordion will be expanded and collapsed with a button which has this attribute: `aria-expanded="true"/"false"`.
- A set of tabs within a web page will need `role="tablist"`, `role="tab"`, and `role="tabpanel"`.

More examples can be found on the ARIA Authoring Practices site.

Labels

There are multiple ways to label a button – besides using onscreen text – for example:

- An aria-label attribute with text specifying the label, i.e., `aria-label="continue`".
- An `aria-labelledby` attribute can point to existing onscreen text as a label.
- An `aria-describedby` attribute can point from an input field to an inline error to connect them in a semantic (meaningful) way. A screen reader will automatically announce the error when the user tabs to the input field.

ARIA live regions

An ARIA live region is a section of a web page that screen readers will automatically announce if its contents change dynamically after the initial page load. The live region provides important status messages. Code examples include:

- `role="alert"` or `aria-live="assertive"` tells the screen reader to interrupt a current message to announce a dynamic content change.
- `aria-live="polite"` tells the screen reader to announce the updated live region after announcing the current content.

aria-hidden

The `aria-hidden="true"` attribute tells screen readers to ignore an element and not announce it. Aria-hidden can be useful for content displayed on-screen but confusing for a screen reader user.

For example:

- An icon placed beside redundant text (like a garbage can icon with the word “Delete” beside it) since a user doesn’t need both to be announced.
- Background content that is greyed out when a popup takes over the page. When a popup takes over the page, a screen reader user won’t be reading the background page.

Resources:

- W3C WAI-ARIA Overview
- WAI-ARIA Authoring Practices 1.2

ACCESSIBLE TEMPLATES

Good accessibility is embedded into each step of the web development process. Designers, developers, and content specialists all have a role to play in developing accessible websites. This page explains how accessible templates can be developed to support this process.

Creating accessible wireframes (designers)

A web page wireframe is a visual design document created by designers. The wireframe shows the layout of a web page by organizing a web page into sections that will be populated later with images and textual content. The wireframe can lay out pages in a way that maximizes **usability** and ease of use. Using a wireframe will ensure that the web pages will have a consistent layout from page to page, improving accessibility for users with cognitive **disabilities**.

Creating accessible templates (developers)

Developers analyze the wireframe document created by designers and create a web page template that is coded in HTML. The code in the template must be accessible to create accessible content pages. The template code can define each section of the page to hold the content that will be supplied later by a content specialist (e.g., header, nav, main, footer, etc.)

The World Wide Web Consortium (W3C) has good resources on how to build an accessible template, including:

- W3C De**W3C**sign System
- ARIA (Accessible Rich Internet Applications) Landmarks Example

Creating accessible content within templates (content

specialists)

While content templates can include some guard rails, like a form within a Content Management System (CMS), website content itself can also create **barriers**. An accessible content template makes it easier for content specialists to produce accessible content. See Document Accessibility for more details.

A content template should allow content specialists to update all content relevant to the web page, including:

- Page `<title>` (this appears in the browser tab)
- Headings and their levels (`h1`, `h2`, `h3`, etc.)
- Paragraphs
- Lists – unordered (bulleted) lists and ordered (numbered) lists
- Images and their alt (alternative) text

How to add content that is accessible

Always try to maintain web page content that is written semantically, i.e., uses standard HTML formatting. Content that is semantic will improve accessibility and works well with **Assistive Technologies** (ATs), like **screen readers** , speech recognition, and even keyboards. Some of the accessibility benefits of content that is formatted this way include:

- The page `<title>` is announced by screen readers when the page loads.
- Headings and their levels (`h1`, `h2`, `h3`, etc.) are announced by screen readers to provide context to users with vision impairments. Headings also provide a way to navigate or scan content on a page.
- Tagged paragraphs improve readability and give users more control when customizing contrast themes.
- Unordered lists (bulleted) and ordered lists (numbered) are announced by screen readers to provide context and to set expectations (how many items are in the list).
- Descriptive (alt) text for images is announced by screen readers to provide meaning for people with vision impairments.

How to request a change to the template

It's important for content specialists to check with the developers or **IT** architect before making updates to the page template or wireframe. If the wireframe changes impact the page's content, the content template may also need changes.

References:

- W3C Content Structure

TESTING THROUGHOUT DEVELOPMENT

Testing is critical to establishing and maintaining the accessibility of a website. Incorporating testing throughout the development process will confirm and validate that accessibility specifications are being met. Designers, content specialists, quality assurance testers, and developers all play a role in testing for accessibility. All aspects of the product, from the coding to content updates, or even changing colours of text or icons can create an unintended accessibility issue.

Over time, changes in **assistive technology** or browser support might also introduce new accessibility problems. Regular updates or monitoring can reduce this risk.

The rest of this page highlights some specific tools that a person in each role may want to consider using.

- **Designers**: Check the colour contrast of your text content or icons. See the section on Colour Contrast Testing.
- **Content Specialists**: Consider the readability of your content and how easy it is to understand. This includes your use of plain language to make your content accessible for people with learning or other cognitive **disabilities** and language barriers. See the subsection on Plain Language.
- **Quality Assurance (QA) Testers:** Keyboard testing and **screen reader** testing ensures that features and workflows are accessible with the keyboard alone (no mouse). See the sections on Keyboard Testing and NVDA Screen Reader Testing.
- **Developers:** Using automation is an excellent way to test for accessibility. Quick fixes can be discovered and prevented by using automated testing tools. Automation does not guarantee accessibility; some manual validation is needed. See the section on Deque axe tools and Accessibility Insights.

Site Monitor Tools

For websites already in production, there are several accessibility dashboard tools that continually monitor the site looking for accessibility issues. These may be particularly appropriate for large organizations with large sites. These are typically paid tools. Some prevalent in the industry include:

- Deque – axe Monitor
- TPGi – ARC Domain Monitoring
- Level Access AMP (Accessibility Management Platform)

DEVIATING FROM STANDARDS

It is important to think beyond specific guidelines in website design. Ultimately what matters more than the guidelines is the user experience. For that reason, there may be occasional exceptions to the guidelines. Here are a few examples where **WCAG** Success Criteria (**SC**) can be flexible to put the user experience first:

1.3.4: Orientation

This SC states that a mobile app or web page should display either portrait or landscape orientation. An exception is if the content would not be understood as a result, for example, a cheque deposit function, a Virtual Reality (VR) goggles app, or a piano app. All of these require a landscape orientation exclusively.

1.4.3: Contrast (Minimum)

This SC states that text and icons must meet a given colour contrast requirement. Company logos, however, are exempt for this requirement.

1.4.5 Images of Text

Use text rather than images of text. Logos are exempt from this requirement.

2.4.3: Focus Order

When a user tabs through the clickable elements on a page, the focus order should go from top-to-bottom, left-to-right. In some cases, an experience can be improved by adjusting this guideline.

For example, when a primary button ("Continue") is on the right side (for **usability**), and the secondary button ("Cancel") is on the left side. In this case, the "Continue" button should be reached before the "Cancel" button.

Additional context to help screen reader users

On-screen text is the best way to label buttons; however, an aria-label is acceptable if the button is recognizable. Be sure to label the icon concisely, for example, "print" rather than "print button."

DESKTOP VS MOBILE CONTENT

Today, mobile and desktop web development are very similar. However, there are still some differences to be considered in development.

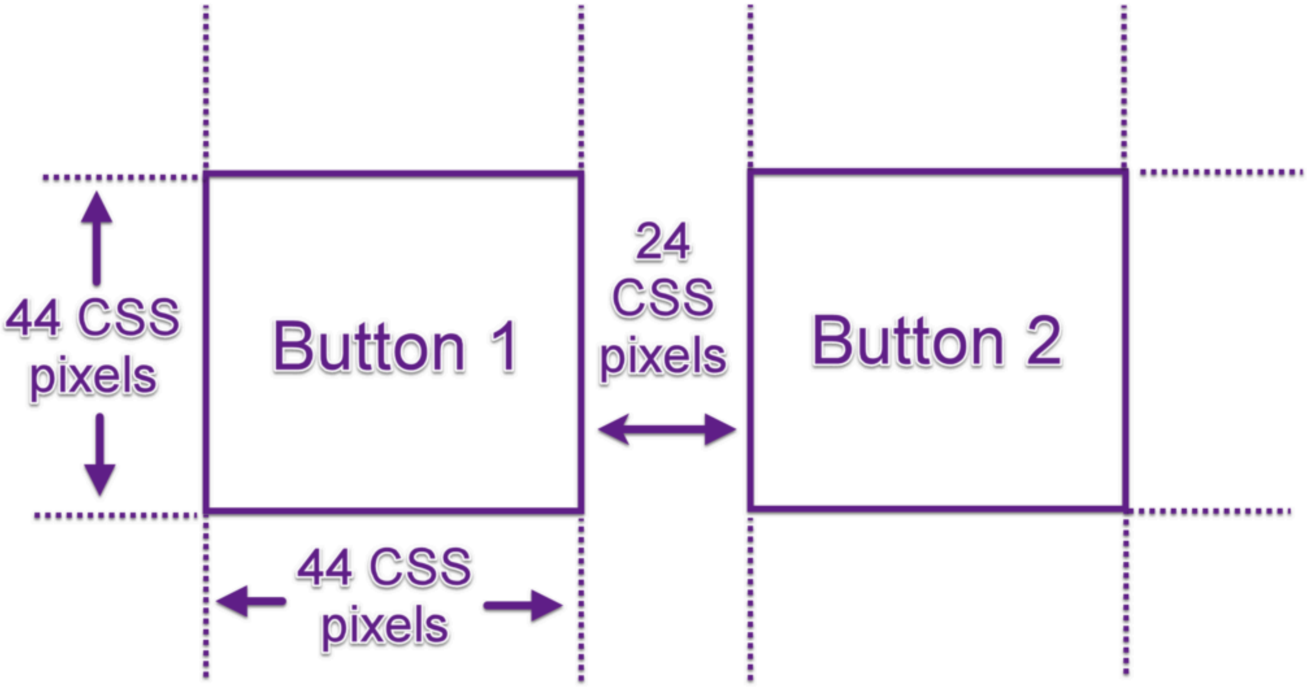
Historically, early mobile devices were often feature-poor and low-resolution. Modern mobile devices, on the other hand, like smartphones and tablets, can typically render full websites. Similarly, web content development has moved away from a distinction between desktop and mobile browsers. Responsive design chooses an optimal layout based on the user's screen dimensions. This is helpful from a developer's perspective since it allows for a single version of a website. This single version can dynamically update the layout as necessary to accommodate different screen/window sizes and resolutions.

There are still considerations specific to each modality when designing a site for multiple platforms.

Click vs Tap

A user may use a mouse to click on elements in a standard desktop browser. The mouse pointer is a precise means of control. Generally, a person who can use a mouse can click on a small target area.

WCAG success criteria specify that targets must have 24 CSS pixels of spacing between them (for level AA **conformance**), and a target size of 44×44 CSS pixels (for AAA conformance).



In addition to the accessibility considerations around small target areas, you should also consider the mobile experience. Mobile users might use their fingers to tap on icons, buttons, and other User Interface (**UI**) elements. A user's finger represents a much larger (and less focused) selection tool when tapping. In addition, the user's finger will block their view of the screen as they are tapping. Therefore, it is essential to ensure that target areas are large and, where possible, sufficiently separated to allow selections without accidentally triggering neighbouring elements.

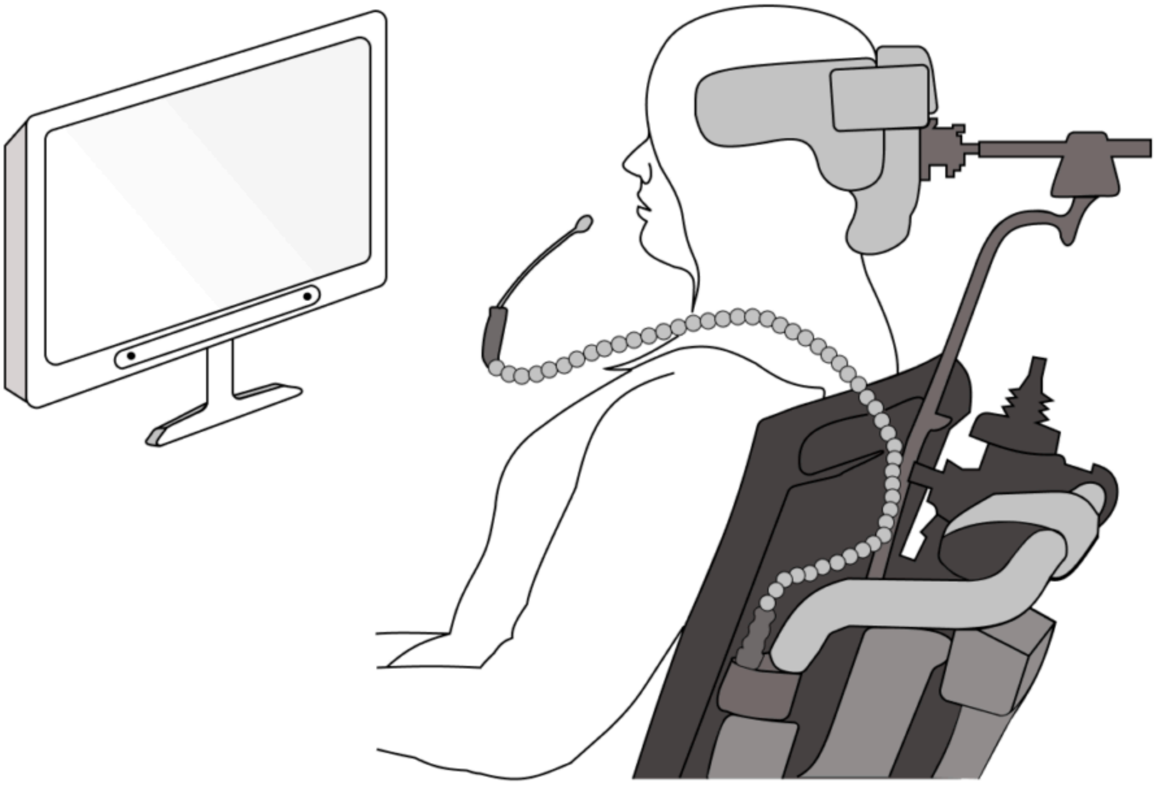
Switch vs Keyboard

On desktop systems, **assistive technologies** often take the form of virtual keyboards. They might use tab keystrokes to move between selectable user interface elements and use arrow keys (or page-up/page-down) to scroll through content.

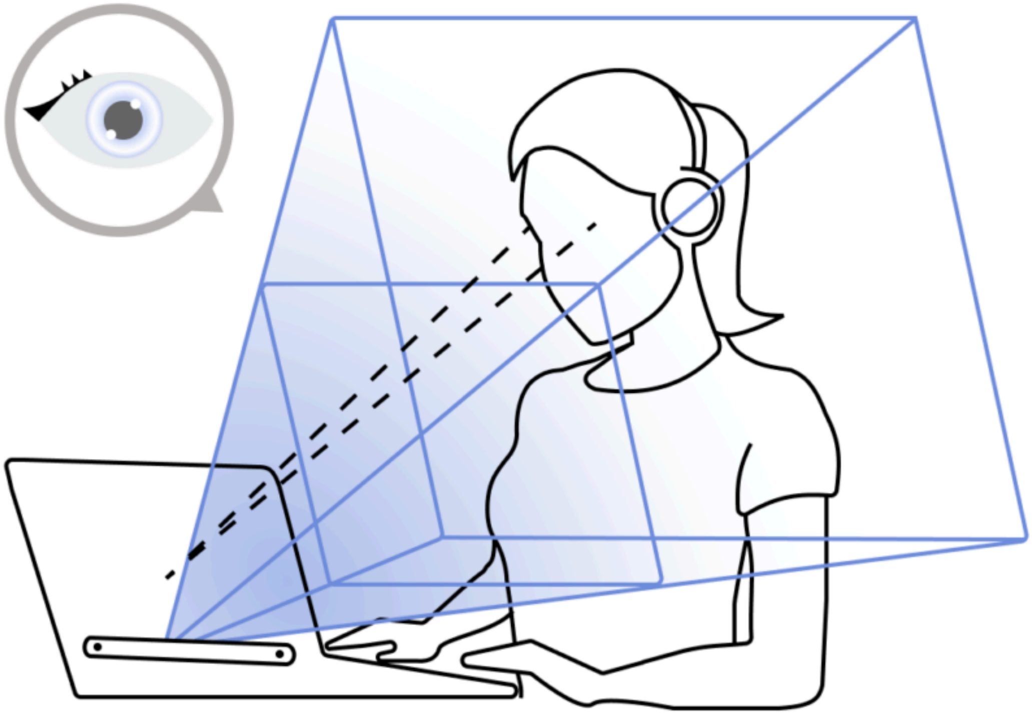
Mobile devices often use different methods. Users with dexterity **disabilities** can use a **switch control** unit consisting of two or more switches to iterate between interactive elements, or to select a "targeting" mode. Typically one switch will be used to cycle through UI elements on a page, and another switch will activate the element. Another method is to have an input switch activate a sweep across a screen, allowing a user to directly target elements on the screen. Switches can be handheld, mounted on a wheelchair, or foot-operated.

Here's a basic switch control unit with two switches – one for cycling through UI elements, and the other for activating them.

With a sip and puff switch, sipping will activate one switch, and puffing will activate the other.



An eye tracking device can be mounted along the bottom of your monitor, where it uses infrared light and a series of cameras to see where your eyes are looking. Looking at a UI element for a pre-set period of time will activate it.



Pinch to Zoom

Mobile users expect to be able to zoom in/out on content on mobile devices. Ensure that your application responds to zoom events, like pinch to zoom gestures, without breaking the layout of the content.

Device font settings

Mobile apps should respect the font system settings for each device, for example, large or bold font. Mobile websites do not need to adapt to the device settings.

BEYOND HTML: ACCESSIBLE APPLICATIONS

Although many standards, such as the Website Content Accessibility Guidelines (**WCAG**) were developed for websites and HTML, the underlying principles can be applied to other technology too. WCAG covers design principles around the presentation of content, the navigability of controls (using a mouse or keyboard, for example), and general principles of user interaction. Understanding these principles at the first stages of development will help you make appropriate design decisions throughout the project, regardless of the specific technology. The World Wide Web Consortium (**W3C**) provides guidance on applying WCAG to other technologies besides websites.

With respect to testing, there are a few standard tools for testing the accessibility of software programs through automation. Microsoft's Accessibility Insights for Windows is an example of this. Most modern operating systems have accessibility technology built into the operating system. Take the time to explore the accessibility features from your target **OS**, and use the tools to test your software throughout your development. See Testing Throughout Development for more examples and guidance.

DOCUMENT ACCESSIBILITY

Documents like PDFs, Excel Sheets, Word Documents, and Presentations create **barriers** for staff and students if not accessibly created.

Planning your content structure up front will save time in preparing any accessible document. If your goal is to create a PDF, the most direct way to make it accessible is from the source (Word, Excel, PowerPoint, etc.). You will also need Adobe Acrobat Pro or PDF Accessibility Checker (PAC) 2 (Windows only) to make your PDFs accessible.

You can follow these guidelines when aiming to create an accessible document from any application:

- Use plain language
- Add descriptive contextual links
- Add alternative text
- Use document headings
- Consider **accessibility** for time-based media (audio / video)
- Use built-in accessibility checkers

Use plain language

Plain language (also called simple language) ensures that your text is concise and easy to understand. Using simple language benefits people with learning or other cognitive disabilities and language barriers. Determine the needed level of simplicity based on the target audience. In a post-secondary context, simplicity of language is further complicated as advanced level reading or technical knowledge may be a requirement within certain curricula or programs. However, for information that is meant to be accessed by the public, use plain language as much as possible.

To write using plain language, try the following tips:

- Cut extra words
- Use short, everyday words, for example:
  - Ask, instead of inquire
  - Next, instead of subsequently
  - Help, instead of render assistance
- Use the active voice (i.e., the doer of the action is at the beginning of sentence)
  - “Customers can now board the train on platform 12” (active voice)

    - "The train on platform 12 can now be boarded" (passive voice)
- Write in short sentences
- Use action verbs
- Avoid difficult words and explain difficult words if you must use them
- Use contractions (i.e., don't instead of do not)
- Use neutral pronouns (they/them)
- Use bulleted lists

For more information and resources on plain language, checkout these links:

- Translation Bureau, Government of Canada, Plain Language
- Online readability checker
- Readability Calculator
- Datayze Readability Analyzer

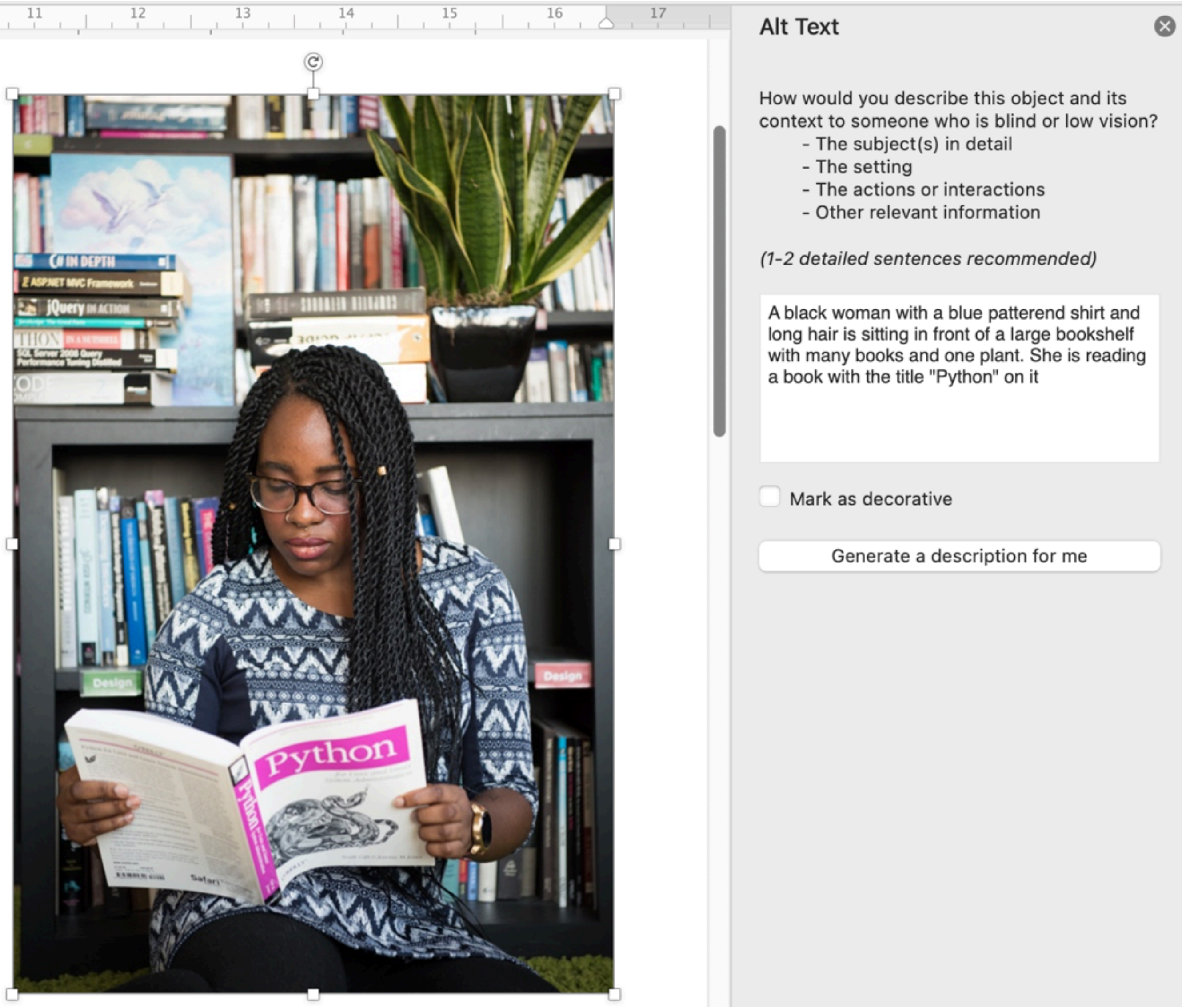
Add descriptive contextual links

On websites and in apps, it is common to have a link that takes you to a different page. Sometimes the text that describes the link is not very clear. For example, if a user who relies on a **screen reader** hears "Click here" without any additional information, they won't know what that links to. Link text should make it easy to understand where the link is going. Avoid "click here" or "read more," because these are not descriptive and do not tell the user what the purpose of those links are or where the links will take them.

Add alternative text

Alternative text are descriptive words applied to visual information, such as images, charts, and diagrams. Alternative text is embedded and usually hidden text. When a screen reader encounters an image, it reads out whatever alternative text is attached to it.

When using MS Word, you can add alternative text, or alt-text, to images by right-clicking on the image and scrolling to "edit alt-text." In MS PowerPoint, select your image, then navigate to the "picture tools" format on the main ribbon. "Alt-text" is found under the accessibility options.



Alt Text

How would you describe this object and its context to someone who is blind or low vision?

- The subject(s) in detail
- The setting
- The actions or interactions
- Other relevant information

(1-2 detailed sentences recommended)

A black woman with a blue patterend shirt and long hair is sitting in front of a large bookshelf with many books and one plant. She is reading a book with the title "Python" on it

☐ Mark as decorative

Most other tools, including email systems and PowerPoint also have built in tools to add descriptive text. Many applications will generate automatic descriptions of an image. Do not rely on these descriptions as they are often poor quality.

Use document headings

Headings can help structure a document. They also increase the document's readability. Headings visually break up text and help the reader navigate and locate the information they are looking for.

It is important that headings are properly coded for persons using screen readers. Use the headings formatting found in the "styles" panel of MS Word. Use level 1 headings for main titles. Use level 2 headings or higher for sub-headings within a specific section. This helps screen reader technology to work effectively and allows a person to navigate the content.

Consider accessibility for time-based media (audio / video)

Be sure to provide a text equivalent for all time-based media formats, including audio and video. Text alternatives include captions and transcripts. Do not rely on audio alone to communicate meaning. Similarly, include descriptive audio in videos and don't rely on video alone to communicate meaning.

Avoid repetitive animation and flashing effects as this may trigger people with photosensitive seizure disorders. You can also consider programming reduced motions settings, meaning that a user can reduce or remove animations if they wish.

Application-specific guidelines

Adobe PDF (Portable Document Format)

PDF documents can be made accessible by including tags as recommended in the latest PDF/UA Universal Accessibility Guidelines and the WCAG 2.0 PDF Guidelines. Including good accessible formatting in your source document can help with PDF tagging, but additional work will still be needed.

Microsoft products

Microsoft includes accessibility checkers in their applications. To run the checker, click on Tools and then "Check Accessibility". The program will identify some (not all) potential barriers for people with disabilities and provide suggestion on how to remove those barriers and improve the accessibility of your documents.

Microsoft Excel

Consider the following tips to make your Excel spreadsheet more accessible:

- Add a description of the document layout. For example, a note indicating the direction of the text flow, whether it is top to bottom or left to right.
- Use consistent conventions throughout the document.
- Apply the proper formats for your data cells. Format text data as text, money as currency, and dates as dates.
- Identify row and column headers in data tables.
- Separate headings so that it is understandable.
- Label each worksheet with a meaningful name as opposed to Sheet1, Sheet2, etc.

Microsoft PowerPoint

Consider the following tips to make your PowerPoint presentation more accessible:

- Keep presentations short and clear. This helps people with learning or other cognitive disabilities stay focused.
- Keep your message as concise as possible; do not crowd text on your slides. This also considers differing amounts of time someone might need to read and understand what is on the slide. Considered a maximum of 4-5 bullets per slide.
- Keep in mind that the layer order defines the reading order for **assistive technologies** from PowerPoint.
- Remember that because presentations are viewed from a distance, font size and colour contrast are highly important. Use a minimum of 24 pt font with high-contrast colours
- Minimize or remove slide animations or transitions because these can cause unnecessary visual distractions.

If your presentation is going to be shared in a digital format, e.g., on your website or in an email, consider persons with vision loss and those using screen-readers.

- Provide alt-text for all images
- Use the built-in slide design templates since they provide navigational structure for screen readers

For more information, please refer to Microsoft's Accessibility Guide for PowerPoint.

ACCESSIBILITY STATEMENTS

Consider developing an accessibility statement to post on your site. An accessibility statement demonstrates your commitment to **accessibility** and specifies your accessibility standards. Documenting your site accessibility features and known limitations invokes trust and invites discussion. You can also invite visitors to the site to provide feedback and respond to that feedback to let them know you're committed to doing your best to incorporate their suggestions. The **AODA** requires organizations to establish a "process for receiving and responding to feedback about the manner in which it provides goods, services or facilities to persons with disabilities".

The World Wide Web Consortium (**W3C**) has an excellent resource on Developing an Accessibility Statement. As an example, the Government of Canada accessibility statement may give you some ideas for creating yours.

10.

ACCESSIBLE PROCUREMENT

Policies or other documentation outlining processes and commitments to accessibility can help identify when and how accessibility criteria will be incorporated into procurement as required by the **AODA**. For some procurements, such as small purchases by faculty, it may not be practical to incorporate accessibility criteria. However, consider incorporating accessibility criteria for other large purchases or critical digital tools. This will ensure equal access for all learners.

Many colleges and universities have policies outlining accessible procurement practices. However, not many Ontario post-secondary institutions have written policies for accessible digital procurement. For example, many American universities have policies on procurement for digital accessibility because of Section 508 legal requirements. While details of these policies may not be directly applicable to Ontario post-secondary institutions, they do provide a general framework for incorporating digital accessibility considerations into procurement processes.

For example, see:

- Accessible Procurement Guide, Council of Ontario Universities
- 9 Steps to Procuring Accessible ICTs for Inclusive Education, Global Initiative for Inclusive ICTs (G3ICT)
- Electronic and Information Technology Accessibility Policy and Procedures, University of Montana
- Guideline on Making Information Technology Usable by All, Government of Canada
- Procedures and Procurement for Digital Accessibility, New York University
- Procure Accessible Technology, University of Wisconsin – Madison Information Technology

Each organization will approach accessible digital procurement slightly differently. The policies above and other leading practices generally point to steps outlined on these pages.

ORGANIZING THE PROCUREMENT

Gathering the Right People

When beginning procurement of major educational technology, it is helpful to gather the correct stakeholders. Ideally, this group would include stakeholders for procurement, information technology, **disability** and **accessibility**, students, teachers/professors, and administrators.

Involve this working group at all stages of procurement, as appropriate and feasible. Stakeholders can help develop the specifications, draft the solicitation documents, evaluate proposals, and meet with vendors.

Gathering Information

Before beginning the procurement process, gather information about what is already known about the digital accessibility **barriers** in your organization. Accessibility Services Offices and adaptive technologists are a great source of information about what barriers students with disabilities are already encountering. Keeping these barriers in mind will help keep accessibility a priority throughout the procurement process.

DETERMINING SPECIFICATIONS

The first step is to determine what, if any, specific requirements apply to your organization. In addition to the general obligation to "incorporate accessibility design, criteria and features when procuring," the current **AODA** requirements relevant to colleges and universities are outlined on the Legal Requirements page. Keep in mind that there may be changes to these legal requirements in the future.

In addition to these legal requirements, post-secondary institutions may want to incorporate accessibility specifications into their procurements. Establishing a policy that defines the level of accessibility will make it easier to consistently provide accessible content. See the Procurement page for guidance on policies.

Defining Specifications

Some organizations include a general statement in solicitations vaguely requiring **compliance** with standards or legislative guidelines. A general reference makes it nearly impossible to determine whether a product meets accessibility requirements. For example, a requirement to "comply with AODA standards" will be meaningless for most procurements because there are no specific AODA standards on procurement. Instead, specifically reference the accessibility requirements that the product is expected to meet.

Exact specifications depend on what is being procured (e.g., hardware, software, tablet, phone) and its intended functionality/ purpose (e.g., voice communication; video transmission). To ensure full accessibility, some specifications that may seem irrelevant will need to be included. For example, if product documentation includes an embedded video, specifications for video and audio may become relevant.

There are two main tools to help you define specifications in procurement documents: one from the Canadian government and one from the US government. They are based on very similar, but slightly different, standards (ETSI and Section 508). The Canadian tool is more applicable to Ontario institutions as it does not refer to specific American legislation.

INCORPORATING SPECIFICATIONS INTO SOLICITATION

Market Research

Because of federal accessibility requirements in the United States and Europe, there are many more accessible products globally available than ever before. (See International Standards page for more information). Market research can inform a procurement strategy by defining which accessible products are available. Many vendors also provide or publish Accessibility Conformance Reports (ACRs) or Voluntary Product Accessibility Templates (VPAT) which outlines their current state of accessibility.

In addition, consider meeting with vendors to discuss the importance of accessibility in this procurement. The Global Initiative for Inclusive **ICTs** (G3ICT) has prepared a Discussion Guide for conversations with vendors, that includes key questions and how leaders in accessible technology may respond. G3ICT recommends:

- Organizing a series of meetings with vendors that have been selected as meeting accessibility criteria in a Call for Tender/RFP
- Clarifying how vendors' products meet accessibility specifications
- Asking questions about the vendor's experience with accessibility and how their business processes ensure the accessibility of their products
- Requesting demonstrations of a vendor's product within the planned setting with the focus on its accessibility and application for different use cases, including individual use and collaboration

Incorporate specifications into solicitation

In addition to specifications, solicitations may need to:

1. Specifically reference the organization's accessibility commitment and standards and their Accessibility for Ontarians with Disabilities Act (**AODA**) obligations.
2. Include specific specifications in the solicitation (see Determining Specifications page for resources). G3ICT recommends including "precise details of the product or service required, including accessibility requirements, and terms of the procurement exercise".
3. Specify that the tender must include accessibility in the design and development process if the tender is

for a new product.

4. Instruct vendors on how they are expected to demonstrate the accessibility **conformance** of their products (e.g., VPAT/ACR). This could include documentation expectations (see evaluation section below), testing, typical user scenarios, and plans to address known gaps. See Section508.gov for a template of such instructions to vendors. Requesting VPATs also has the indirect benefit of communicating the importance of accessibility to vendors.
5. Request references from vendors specifically about the accessibility of the product.
6. Clarify when these specifications are minimum acceptance criteria and how they will be evaluated alongside other factors.

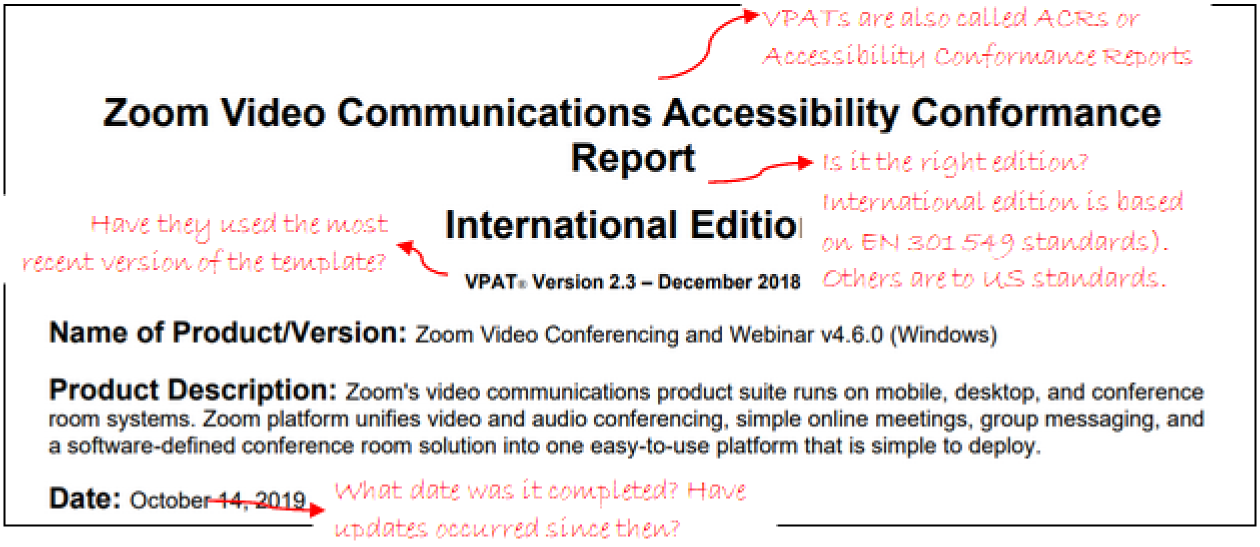
INTERPRETING VENDOR CLAIMS

VPATs and ACRs

Vendor claims of accessibility are not reliable without being confirmed through written documentation and validation. The commonly accepted method of reporting accessibility is through Accessibility Compliance Reports (ACRs). There are also called Voluntary Product Accessibility Templates (VPATs). A VPAT is a generic template that companies use to draft reports on their product's accessibility, based on either Section 508 (American) or EN 201 549 (European) standards.

For most commercial off-the-shelf products, VPATs are available online. Most major vendors whose customers include post-secondary institutions in the United States or Europe will have previously completed VPATs and will be able to share them.

It is important to note that a VPAT is not an audit report, though the vendor must have performed an audit to complete the VPAT. While useful, VPATs cannot be assumed to be accurate. Since they are written by the vendor, VPATs may exaggerate the accessibility of any given product.



VPATs are also called ACRs or Accessibility Conformance Reports

Zoom Video Communications Accessibility Conformance Report

Is it the right edition? International edition is based on EN 301 549 standards). Others are to US standards.

International Editio

Have they used the most recent version of the template?

VPAT® Version 2.3 – December 2018

Name of Product/Version: Zoom Video Conferencing and Webinar v4.6.0 (Windows)

Product Description: Zoom's video communications product suite runs on mobile, desktop, and conference room systems. Zoom platform unifies video and audio conferencing, simple online meetings, group messaging, and a software-defined conference room solution into one easy-to-use platform that is simple to deploy.

Date: October 14, 2019

What date was it completed? Have updates occurred since then?

Red Flags

The following are some items for concern or indications that a VPAT may not be accurate:

- An old version of a VPAT (VPAT 1.X) indicates that the vendor has not updated its documents.

Similarly, a VPAT dated more than 12 months ago would suggest that the product has not been updated.

- The VPAT is not completed or is completed with non-standard language (e.g., it uses “Passes” and “Fails” rather than “Supports”, “Partially Supports”, or “Does Not Support”). This indicates a lack of expertise in accessibility by the author.
- One VPAT for multiple products or inaccurate/ unclear descriptions of the product. A VPAT needs to be product-specific to be accurate.
- Evaluation methods are unstated or indicate that only automated testing was completed. Automated testing can only test for approximately 30% of **conformance**, but it cannot be relied upon alone.
- The VPAT states that it “Supports” all accessibility specifications. While this is possible, it is incredibly rare for a product to support every criterion. It may indicate that the author did not understand or confirm **compliance**. Note that a VPAT may say “Supports” instead of “Not Applicable” where a criterion is not relevant.
- VPATs that consistently state “Supports” and often includes minimal content in the “Remarks” column. The Remarks column is not intended to be blank. It should provide information on how the authors arrived at this conclusion. If most responses in the VPAT paraphrase the criterion, question the authors on how they reached those conclusions.
- Authors may also put “Not Applicable” for many criteria. While this may be an appropriate response, it should be accompanied by a brief explanation. “Not Applicable” responses without explanation are suspect as they make it challenging to determine why the criterion does not apply.

If you encounter red flags

If any of these red flags are raised when reviewing a VPAT, you can do some quick tests to confirm the accuracy of the claims in the VPAT. This is not a substitute for a full accessibility audit but it can validate some claims in a VPAT. For example, you could do the following tests and check that the results are accurately described in the VPAT:

- Test the product with the keyboard (see 4.5.2 if you’re not sure what this entails).
  - Are all functional items available by keyboard?
  - Do all active items have a highly visible keyboard focus indicator?
  - Can you activate hidden content like menus?
  - Can you complete and submit forms?
- Is there movement on the site? Does it have an accessible (mouse, keyboard, touch) pause feature, or does it stop within 5 seconds?
- If you submit an empty or incorrect form (e.g., a log-in), are the errors described appropriately?

If the results are not consistent with the VPAT, the VPAT is not reliable. In that case, you cannot know if the product is accessible without testing it yourself.

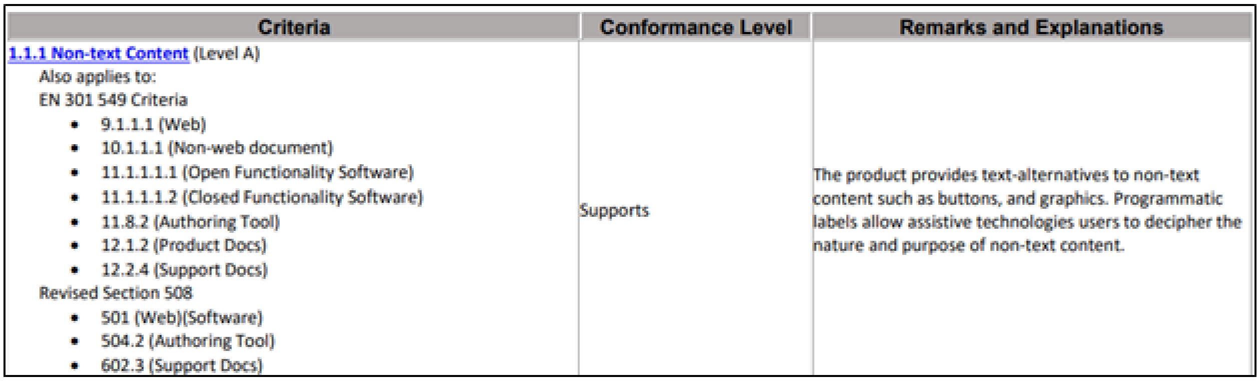
If questions arise from the VPAT, contact the vendor. You may want to ask questions like:

- Why is this VPAT dated 18 months ago, has the product not been updated?
- Why did this criterion say Not Applicable when the product has that feature?
- What testing was done to reach these results?

Signs of a Reliable VPAT

In contrast, the following are positive indications of the reliability of a VPAT:

- It was authored by a third party, with greater expertise in the accessibility criteria and are more likely to have an objective view.
- There is substantial content on the VPAT's cover page, including detailed information about the exact type of testing employed. For example, listing multiple **assistive technologies** and types confirms the rigor of the testing. Similarly, a note about the scope of testing is reassuring.
- Examples of how the product meets specific success criterion demonstrates an understanding of the success criteria.
- The VPAT is publicly available, which provides an indication of the vendor's confidence in the report.



Criteria	Conformance Level	Remarks and Explanations
1.1.1 Non-text Content (Level A) Also applies to: EN 301 549 Criteria • 9.1.1.1 (Web) • 10.1.1.1 (Non-web document) • 11.1.1.1.1 (Open Functionality Software) • 11.1.1.1.2 (Closed Functionality Software) • 11.8.2 (Authoring Tool) • 12.1.2 (Product Docs) • 12.2.4 (Support Docs) Revised Section 508 • 501 (Web)(Software) • 504.2 (Authoring Tool) • 602.3 (Support Docs)	Supports	The product provides text-alternatives to non-text content such as buttons, and graphics. Programmatic labels allow assistive technologies users to decipher the nature and purpose of non-text content.

EVALUATION AND NON-CONFORMANCE

The Government of Canada's ICT Accessibility Requirements Wizard produces evaluation templates to match the specifications that it creates. For each procurement, you will need to determine which accessibility specifications are minimum requirements and which criteria will be evaluated and scored alongside other requirements. One factor in this determination will be applicable legal requirements. Other factors will include the quality of alternative products. Where there are both accessible and inaccessible products that meet a post-secondary institution's needs, the accessible product can be prioritized. For products that impact many people or are more critical, you may consider higher accessibility expectations.

At times, you may conclude that the best course of action is to procure a product that does not fully meet the accessibility specifications. The **AODA** states that if "a designated public sector organization determines that it is not practicable to incorporate accessibility design, criteria and features when procuring or acquiring goods, services or facilities, it shall provide, upon request, an explanation." In such situations, it would be best practice to document the decision, why it was not practical to procure an accessible alternative and develop a corrective action plan if possible.

Another best practice to consider when a product will not be fully accessible is to develop a plan for how to work around the accessibility issues. This workaround is sometimes called an **accommodation** or an "equally effective alternative action plan." These are critical to ensure that people with **disabilities** still have access to the needed information or resources.

An organization can be proactive in creating an equally effect alternative action plan for specific known issues or reactive by responding to individual requests. Below is an extract of an equally effective alternative access plan for known challenges with Zoom.

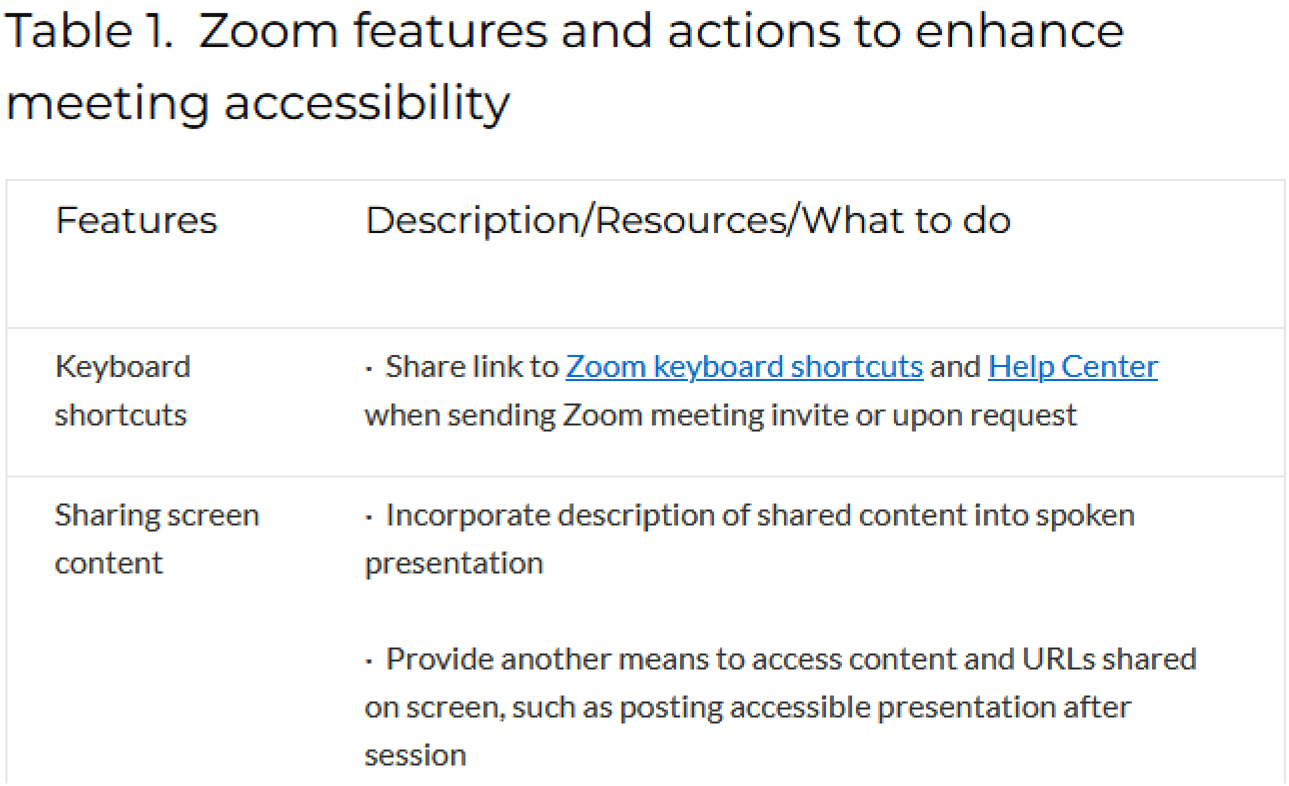


Table 1. Zoom features and actions to enhance meeting accessibility

Features	Description/Resources/What to do
Keyboard shortcuts	· Share link to Zoom keyboard shortcuts and Help Center when sending Zoom meeting invite or upon request
Sharing screen content	· Incorporate description of shared content into spoken presentation · Provide another means to access content and URLs shared on screen, such as posting accessible presentation after session

Screenshot captured from https://dssbackyardfence.wordpress.com/

UDL (Universal Design for Learning in Higher Education) on Campus has resources on developing an equally effective alternative action plan. Note that as an American resource, it will include information relevant to those standards rather than Canadian or European standards.

CONTRACT LANGUAGE

Once a product and vendor have been selected, the contract will incorporate the various accessibility specifications, including how delivery will be tested and confirmed, and how non-**conformance** will be addressed. The following is a checklist of things to consider when finalizing contract details:

- State the applicable accessibility standard (e.g., **WCAG** 2.1 Level AA).
- Define the process for handling exceptions to standards and specifications.
- Agree on deployment and support procedures, making sure that the vendor's dedicated support team covers accessibility.
- Include all relevant information on testing and delivery.
- Include that the vendor is responsible if the product turns out not to be accessible by requiring the vendor to remediate **barriers**, deliver a new product, return money paid, or pay damages.
- Include provision for soliciting feedback from both vendors and users.
- As accessibility challenges may not be readily apparent, specify the vendor's responsibility to remediate a barrier whenever it is identified.

For additional information, you can visit the Global Initiative for Inclusive **ICTs** (G3ICT) resource 9 Steps to Procuring Accessible ICTs for Inclusive Education

Section508.gov (the US accessibility legislation resource) has developed sample contract language for various situations:

- Custom ICT development services
- Installation, configuration and integration services
- Maintenance, upgrades and replacements
- Service personnel
- Hosting services
- Validation for ICT items
- Documentation
- Conformance reporting
- Non-**compliance**

This language will need to be adapted to refer to the specific accessibility standards of your college or university, instead of the Section 508 requirements.

PART IV

TESTING

Whether purchased or developed, the only way of knowing that technology is accessible is to test it. This section includes guidance on when to test, how to test, the testing process, and some common tools to include in testing.

Covered in this Section:

- Establishing success criteria for websites
- When to test
- Ways of Testing
- Overview of the Testing Process
    - Writing Up Test Results
    - Priority and Severity
- Testing Tools
    - Keyboard Testing
    - Chrome DevTools Testing
    - Colour Contrast Testing
    - Deque axe Browser Extension Testing
    - Screen Reader Testing
    - Accessibility Insights for Windows
    - Installation-Free Testing
    - Document Accessibility Checkers

11.

ESTABLISHING SUCCESS CRITERIA FOR WEBSITES

To improve the accessibility of your web content, you can aim to conform to Web Content Accessibility Guidelines (WCAG). See International Standards page for more information.

WCAG 2.0 vs 2.1

The World Wide Web Consortium (W3C) and Web Accessibility Initiative (WAI) are responsible for the creation, maintenance, and updates for WCAG. WCAG is regularly reviewed and updated to reflect a continuous process of improving accessibility and in reaction to advancements in technology and web content development. Each update is numbered with two digits, separated by a period. The first digit represents the major version number, and the second digit represents the minor version number.

The Accessibility for Ontarians with Disabilities Act (**AODA**) requires only WCAG 2.0 **conformance** at present. Given the versions of WCAG that are currently published and the broad use of touch screen technology, it is a good idea to start aiming for WCAG 2.1 now. This will prepare your website for the near future and WCAG 2.1 adds success criteria related to touch screens. Keep in mind that WCAG 2.2 and WCAG 3.0 are also being drafted currently and are coming soon.

Levels A, AA, and AAA

There are a number of success criteria within WCAG. Each success criterion corresponds to a specific aspect of digital accessibility. WCAG denotes three levels of accessibility depending on the number of success criteria that a website satisfies. When a web page satisfies success criteria, we say it is **conformant** with the criteria (rather than compliant).

The three levels of accessibility, from lowest to highest, are Level A, Level AA, and Level AAA. Level A is basic accessibility, AA is considered very good, and AAA is excellent accessibility. Most organizations aim for Level AA, because sometimes it is not possible to satisfy every Level AAA success criterion. Even if your content is only required to be Level AA conformant, it is good practice to document any Level AAA features.

Table 1 below shows the number of success criteria that must be satisfied to achieve the different levels of accessibility associated with WCAG 2.0 and 2.1.

	Level A	Level AA	Level AAA
WCAG 2.0	25 Success Criteria	13 Success Criteria	23 Success Criteria
WCAG 2.1	30 Success Criteria	20 Success Criteria	28 Success Criteria

Table 1: WCAG 2.0 and 2.1 Requirements

- For a web page to meet Level A – it must satisfy all of the Level A Success Criteria.
- For a web page to meet Level AA – it must satisfy all of the Level A Success Criteria and all of the Level AA Success Criteria.
- For a web page to meet Level AAA – it must satisfy all of the Level A Success Criteria, all of the Level AA Success Criteria, and all of the Level AAA Success Criteria.

Success Criteria

Success Criteria are written in clear language and are designed to be testable. This means that it is clearly identifiable if web content conforms to or fails a specific criterion. For example, Success criterion 1.1.1 states that "All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for [certain situations]." Based on this, someone can examine all non-text content and confirm that this criterion is met, or not

Typically, testing is done through a combination of automated testing and human evaluation (more on testing on page). Formal testing can also be complemented by **usability** testing, where users with disabilities interact with the web content and can give direct feedback on what does and does not work for them. Typically, testing is done through a combination of automated testing and human evaluation. See the Ways of Testing page for more information. Formal testing can also be complemented by usability testing, where users with disabilities interact with web content and can give direct feedback on what does and does not work for them.

All Success Criteria can be found on the WCAG Quick Reference.

Techniques and Failures

For each of the Success Criteria, WCAG provides additional information, including sufficient and advisory techniques, and example failures. This additional information is to support you through the process of meeting the Success Criteria:

- **Sufficient Techniques** are reliable ways to meet the success criteria. You can use other techniques not listed, if they work. See: Sufficient Techniques.

- **Advisory Techniques** are suggested ways to improve accessibility, but they are not required. See: Advisory Techniques.
- **Failures** are test results which show success criteria were not met. See: Failures.

More detail is available on the "Understanding Techniques for WCAG Success Criteria" page.

Normative vs. Informative

It is important to keep in mind that the sufficient and advisory techniques are provided as additional information to support you in meeting the criteria. As described in WCAG, techniques are **informative** which means that you do not need to use any specific technique, as long as you use one that works.

On the other hand, all Success Criteria are **normative** – this means that your web pages **must** meet them in order to be conformant with WCAG.

Resources: How to Meet WCAG (Quick Reference)

12.

WHEN TO TEST

Good accessibility does not happen by accident. It is crucial to consider accessibility in the early stages of developing or redesigning a website or web application and incorporate accessibility evaluations early and throughout the process. The earlier you can identify accessibility problems, the easier and less costly it is to address them.

To borrow a popular expression in modern software development: "shift left." It means that its best to test early to fix issues with minimal expense. It will also save time later in the process when you won't have to re-test similar features.

Content Development

When new content or a new webpage is being created, test for accessibility as soon as possible by inspecting page elements and viewing the code in their browser's **DevTools** (developer tools). Developers can also use automated testing tools like Deque's axe to increase the frequency of testing. Testing with specific known technologies, like screen readers (**NVDA** on Windows or VoiceOver on Apple products) is also recommended.

Content Updates

As is the case with new content, it is important to review your content updates for accessibility before they are published. One way to do this is to leverage the preview feature that is available from most Content Management Systems. In some cases, you can work with your developer to test your updates early. It is always better to review content and ensure its accessibility before it gets published to your production site.

Ongoing Site Scan

For websites already in production, there are several accessibility dashboard tools that continually monitor the site looking for accessibility issues. These may be particularly appropriate for large organizations with large sites. These are typically paid tools. Some prevalent in the industry include:

- Deque – axe Monitor (which is different than Deque axe's free testing browser extension)
- TPGi – ARC Domain Monitoring
- Level Access – AMP (Accessibility Management Platform)

Resources : Evaluating Web Accessibility Overview

13.

WAYS OF TESTING

The most thorough and complete way to test a website for its accessibility is to complete a full audit, meaning going through the entire site to identify and address accessibility issues. However, it is not always practical to audit an entire site all at once. When a full audit is not practical, prioritized quick tests and early bug fixes are a helpful way to reduce errors that may be found in later audits.

Quick Checks

One way to prioritize your testing is to focus on features or pages that users have provided feedback on. With a limited amount of time, you can test a particular page or a particular workflow that is related to those features highlighted by user feedback.

If you find issues on one page or feature, try to determine if similar issues appear in related pages of your site. Once you identify and address one issue, you may be able to save time when solving other, related issues.

In-House vs. Third-Party Audits

When a full audit can be completed, it can either be performed by internal staff or by a third party.

In-House Audits

Auditing your own site (or helping internal IT staff do an audit) requires a certain level of familiarity with digital accessibility and accessibility testing, and enough human resources to carry out the audit. Completing the audit in-house could have additional benefits, such as increased staff familiarity with the site. Like quick checks, once you find one issue, you can look at related content or similarly structured pages that may have the same issue. Looking for these common patterns or issues and fixing them can lead to common solutions.

Third-Party Audits

Another option is to hire an accessibility consulting company to complete an audit, which comes with its own benefits. The auditors should have very up-to-date knowledge of web trends, testing techniques, and coding tips for your team.

Especially, when conducting external audits, it is important to approach the work strategically and define a clear scope of work. One way to prioritize the work is based on traffic, meaning the high-traffic or busiest pages will be tested first. A way to organize the work is through defining a series of sprints: short timeframes (typically a couple of weeks) during which a set number of pages will be tested and the findings will be reported on. Testing your site in sections could give you time to fix common issues between sprints. For example, you could define four batches of pages to test, or four sprints. After the first batch is tested, an audit report would be delivered, and your IT staff would have some time to address the issues. Then when testing starts for the second sprint, there should be fewer issues, since many would have already been addressed.

Usability Testing

Another option is to recruit (and preferably, pay) people with disabilities to test your site. People with disabilities will give you direct feedback based on their lived experiences, which will differ from person to person. Consider a broad range of people with different disabilities, for example people who:

- Are blind
- Have low-vision or colour blindness
- Have dexterity issues
- Have cognitive issues, such as **ADHD** (attention deficit hyperactivity disorder) or **ASD** (autism spectrum disorder)

When taking this approach, it is important to remember that every person with disabilities has a unique experience, and so you should not rely on users alone to determine whether your site is accessible or not. **Usability** testing complements, and does not replace, accessibility guideline **conformance**.

14.

OVERVIEW OF THE TESTING PROCESS

Once you have decided to conduct accessibility testing on your website, there are four main steps:

1. Planning the test
2. Carrying out the testing
3. Compiling the results of the testing
4. Fixing all the identified defects or issues.

The rest of this page goes over some tips on planning your testing process and conducting the testing itself.

Planning Tips

Planning your testing process ahead of time can make the testing process more efficient and smooth. When you plan your testing process, consider the following:

- Know your ultimate objectives and keep them in mind. For example, you may want to consider WCAG Success Criteria before testing and the WAI-ARIA Authoring Practices site for **components** like accordions or menus. A checklist or spreadsheet may also be helpful. Click here to download a sample WCAG 2.1 audit template (in Excel format).
- Understand your end users' priorities. There may be specific pages and workflows (such as enrolling in a course or checking a grade) that are more frequently used or should otherwise be prioritized. You can determine these priorities by discussing the site with stakeholders or users or you can reference site data and analytics.
- Define modules for testing, where each module requires roughly the same level of effort to test. Usually, one page equals one module. For complex components or **widgets** on a page, you will want to treat them as separate modules. For example, if your page has some static text, a search bar, and a calendar, you may want to define these as three separate modules.
- Organize testing into sprints, where each sprint will cover a specified set of modules.
- Schedule time to fix issues in between each sprint.
- Determine if you need access to test accounts or unique data to test specific features. For example, you

may need a hypothetical student's account and hypothetical courses and grades to test certain functionality.
- Find people to test. Make sure they have the expertise and the time to do the testing.
- Establish clear communications with IT staff and staff responsible for the content and develop workflows related to fixing issues once they're identified.

Testing Tips

When testing for accessibility, it is good practice to:

- Use appropriate testing tools (like those discussed the Testing Tools section). Make sure you're able to install them and understand how to use them. These tools can be used in any order. Recommended tools include:
  - Keyboard
  - Chrome DevTools
  - TPGi Colour Contrast Analyser
  - Deque axe browser extension
  - NVDA screen reader
  - To make sure you've covered all the WCAG success criteria, it's a good practice to log them into a spreadsheet. (Click here to download a sample WCAG 2.1 audit logging spreadsheet.)
  - To test a Windows application, follow Accessibility Insights
- Log issues into a bug tracking system like Jira or a spreadsheet. Describe the unwanted behaviour, and recommend a specific fix if possible. Screenshots of the error are helpful.
- Point to a specific WCAG success criterion for all identified issues, where possible.
- Communicate bugs to third-party vendors if you do not have access to fix the error.
- Whenever possible, work with IT staff and staff responsible for the content to address any issues promptly. Document issues that cannot be prioritized in a backlog.
- Test your fixes and conduct regression testing to ensure the site and its features are still working as expected.

Resources:

- Evaluating Web Accessibility Overview

- WAI-ARIA Authoring Practices site

WRITING UP TEST RESULTS

It is good practice to record the results of any accessibility testing you conduct. This will make it easier to communicate your results to any stakeholders or organization leadership and will also help you prioritize and action any needed fixes.

Important Information

Include as much of the following information as possible for each issue:

- A short description of the issue to make tracking the issue easy.
- A longer description of the issue that explains why the issue does not conform to a particular **WCAG** success criterion or how it will impact users with specific **disabilities** or break some organizational standard.
- A severity for the issue to describe the impact on the user.
- Steps to find, recreate, and test the issue (e.g. a URL or a login ID).
- Current HTML code (which is causing the issue).
- Recommended HTML code (to fix the issue). In cases where the issue is linked to a complex **component** (like a set of tabs, a mega menu, or a carousel), it might be best to point to an existing accessible pattern, like the ones on the ARIA Authoring Practices site, instead of developing recommended HTML code
- A screenshot, if possible, preferably with the issue highlighted with a box or an arrow.

Writing a testing report

The format of a testing report can vary depending on the audience, for example:

- **Executives** may appreciate a high-level executive summary, which provides some insight into an overview of accessibility challenges for users and legal liabilities.
- **Managers** may want an overview of the results, like a dashboard that showcases the top five issues or issues grouped by category.
- **Developers** may appreciate recommended code fixes or links to accessible patterns to solve issues. Developers may want a spreadsheet (like this one) or other format that can easily be imported to their

bug tracking software.

Templates

There are various formats you can use to create a testing report.

- If you use some form of enterprise software that scans your site, it may have a feature to export a report for you.
- You may want to create a report template, based on the above tips or a previous report.
- The **W3C** (World Wide Web Consortium) site offers a Template for Accessibility Evaluation Reports, that includes key information like the scope of review, the reviewer(s), the review process, results, and recommended actions.

Ultimately, the report should meet your needs and the needs of your audience. Consider who will be using and reviewing this information and consider if you need to develop multiple types of templates for different audiences.

PRIORITY AND SEVERITY

It may not be practical to fix all issues or defects right away, test the fixes, and implement them. For this reason, assign a Severity and Priority to each issue to determine the order issues will be fixed in.

Severity

The accessibility consultant, or the person doing the testing, should assign severity to each issue. The severity indicates how the issue could impact users.

Here's an example of a severity scale:

- **S-1 Critical** – Users with some disabilities cannot use any part of a system or website. An immediate fix is needed.
- **S-2 High** – Users with some disabilities will find it challenging to use the system or website. Some incorrect results are encountered which could significantly impact proper operation. An immediate fix is highly recommended.
- **S-3 Medium** – Users can proceed with limited functionality. They might encounter an issue within a particular feature or workflow, but there is a straightforward workaround that exists.
- **S-4 Low** – A minor issue that will not affect the operation or **usability** of the system. This may include minor usability issues or functional deviations that can be avoided.

Priority

The Priority should be related to the Severity, and assigned by the product owner, or website manager. In contrast to the Severity, which describes the issue in isolation, setting the Priority also considers the context of all the issues and how quickly the issues could be fixed. For example, there may be more S-1 Critical issues identified than can be reasonably fixed in a set amount of time (e.g., a sprint). The product owner may prioritize some S-1 Critical issues over others based on IT capacity and feedback from leadership.

Here's an example of a Priority scale that mirrors the Severity scale above:

- P-1 Critical
- P-2 High

- P-3 Medium
- P-4 Low

Use your judgment

Ultimately, it is up to you to use your best judgement to assign severity and priority to each issue, based on how users will be impacted and how important each feature is.

15.

TESTING TOOLS

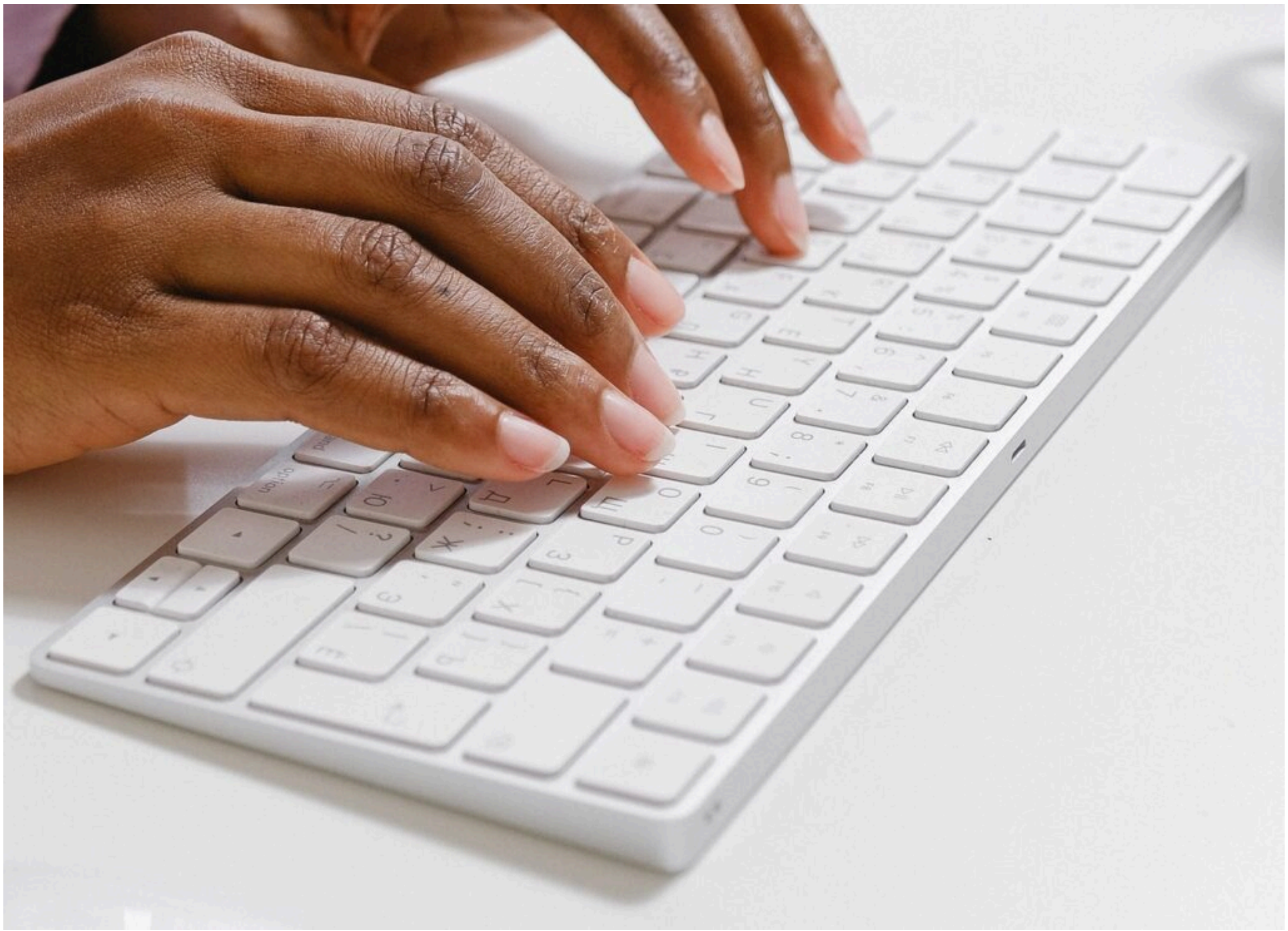
There are a variety of tools available to test the accessibility of your webpage and other types of content (e.g. documents, PowerPoint slides). The next few pages will describe these tools and how you can use them to test for accessibility. Some testing tools, like a keyboard, colour contrast checker, and screen reader, are relevant tools to use in all types of accessibility reviews. Other tools are technology-specific, like document checkers, web browser checkers, and Microsoft’s Accessibility Insights (for checking Windows apps).

Test page

The World Wide Web Consortium (**W3C**) has provided a Test page with several defects. It makes an excellent testing sandbox that can be used to try out all of the website-related testing tools above.

KEYBOARD TESTING

A good way to start your accessible testing is with a keyboard. It is equipment you already have and are familiar with. For some people with disabilities, they exclusively use their keyboard and don’t use a mouse at all. Therefore, **all webpage features would need to be keyboard accessible**, meaning someone should not need a mouse to interact with your webpage.



Keyboard commands

All internet browsers include these keyboard commands:

- Tab goes to the next clickable element (link or form control), Shift+Tab goes to the previous one.
- When on a link, pressing Enter will click it.

- When on a button, pressing Enter or Space will click it.
- When on a radio button or checkbox, pressing Up/Down or Left/Right arrow keys will cycle through all the elements in the group.

Hover state vs. Focus state

One way that a webpage highlights where a user is on the page is by using visual cues. These visual cues can be called a hover state for mouse navigation or a focus state for keyboard navigation.

A **hover** state – is a visual highlight feature that appears on a clickable element when a user moves their **mouse** pointer over it.

A **focus** state – is a visual highlight feature that appears on a clickable element when a user tabs to it (with their **keyboard**).

An easily recognizable and visible **focus state** helps sighted users that only use a keyboard to navigate and interact with a webpage.

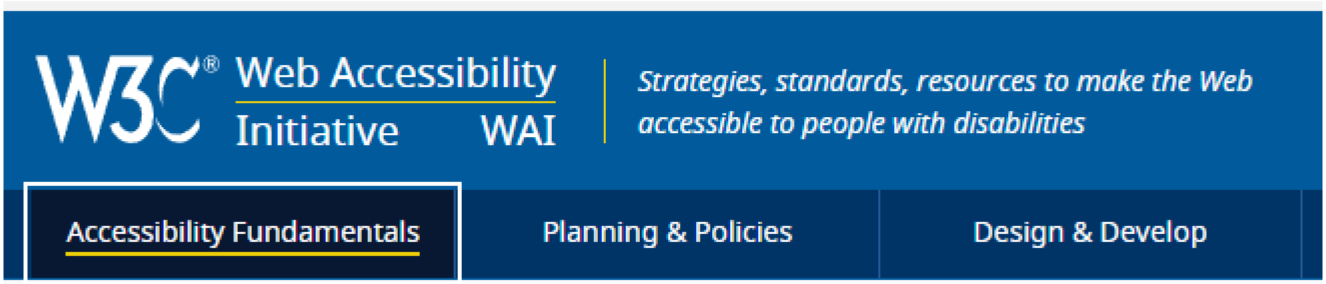
When designers/developers customize the focus state, it can make it more difficult to recognize easily. Therefore, it's especially critical to test the focus state on all clickable elements if elements have been customized.

Visible focus indicator

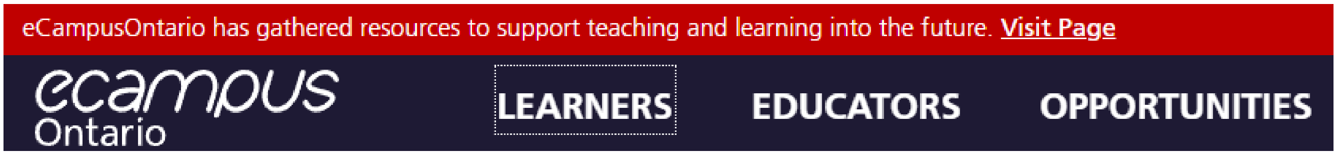
Specific instances of a focus state can be called a focus indicator. The result is that when a link or button receives focus (by the user tabbing to it), a visible focus indicator is displayed. A visible focus indicator is any dotted line, highlight, or visible marker that shows which clickable element is in focus and actionable.

Examples of a visible focus indicator:

- Dotted (or solid) line around a link:

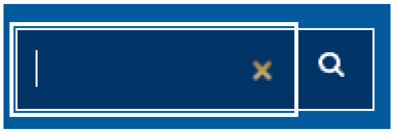


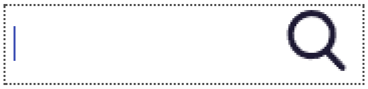
Strategies, standards, resources to make the Web accessible to people with disabilities



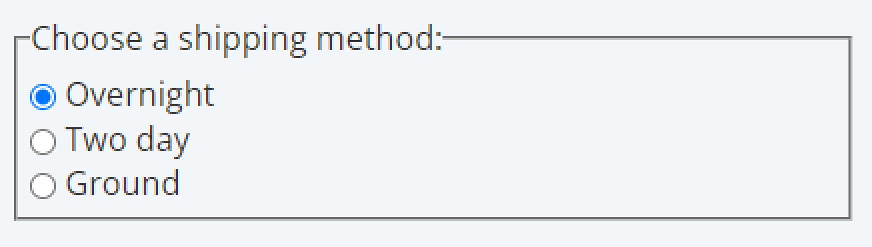
eCampusOntario has gathered resources to support teaching and learning into the future. Visit Page

- Cursor inside a text box:



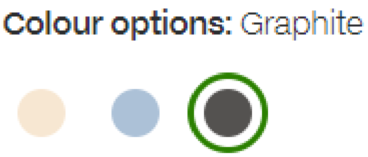


- Styling that shows you which radio button you've selected:



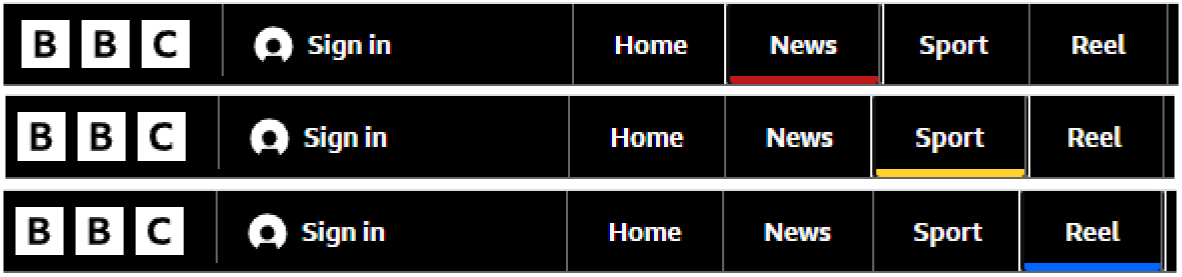
Choose a shipping method:

● Overnight
○ Two day
○ Ground



Colour options: Graphite

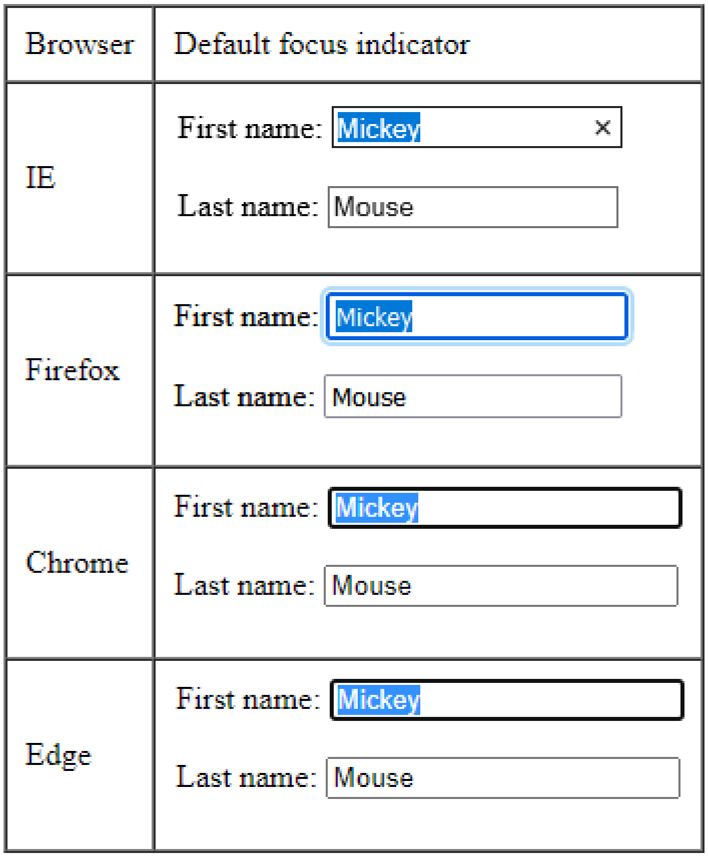
- An underline or other marker:



Acceptable visible focus indicators

Currently, Web Content Accessibility Guidelines (WCAG) 2.1 only requires that a focus indicator be visible (2.4.7: Focus Visible). WCAG 2.2 (currently in draft) will provide new, additional success criteria related to what the focus indicator looks like (e.g., size, colour contrast). While these are not current requirements, it is a good idea to prepare for this improvement.

Browsers vary in how strong their focus indicators are, although recent browser versions have been improving. In the table below, Microsoft Internet Explorer has a very poor focus indicator. In comparison, the other browsers have strong focus indicators – Mozilla Firefox has a double border, and Google Chrome and Microsoft Edge have dark borders.



Browser	Default focus indicator
IE	First name: Mickey × Last name: Mouse
Firefox	First name: Mickey Last name: Mouse
Chrome	First name: Mickey Last name: Mouse
Edge	First name: Mickey Last name: Mouse

Test cases

When conducting keyboard accessibility testing, here are some recommended test cases:

Step	Test	Pass	Fail
1	Use Tab to navigate around the page, and verify that you can reach all clickable elements.	All clickable elements can be reached.	Some clickable elements can't be reached.
2	As you tab around the page, all clickable elements must show a visible focus indicator.	All clickable elements show a visible focus indicator.	Some clickable elements do not show a visible focus indicator.
3	All clickable elements must be reached in a logical order (top to bottom, left to right).	All clickable elements are reached in a logical order.	Some clickable elements are not reached in a logical order.
4	Tab to some links and press Enter.	Links function as though they were clicked.	Links do not function as though they were clicked.
5	When you've reached a radio button or checkbox, press the Up/Down arrow keys to cycle through them.	You were able to reach all the elements in the group.	You were not able to reach all the elements in the group.
6	When you've reached a submit button, press Enter or Space to "click" it.	Submit button was clicked.	Submit button was not clicked.

Table 1: Recommended keyboard accessibility test cases

Resources:

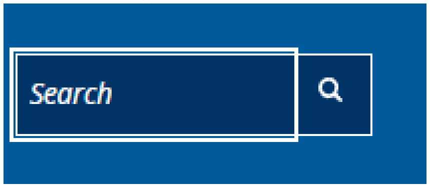
- WCAG Success Criterion 2.4.7: Focus Visible
- WCAG Success Criterion 2.4.11: Focus Appearance (Minimum)

CHROME DEVTOOLS

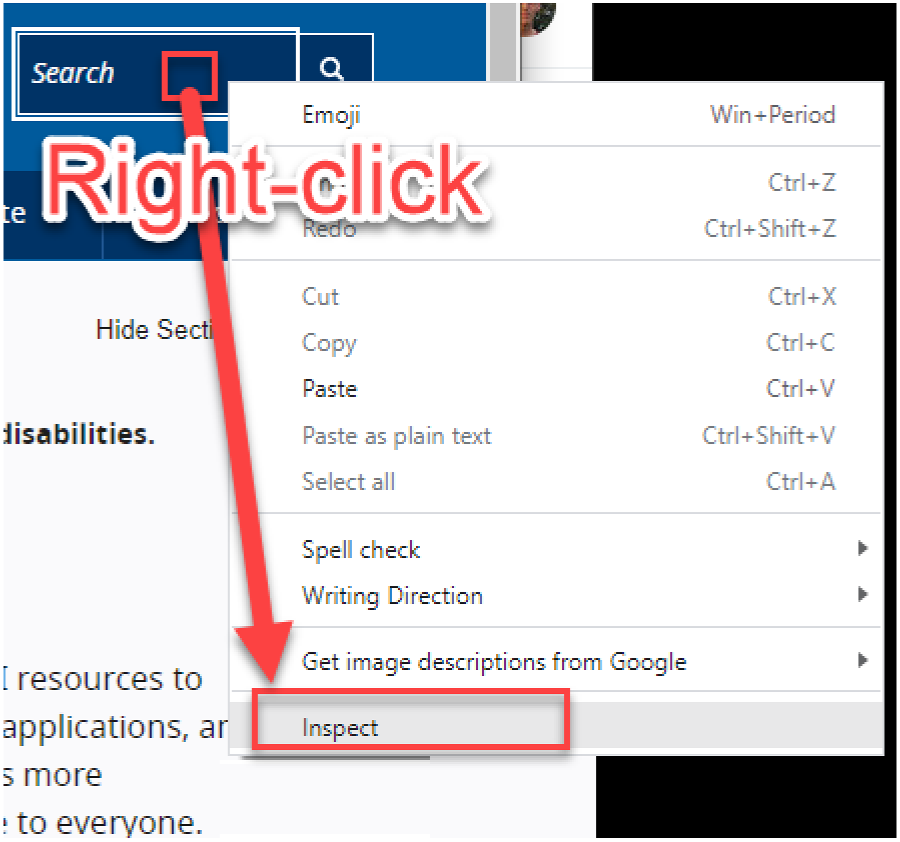
Chrome **DevTools** (developer tools) is a feature of the Chrome browser that can be used to display the HTML code on the current page. It can also be useful in finding accessibility issues, determining their causes, and testing fixes.

How to inspect a page element

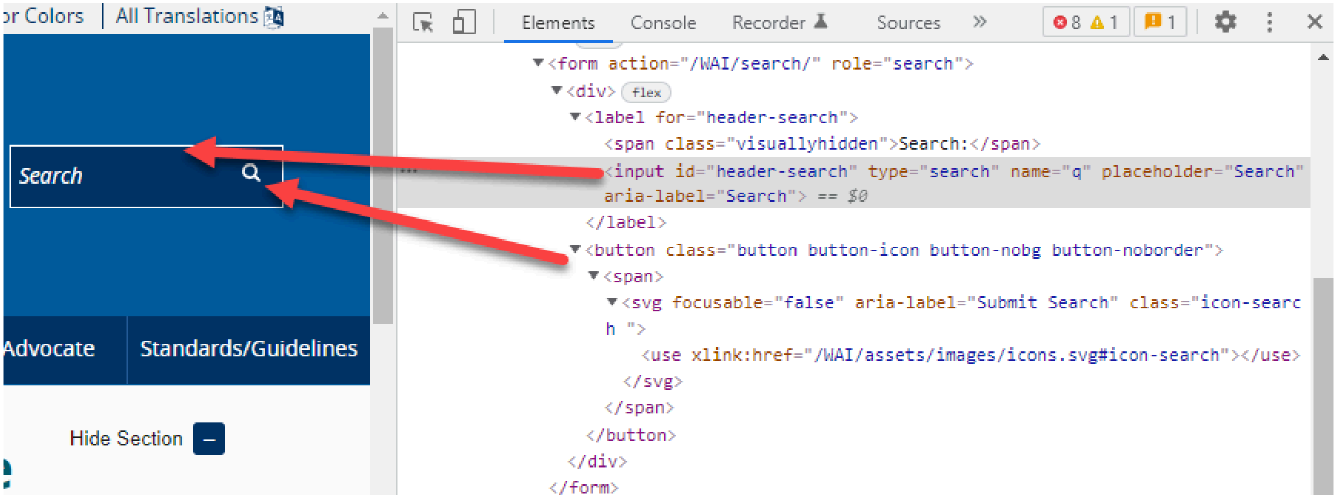
To reveal the HTML code related to a specific page element, you can right-click the element and select Inspect. As an example, let's consider this Search field.



You can right-click on the Search field...and then select Inspect.



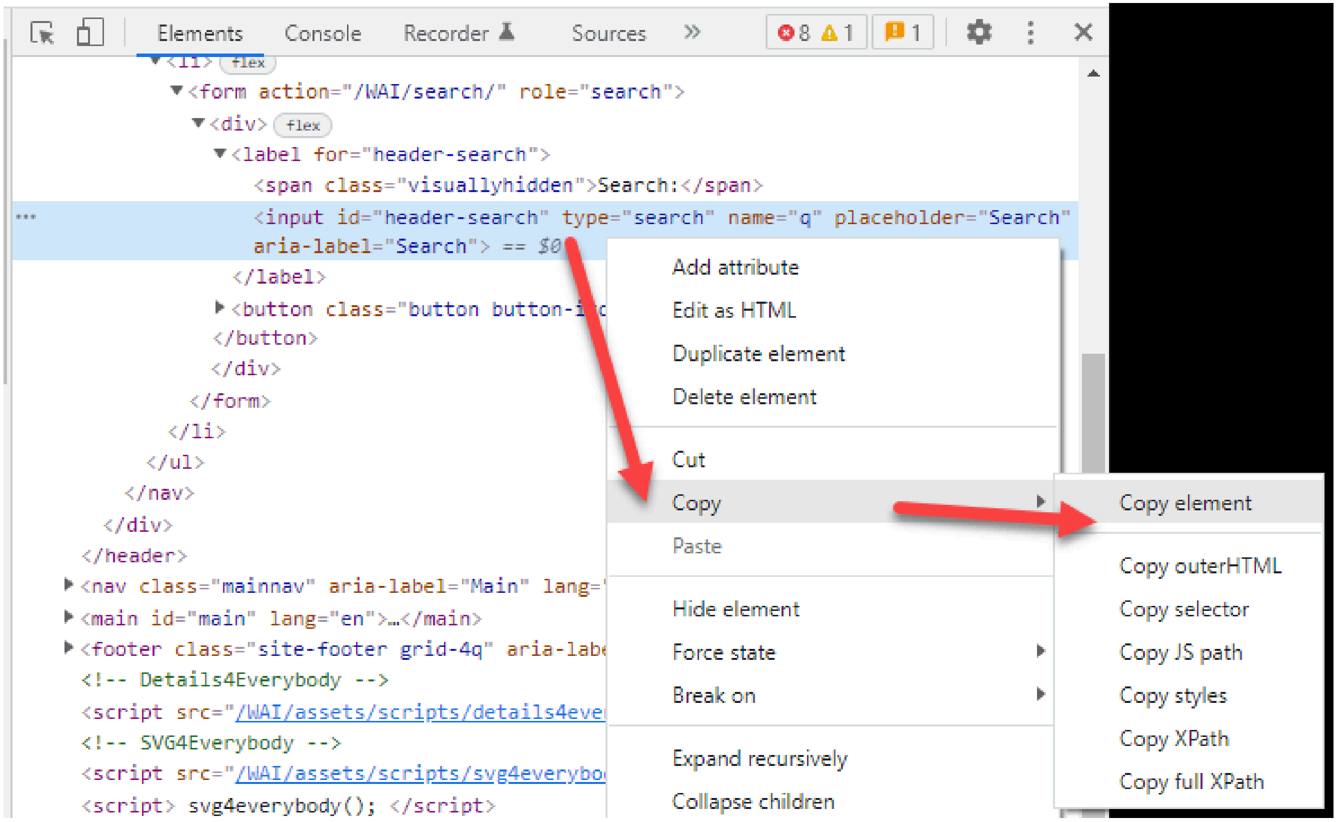
DevTools then opens as a panel beside the web page. It displays the current HTML code in the page and highlights the specific code related to the element you right-clicked on.



```
▼<form action="/WAI/search/" role="search">
  ▼<div> flex
    ▼<label for="header-search">
        <span class="visuallyhidden">Search:</span>
        <input id="header-search" type="search" name="q" placeholder="Search"
        aria-label="Search"> == $0
      </label>
    ▼<button class="button button-icon button-nobg button-noborder">
      ▼<span>
        ▼<svg focusable="false" aria-label="Submit Search" class="icon-searc
          h ">
            <use xlink:href="/WAI/assets/images/icons.svg#icon-search"></use>
          </svg>
        </span>
      </button>
    </div>
  </form>
```

How to copy code

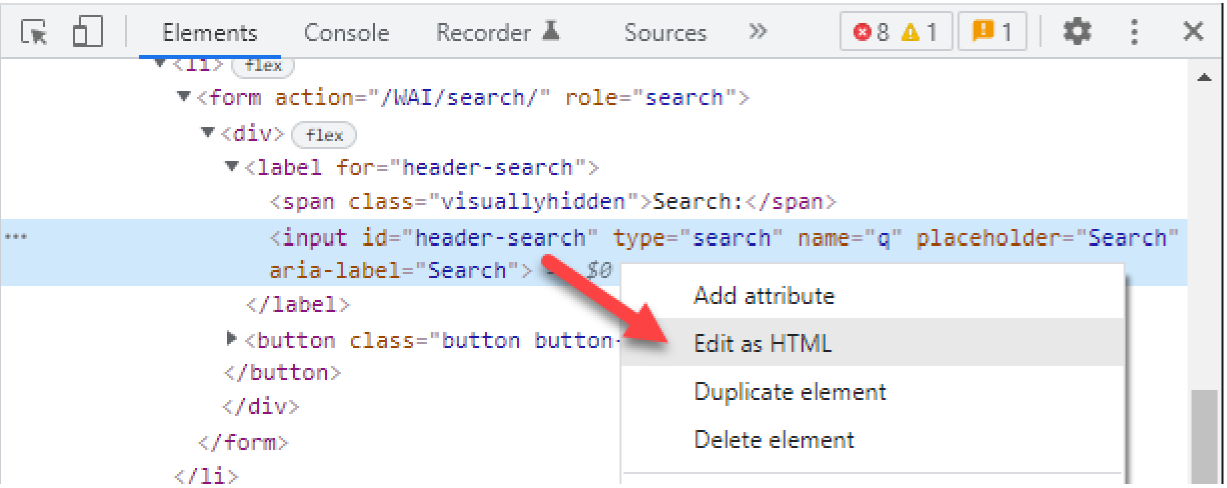
When reporting an accessibility issue, it's a good idea to include the code for the element with the defect. To do this, right-click on the line of code, and select Copy and Copy element. Now you can paste this code into your audit report.



```
          ▼<li> flex
            ▼<form action="/WAI/search/" role="search">
              ▼<div> flex
                ▼<label for="header-search">
                    <span class="visuallyhidden">Search:</span>
                    <input id="header-search" type="search" name="q" placeholder="Search"
                    aria-label="Search"> == $0
                  </label>
                ▶<button class="button button-ic
                  </button>
                </div>
              </form>
            </li>
          </ul>
        </nav>
      </div>
    </header>
  ▶<nav class="mainnav" aria-label="Main" lang="
  ▶<main id="main" lang="en">…</main>
  ▶<footer class="site-footer grid-4q" aria-labe
    <!-- Details4Everybody -->
    <script src="/WAI/assets/scripts/details4eve
    <!-- SVG4Everybody -->
    <script src="/WAI/assets/scripts/svg4everybo
    <script> svg4everybody(); </script>
```

How to edit code

You can also test code fixes right in the browser. Just right-click on the code in question, select Edit as HTML, and then make any changes you like. Once the code is changed, you can re-test the element. Note: this only changes the code temporarily. Your changes will be erased if you refresh the page or navigate to another page and back again.



```
▼<li> flex
  ▼<form action="/WAI/search/" role="search">
    ▼<div> flex
      ▼<label for="header-search">
          <span class="visuallyhidden">Search:</span>
          <input id="header-search" type="search" name="q" placeholder="Search"
          aria-label="Search"> $0
        </label>
      ▶<button class="button button-
      </button>
      </div>
    </form>
  </li>
```

Resources: Chrome DevTools

COLOUR CONTRAST TESTING

Good colour contrast in your content is important for people with vision impairments. It can make the difference between perceiving and interpreting your content (visuals and text) and not.

Colour Contrast

Colour contrast measures the difference in brightness between foreground and background colours – usually the colour of text and the background behind it. Web Content Accessibility Guidelines (WCAG) 2.1 colour contrast guidelines require the following:

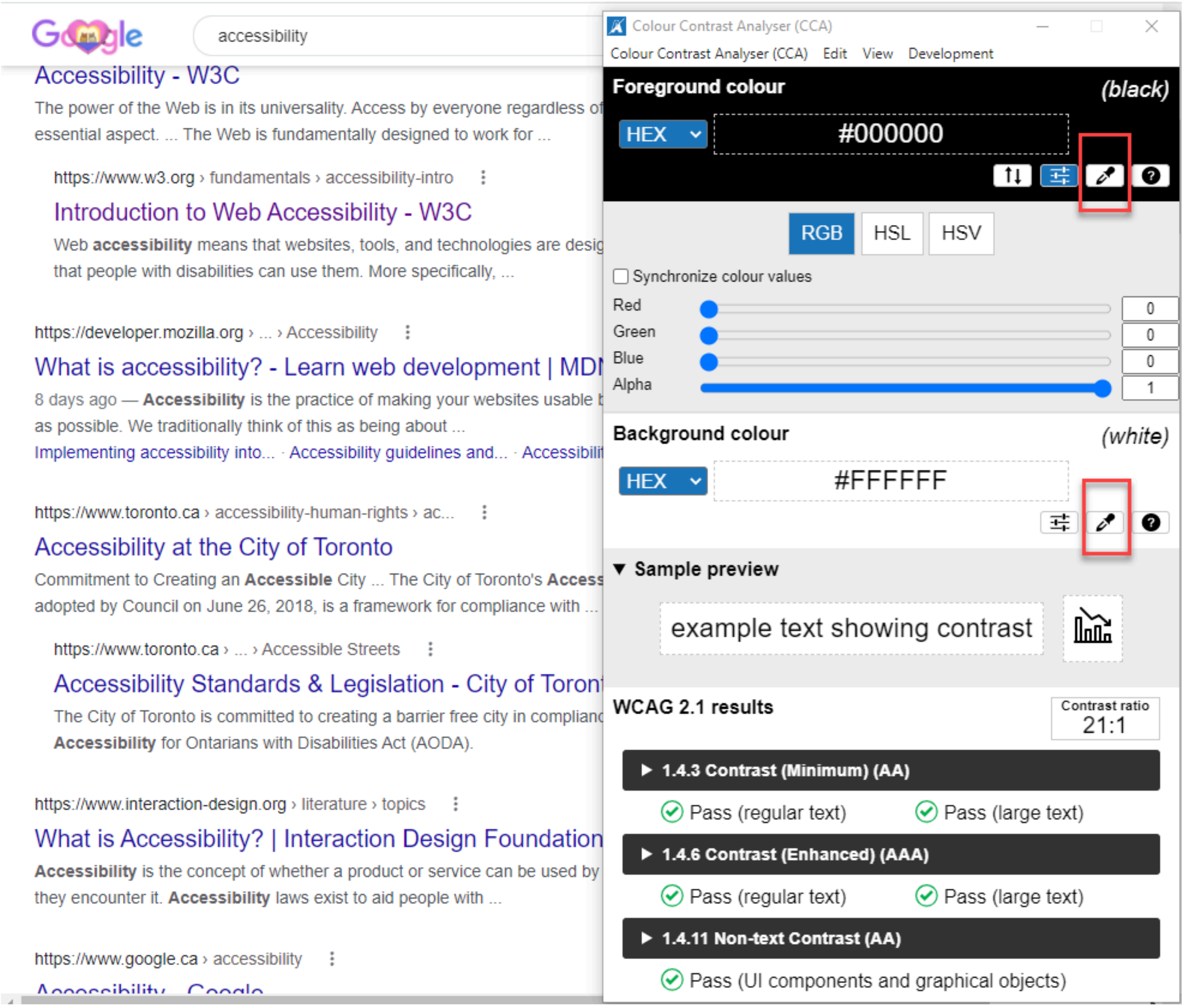
- Large text must have a colour contrast of 3.0:1 or higher. (Large text is defined as any of: 1.5em, 18pt, 24px, 1.2em bold, 14pt bold, 19px bold, or larger.)
- Small text must have a colour contrast of 4.5:1 or higher.
- Icons must have a colour contrast of 3.0:1 or higher.
- Logos are exempt from colour contrast guidelines.

There are two common colour contrast assessment tools that are explained below: TPGi Colour Contrast Analyser and Chrome Color Picker.

TPGi Colour Contrast Analyser

The TPGi Colour Contrast Analyser (CCA) can be used to sample any colours you can see on your screen. As a result, it can be used to test any kind of digital content, including web pages, Word documents, PDFs, etc. To use it, do the following:

1. CCA has two eyedroppers – one to sample foreground colours and one to sample background colours.


Google
accessibility
Colour Contrast Analyser (CCA)
Colour Contrast Analyser (CCA) Edit View Development
RGB HSL HSV
Sample preview
example text showing contrast

Accessibility - W3C

The power of the Web is in its universality. Access by everyone regardless of essential aspect. ... The Web is fundamentally designed to work for ...

https://www.w3.org › fundamentals › accessibility-intro

Introduction to Web Accessibility - W3C

Web **accessibility** means that websites, tools, and technologies are desig that people with disabilities can use them. More specifically, ...

https://developer.mozilla.org › ... › Accessibility

What is accessibility? - Learn web development | MDI

8 days ago — **Accessibility** is the practice of making your websites usable b as possible. We traditionally think of this as being about ...

Implementing accessibility into... · Accessibility guidelines and... · Accessibilit

https://www.toronto.ca › accessibility-human-rights › ac...

Accessibility at the City of Toronto

Commitment to Creating an **Accessible** City ... The City of Toronto's **Access** adopted by Council on June 26, 2018, is a framework for compliance with ...

https://www.toronto.ca › ... › Accessible Streets

Accessibility Standards & Legislation - City of Toront

The City of Toronto is committed to creating a barrier free city in complianc **Accessibility** for Ontarians with Disabilities Act (AODA).

https://www.interaction-design.org › literature › topics

What is Accessibility? | Interaction Design Foundation

Accessibility is the concept of whether a product or service can be used by they encounter it. **Accessibility** laws exist to aid people with ...

https://www.google.ca › accessibility

Accessibility - Google

Foreground colour (black)

HEX #000000

□ Synchronize colour values

Red 0

Green 0

Blue 0

Alpha 1

Background colour (white)

HEX #FFFFFF

WCAG 2.1 results

Contrast ratio 21:1

1.4.3 Contrast (Minimum) (AA)

✓ Pass (regular text) ✓ Pass (large text)

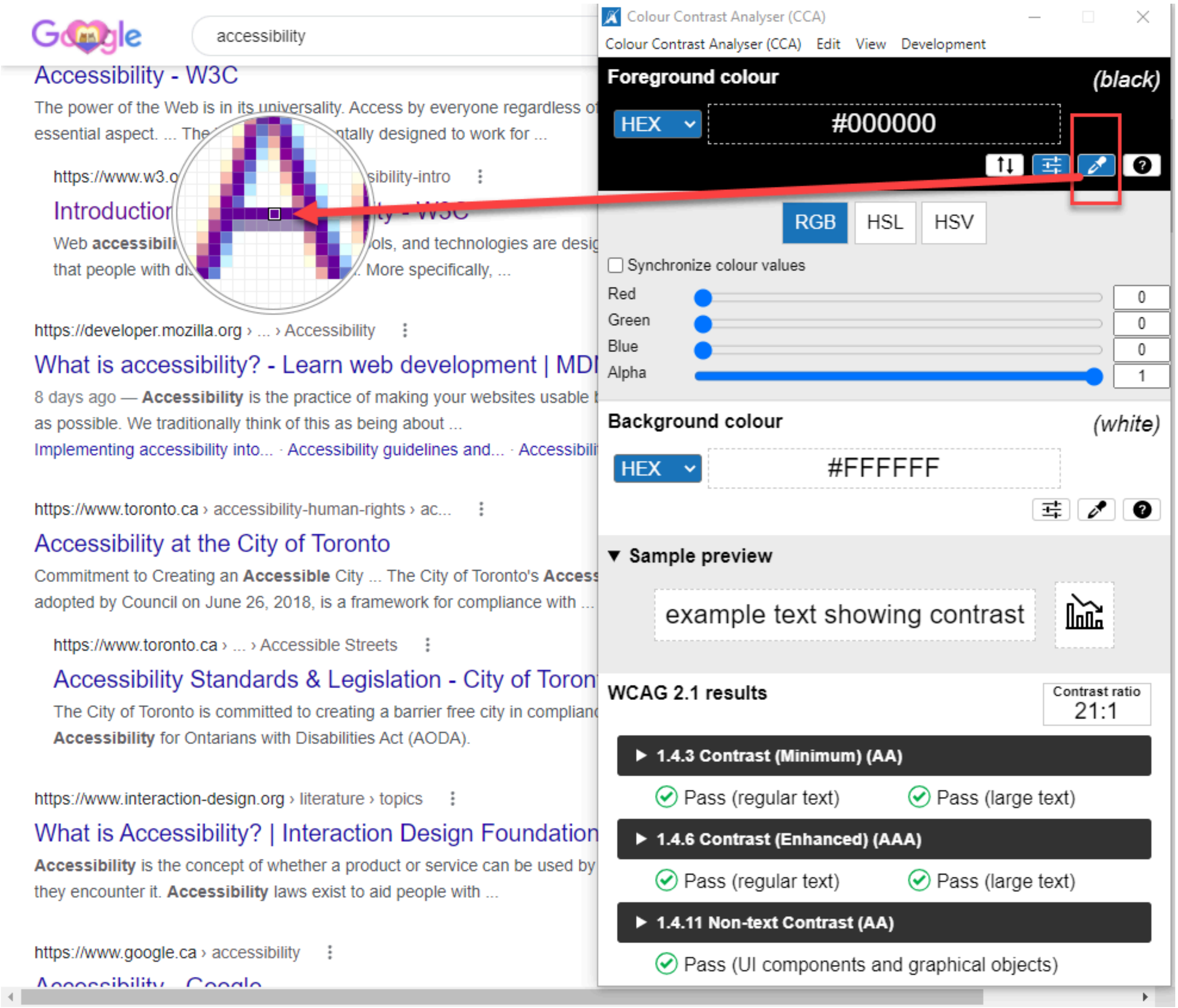
1.4.6 Contrast (Enhanced) (AAA)

✓ Pass (regular text) ✓ Pass (large text)

1.4.11 Non-text Contrast (AA)

✓ Pass (UI components and graphical objects)

2. Select the first eyedropper, and then click on the darkest pixel (that you can find) in the text.



Google
accessibility
Colour Contrast Analyser (CCA)
Colour Contrast Analyser (CCA) Edit View Development
RGB
HSL
HSV
Sample preview
example text showing contrast

Accessibility - W3C

The power of the Web is in its universality. Access by everyone regardless o essential aspect. ... The ntally designed to work for ...

https://www.w3.o sibility-intro

Introductior ty - W3C

Web accessibili ols, and technologies are desig that people with di . More specifically, ...

https://developer.mozilla.org › ... › Accessibility

What is accessibility? - Learn web development | MDI

8 days ago — **Accessibility** is the practice of making your websites usable t as possible. We traditionally think of this as being about ...

Implementing accessibility into... · Accessibility guidelines and... · Accessibili

https://www.toronto.ca › accessibility-human-rights › ac...

Accessibility at the City of Toronto

Commitment to Creating an **Accessible** City ... The City of Toronto's **Access** adopted by Council on June 26, 2018, is a framework for compliance with ...

https://www.toronto.ca › ... › Accessible Streets

Accessibility Standards & Legislation - City of Toron

The City of Toronto is committed to creating a barrier free city in complianc **Accessibility** for Ontarians with Disabilities Act (AODA).

https://www.interaction-design.org › literature › topics

What is Accessibility? | Interaction Design Foundation

Accessibility is the concept of whether a product or service can be used by they encounter it. **Accessibility** laws exist to aid people with ...

https://www.google.ca › accessibility

Accessibility - Google

Foreground colour *(black)*

HEX #000000

☐ Synchronize colour values

Red 0

Green 0

Blue 0

Alpha 1

Background colour *(white)*

HEX #FFFFFF

WCAG 2.1 results

Contrast ratio 21:1

▶ 1.4.3 Contrast (Minimum) (AA)

✓ Pass (regular text) ✓ Pass (large text)

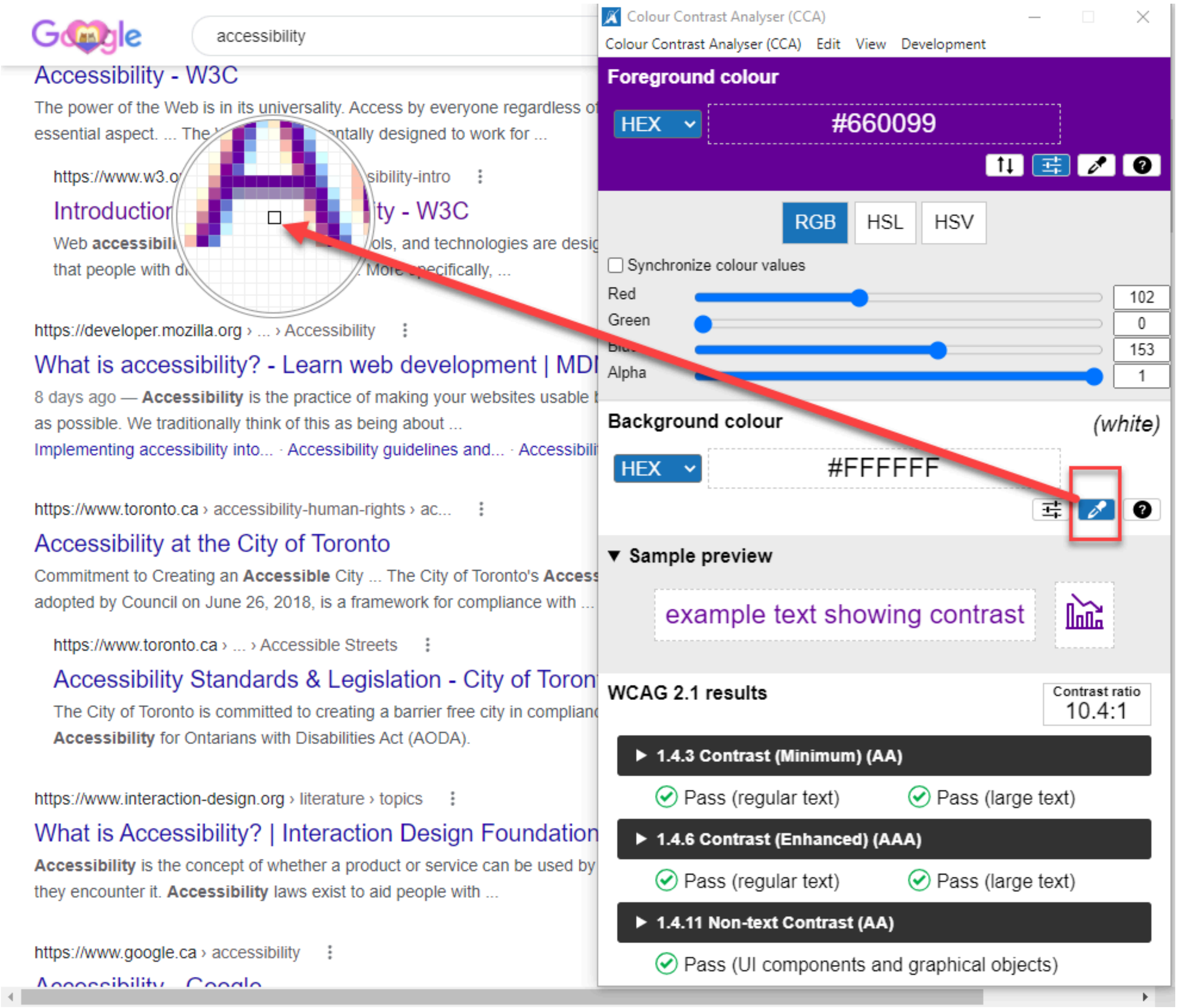
▶ 1.4.6 Contrast (Enhanced) (AAA)

✓ Pass (regular text) ✓ Pass (large text)

▶ 1.4.11 Non-text Contrast (AA)

✓ Pass (UI components and graphical objects)

3. Select the second eyedropper, and then sample the background colour (white space in this case).



accessibility

Accessibility - W3C

The power of the Web is in its universality. Access by everyone regardless o

essential aspect. ... The ntally designed to work for ...

https://www.w3.o sibility-intro

Introductio ty - W3C

Web accessibil ols, and technologies are desig

that people with d More specifically, ...

https://developer.mozilla.org › ... › Accessibility

What is accessibility? - Learn web development | MD

8 days ago — **Accessibility** is the practice of making your websites usable

as possible. We traditionally think of this as being about ...

Implementing accessibility into... · Accessibility guidelines and... · Accessibili

https://www.toronto.ca › accessibility-human-rights › ac...

Accessibility at the City of Toronto

Commitment to Creating an **Accessible** City ... The City of Toronto's **Access**

adopted by Council on June 26, 2018, is a framework for compliance with ...

https://www.toronto.ca › ... › Accessible Streets

Accessibility Standards & Legislation - City of Toron

The City of Toronto is committed to creating a barrier free city in complian

Accessibility for Ontarians with Disabilities Act (AODA).

https://www.interaction-design.org › literature › topics

What is Accessibility? | Interaction Design Foundation

Accessibility is the concept of whether a product or service can be used by

they encounter it. **Accessibility** laws exist to aid people with ...

https://www.google.ca › accessibility

Colour Contrast Analyser (CCA)

Foreground colour

HEX: #660099

RGB

☐ Synchronize colour values

Red	102
Green	0
Blue	153
Alpha	1

Background colour (white)

HEX: #FFFFFF

Sample preview

example text showing contrast

WCAG 2.1 results

Contrast ratio 10.4:1

1.4.3 Contrast (Minimum) (AA)

✓ Pass (regular text) ✓ Pass (large text)

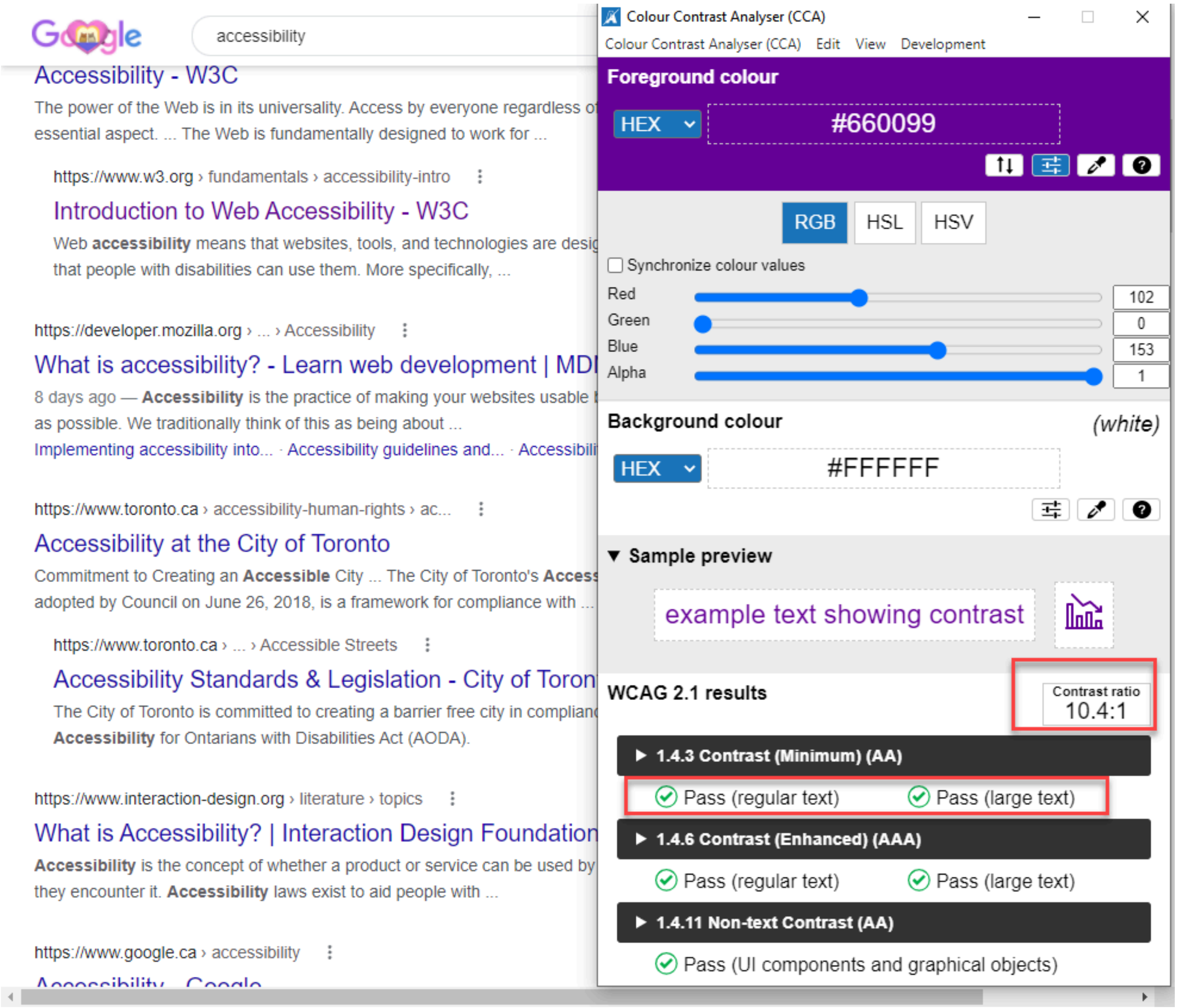
1.4.6 Contrast (Enhanced) (AAA)

✓ Pass (regular text) ✓ Pass (large text)

1.4.11 Non-text Contrast (AA)

✓ Pass (UI components and graphical objects)

4. CCA will display the contrast ratio.



accessibility

Accessibility - W3C

The power of the Web is in its universality. Access by everyone regardless o
essential aspect. ... The Web is fundamentally designed to work for ...

https://www.w3.org › fundamentals › accessibility-intro

Introduction to Web Accessibility - W3C

Web **accessibility** means that websites, tools, and technologies are desig
that people with disabilities can use them. More specifically, ...

https://developer.mozilla.org › ... › Accessibility

What is accessibility? - Learn web development | MDI

8 days ago — **Accessibility** is the practice of making your websites usable
as possible. We traditionally think of this as being about ...
Implementing accessibility into... · Accessibility guidelines and... · Accessibili

https://www.toronto.ca › accessibility-human-rights › ac...

Accessibility at the City of Toronto

Commitment to Creating an **Accessible** City ... The City of Toronto's **Access**
adopted by Council on June 26, 2018, is a framework for compliance with ...

https://www.toronto.ca › ... › Accessible Streets

Accessibility Standards & Legislation - City of Toron

The City of Toronto is committed to creating a barrier free city in complian
Accessibility for Ontarians with Disabilities Act (AODA).

https://www.interaction-design.org › literature › topics

What is Accessibility? | Interaction Design Foundation

Accessibility is the concept of whether a product or service can be used by
they encounter it. **Accessibility** laws exist to aid people with ...

https://www.google.ca › accessibility

Colour Contrast Analyser (CCA)

Colour Contrast Analyser (CCA) Edit View Development

Foreground colour

HEX #660099

RGB HSL HSV

□ Synchronize colour values

Red 102
Green 0
Blue 153
Alpha 1

Background colour (white)

HEX #FFFFFF

▼ Sample preview

example text showing contrast

WCAG 2.1 results

Contrast ratio 10.4:1

▶ 1.4.3 Contrast (Minimum) (AA)

✓ Pass (regular text) ✓ Pass (large text)

▶ 1.4.6 Contrast (Enhanced) (AAA)

✓ Pass (regular text) ✓ Pass (large text)

▶ 1.4.11 Non-text Contrast (AA)

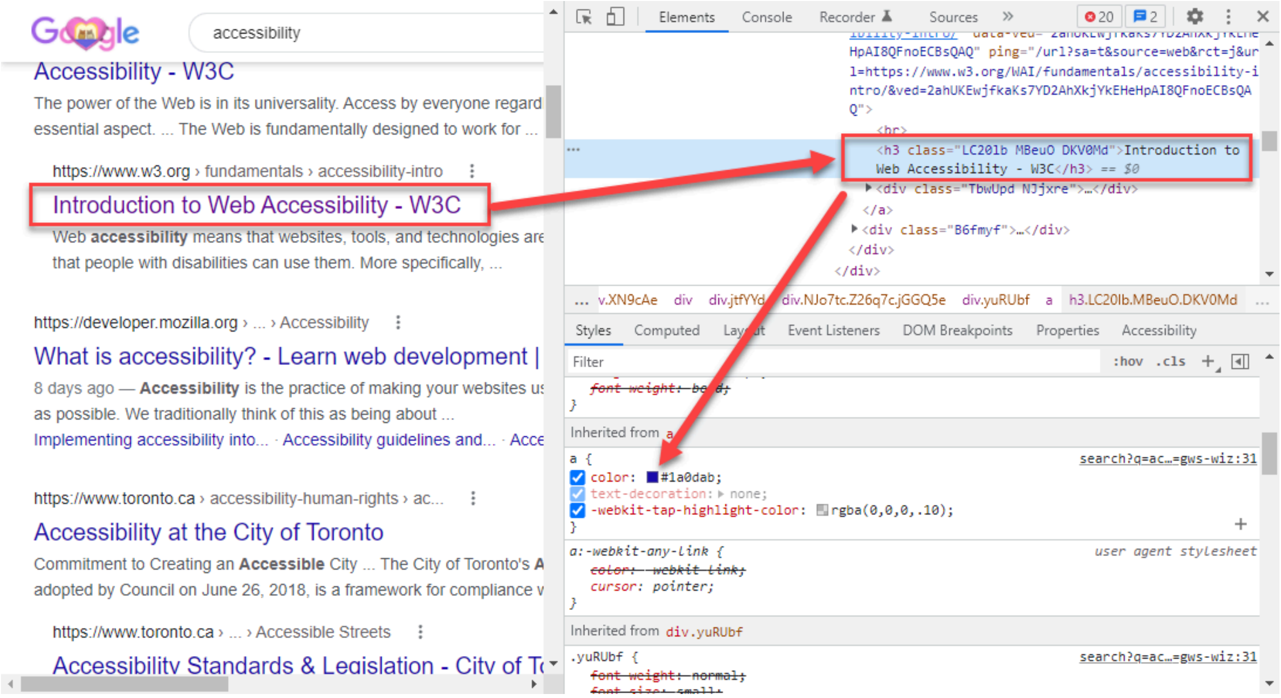
✓ Pass (UI components and graphical objects)

In this example, the colour contrast is 10.4:1 and clearly passes the minimum colour contrast as set out by WCAG 2.1.

Chrome "Color Picker"

In the Chrome browser, the **DevTools** panel includes a "Color Picker" feature. To use it, do the following:

1. Find the element (text or icon) you'd like to test, right-click on it, and select Inspect.
2. In the DevTools panel, select the Elements tab (unless it's already selected).
3. Click the colour square next to the element's color value in the Styles pane.



Accessibility - W3C

The power of the Web is in its universality. Access by everyone regard
essential aspect. ... The Web is fundamentally designed to work for ...

https://www.w3.org › fundamentals › accessibility-intro

Introduction to Web Accessibility - W3C

Web **accessibility** means that websites, tools, and technologies are
that people with disabilities can use them. More specifically, ...

https://developer.mozilla.org › ... › Accessibility

What is accessibility? - Learn web development |

8 days ago — **Accessibility** is the practice of making your websites us
as possible. We traditionally think of this as being about ...
Implementing accessibility into... · Accessibility guidelines and... · Acce

https://www.toronto.ca › accessibility-human-rights › ac...

Accessibility at the City of Toronto

Commitment to Creating an **Accessible** City ... The City of Toronto's A
adopted by Council on June 26, 2018, is a framework for compliance v

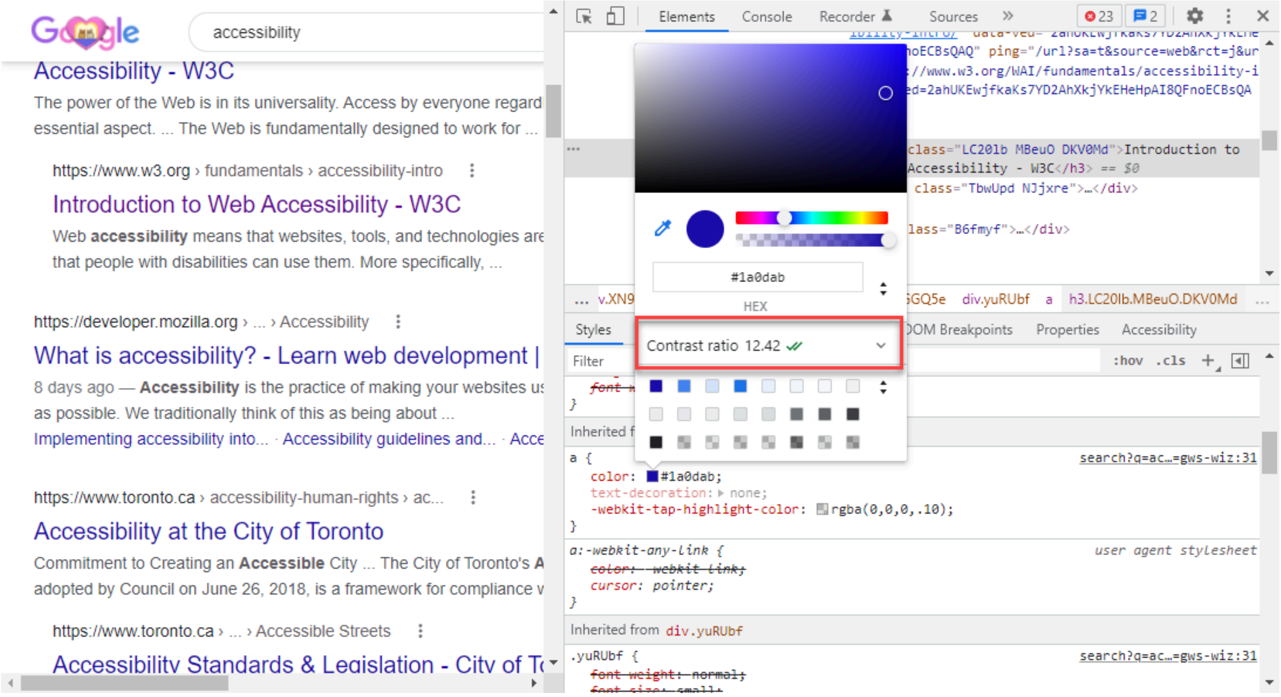
https://www.toronto.ca › ... › Accessible Streets

Accessibility Standards & Legislation - City of To

```
HpAI8QFnoECBsQAQ" ping="/url?sa=t&source=web&rct=j&ur
l=https://www.w3.org/WAI/fundamentals/accessibility-i
ntro/&ved=2ahUKEwjfkaKs7YD2AhXkjYKEHeHpAI8QFnoECBsQA
Q">
  <br>
  <h3 class="LC20lb MBeuO DKV0Md">Introduction to
  Web Accessibility - W3C</h3> == $0
  ▸<div class="TbwUpd NJjxre">…</div>
  </a>
 ▸<div class="B6fmyf">…</div>
 </div>
</div>
... v.XN9cAe  div  div.jtfYYd  div.NJo7tc.Z26q7c.jGGQ5e  div.yuRUbf  a  h3.LC20lb.MBeuO.DKV0Md
```

```
  ~~font-weight: bold;~~
}
Inherited from a
a {                                     search?q=ac…=gws-wiz:31
☑ color: ■#1a0dab;
☑ text-decoration: ▸ none;
☑ -webkit-tap-highlight-color: ▨rgba(0,0,0,.10);
}
a:-webkit-any-link {               user agent stylesheet
  ~~color: -webkit-link;~~
  cursor: pointer;
}
Inherited from div.yuRUbf
.yuRUbf {                               search?q=ac…=gws-wiz:31
  ~~font-weight: normal;~~
  ~~font-size: small;~~
```

4. Have a look at the Contrast ratio section of the Color Picker. It will display the element's contrast ratio. One checkmark means that the element meets the WCAG single A recommendation (4.5:1). Two checkmarks means that it meets the WCAG double A recommendation (7.0:1).



#1a0dab

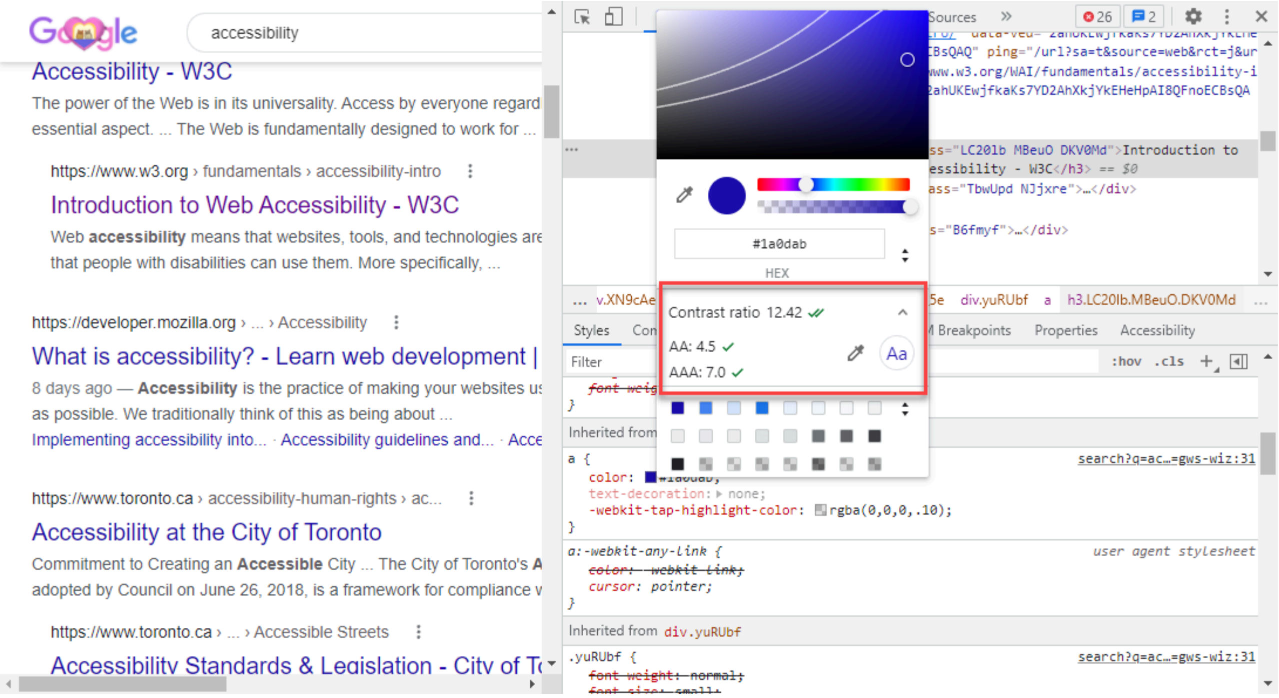
HEX

Contrast ratio 12.42 ✓✓

```
a {                                     search?q=ac…=gws-wiz:31
  color: ■#1a0dab;
  text-decoration: ▸ none;
  -webkit-tap-highlight-color: ▨rgba(0,0,0,.10);
}
a:-webkit-any-link {               user agent stylesheet
  ~~color: -webkit-link;~~
  cursor: pointer;
}
Inherited from div.yuRUbf
.yuRUbf {                               search?q=ac…=gws-wiz:31
  ~~font-weight: normal;~~
  ~~font-size: small;~~
```

5. You can also use the eyedroppers to test other colours you see on the page. The eyedropper for foreground colours is above the hex code. The eyedropper for background colours is in the collapsed

section under the Contrast ratio.



accessibility
Accessibility - W3C
The power of the Web is in its universality. Access by everyone regard
essential aspect. ... The Web is fundamentally designed to work for ...
https://www.w3.org › fundamentals › accessibility-intro
Introduction to Web Accessibility - W3C
Web accessibility means that websites, tools, and technologies are
that people with disabilities can use them. More specifically, ...
https://developer.mozilla.org › ... › Accessibility
What is accessibility? - Learn web development |
8 days ago — Accessibility is the practice of making your websites us
as possible. We traditionally think of this as being about ...
Implementing accessibility into... · Accessibility guidelines and... · Acce
https://www.toronto.ca › accessibility-human-rights › ac...
Accessibility at the City of Toronto
Commitment to Creating an Accessible City ... The City of Toronto's A
adopted by Council on June 26, 2018, is a framework for compliance v
https://www.toronto.ca › ... › Accessible Streets
Accessibility Standards & Legislation - City of To
Sources
26
2
#1a0dab
HEX
Contrast ratio 12.42 ✓
AA: 4.5 ✓
AAA: 7.0 ✓
Aa
Styles
Con
Filter
Breakpoints
Properties
Accessibility
:hov
.cls
Inherited from
Inherited from div.yuRUbf
user agent stylesheet
search?q=ac…=gws-wiz:31
search?q=ac…=gws-wiz:31
v.XN9cAe
div.yuRUbf
a
h3.LC20lb.MBeuO.DKV0Md

```
BsQAQ" ping="/url?sa=t&source=web&rct=j&ur
ww.w3.org/WAI/fundamentals/accessibility-i
ahUKEwjfkaKs7YD2AhXkjYkEHeHpAI8QFnoECBsQA
ss="LC20lb MBeuO DKV0Md">Introduction to
essibility - W3C</h3> == $0
ass="TbwUpd NJjxre">…</div>
s="B6fmyf">…</div>

a {
  color: #1a0dab;
  text-decoration: none;
  -webkit-tap-highlight-color: rgba(0,0,0,.10);
}
a:-webkit-any-link {
  ~~color: -webkit-link;~~
  cursor: pointer;
}
.yuRUbf {
  ~~font-weight: normal;~~
  ~~font-size: small;~~
```

Resources:

- TPGi Colour Contrast Analyser (CCA)
- View the contrast ratio of a text element in the Color Picker

DEQUE AXE BROWSER EXTENSION TESTING

With the growing recognition of how important digital accessibility is, there are new tools emerging that can help you make your site more accessible. One of these tools is Deque's axe browser extension. Deque's axe browser extension is a free automated testing tool. It's useful when combined with manual testing, like keyboard-only checks, and screen reader testing.

Note: Deque also offers paid versions of axe that offer more automated testing features for workflows and a way to export all defects on a given page.

How to install axe in your browser

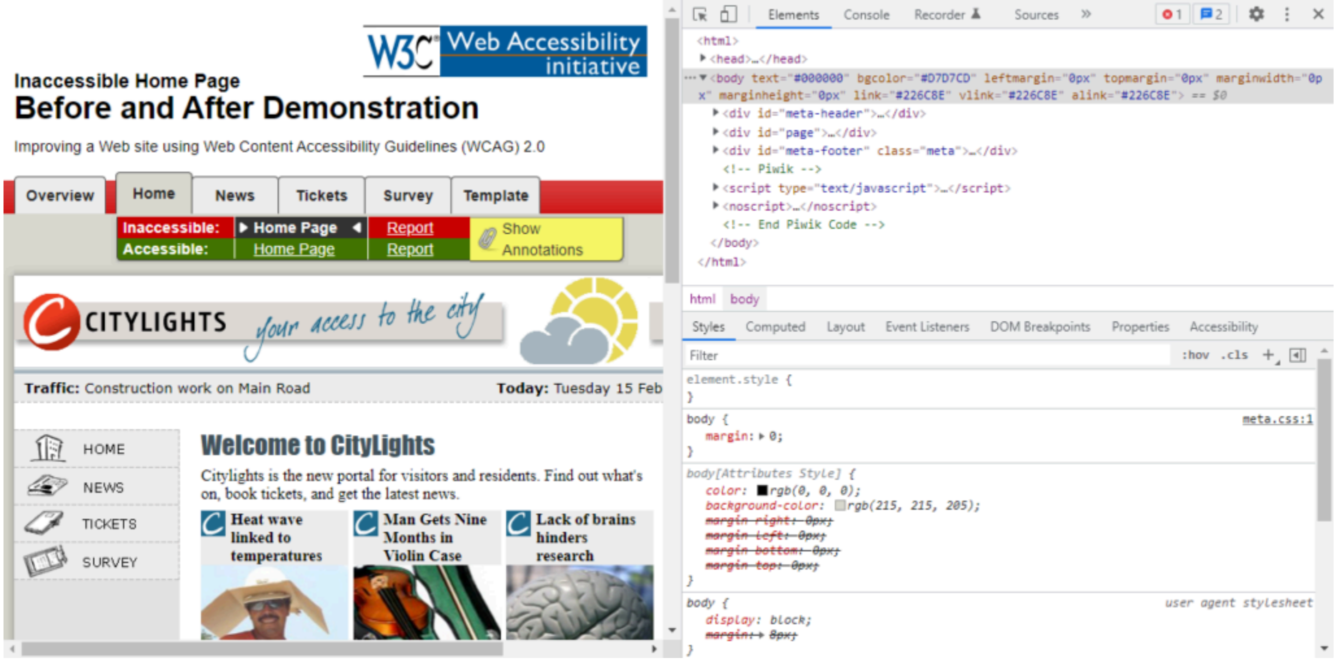
To install the browser extension, go to Deque's axe website and select the appropriate version for your specific browser (Chrome, Firefox, or Edge).

How to test with axe

The following example will test a **W3C** page that has plenty of defects – the URL is: https://www.w3.org/WAI/demos/bad/before/home.html. This website was purposefully designed with defects to show some common accessibility issues.

To test the above web page (or any web page you're currently viewing) in Chrome, follow these steps:

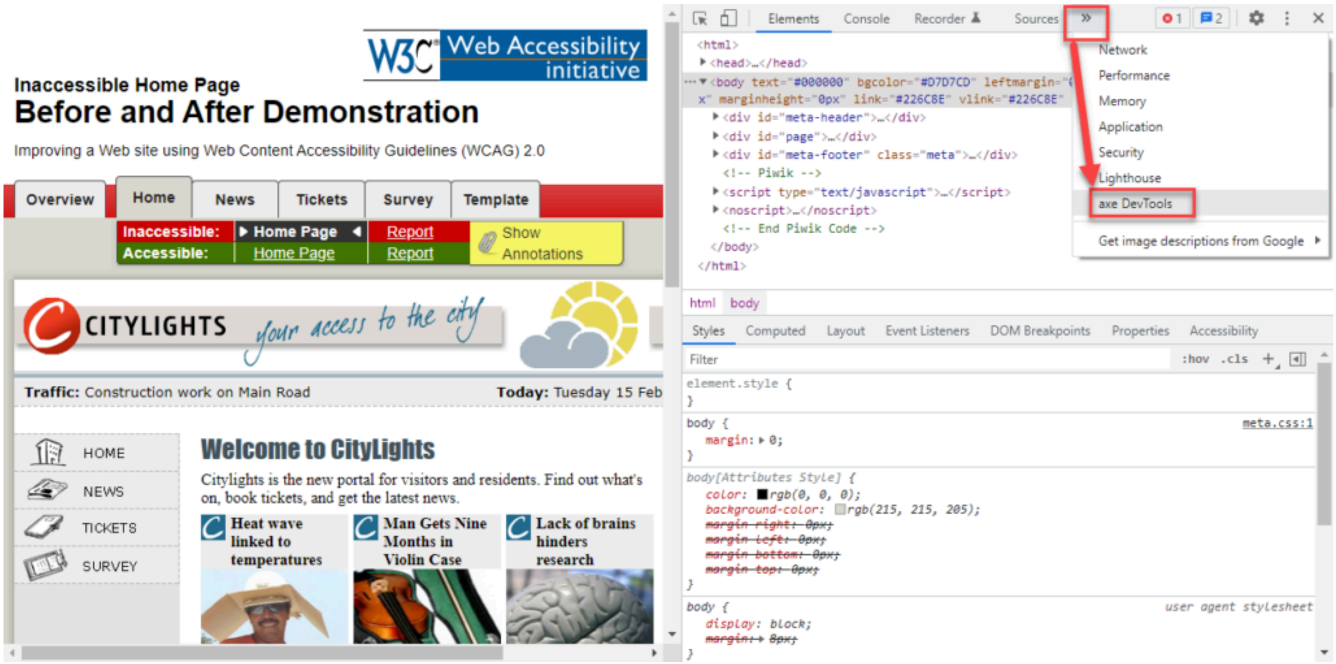
1. Open **DevTools** by pressing F12, or Control+Shift+I, or Command+Option+I (macOS). Make sure you're on the Elements tab.


W3C Web Accessibility initiative
Inaccessible Home Page
Before and After Demonstration
Improving a Web site using Web Content Accessibility Guidelines (WCAG) 2.0
Overview
Home
News
Tickets
Survey
Template
Inaccessible: Home Page Report
Accessible: Home Page Report
Show Annotations
CITYLIGHTS your access to the city
Traffic: Construction work on Main Road
Today: Tuesday 15 Feb
HOME
NEWS
TICKETS
SURVEY
Welcome to CityLights
Citylights is the new portal for visitors and residents. Find out what's on, book tickets, and get the latest news.
Heat wave linked to temperatures
Man Gets Nine Months in Violin Case
Lack of brains hinders research
Elements
Console
Recorder
Sources
»
Styles
Computed
Layout
Event Listeners
DOM Breakpoints
Properties
Accessibility
Filter
:hov .cls

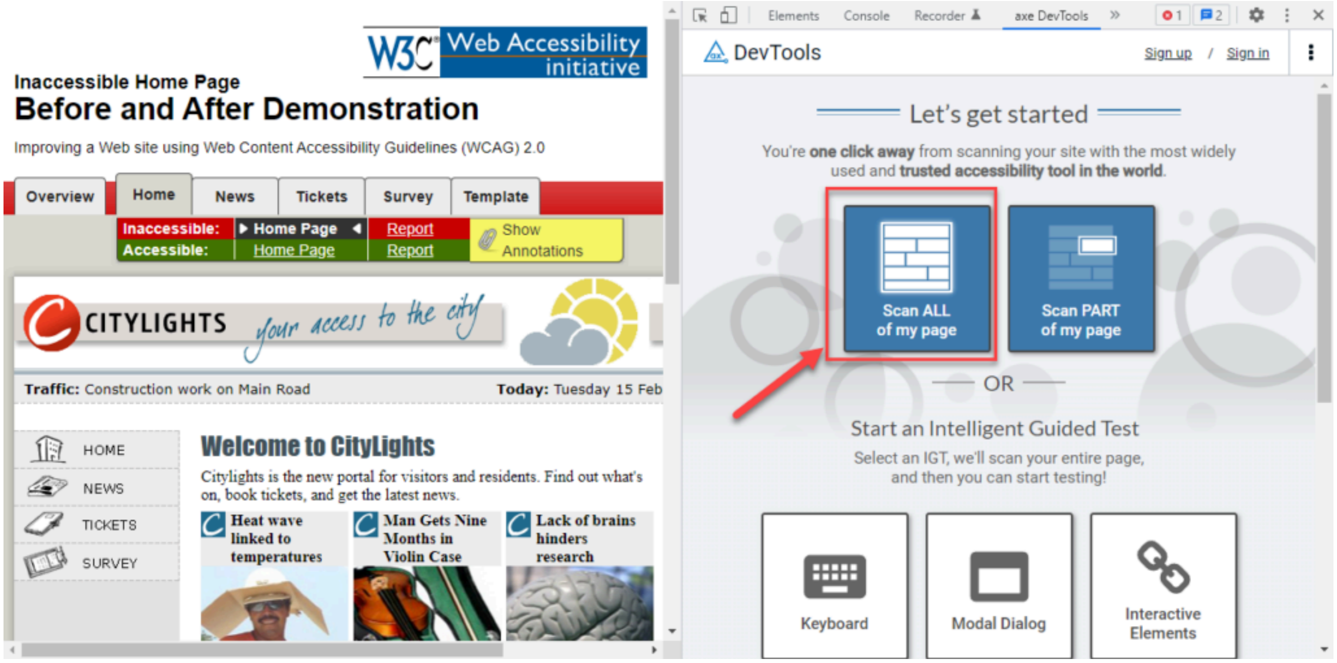
```
<html>
 ▸<head>…</head>
…▾<body text="#000000" bgcolor="#D7D7CD" leftmargin="0px" topmargin="0px" marginwidth="0px" marginheight="0px" link="#226C8E" vlink="#226C8E" alink="#226C8E"> == $0
   ▸<div id="meta-header">…</div>
   ▸<div id="page">…</div>
   ▸<div id="meta-footer" class="meta">…</div>
    <!-- Piwik -->
   ▸<script type="text/javascript">…</script>
   ▸<noscript>…</noscript>
    <!-- End Piwik Code -->
  </body>
</html>
```

```
element.style {
}
body {                                   meta.css:1
  margin: ▸ 0;
}
body[Attributes Style] {
  color: ■rgb(0, 0, 0);
  background-color: □rgb(215, 215, 205);
  ~~margin-right: 0px;~~
  ~~margin-left: 0px;~~
  ~~margin-bottom: 0px;~~
  ~~margin-top: 0px;~~
}
body {                       user agent stylesheet
  display: block;
  ~~margin: ▸ 8px;~~
}
```

2. Click the "»" button (if you're using a **screen reader**, it's the "More tabs" button). This will open a submenu. Select "axe DevTools" from the submenu.



3. A new tab, "axe DevTools" will appear. Click the "Scan ALL of my page" tile.


W3C Web Accessibility initiative
Inaccessible Home Page
Before and After Demonstration
Improving a Web site using Web Content Accessibility Guidelines (WCAG) 2.0
Overview
Home
News
Tickets
Survey
Template
Inaccessible: Home Page Report
Accessible: Home Page Report
Show Annotations
CITYLIGHTS your access to the city
Traffic: Construction work on Main Road
Today: Tuesday 15 Feb
HOME
NEWS
TICKETS
SURVEY
Welcome to CityLights
Heat wave linked to temperatures
Man Gets Nine Months in Violin Case
Lack of brains hinders research
Elements
Console
Recorder
axe DevTools
DevTools
Sign up / Sign in
Scan ALL of my page
Scan PART of my page
OR
Keyboard
Modal Dialog
Interactive Elements

Citylights is the new portal for visitors and residents. Find out what's on, book tickets, and get the latest news.

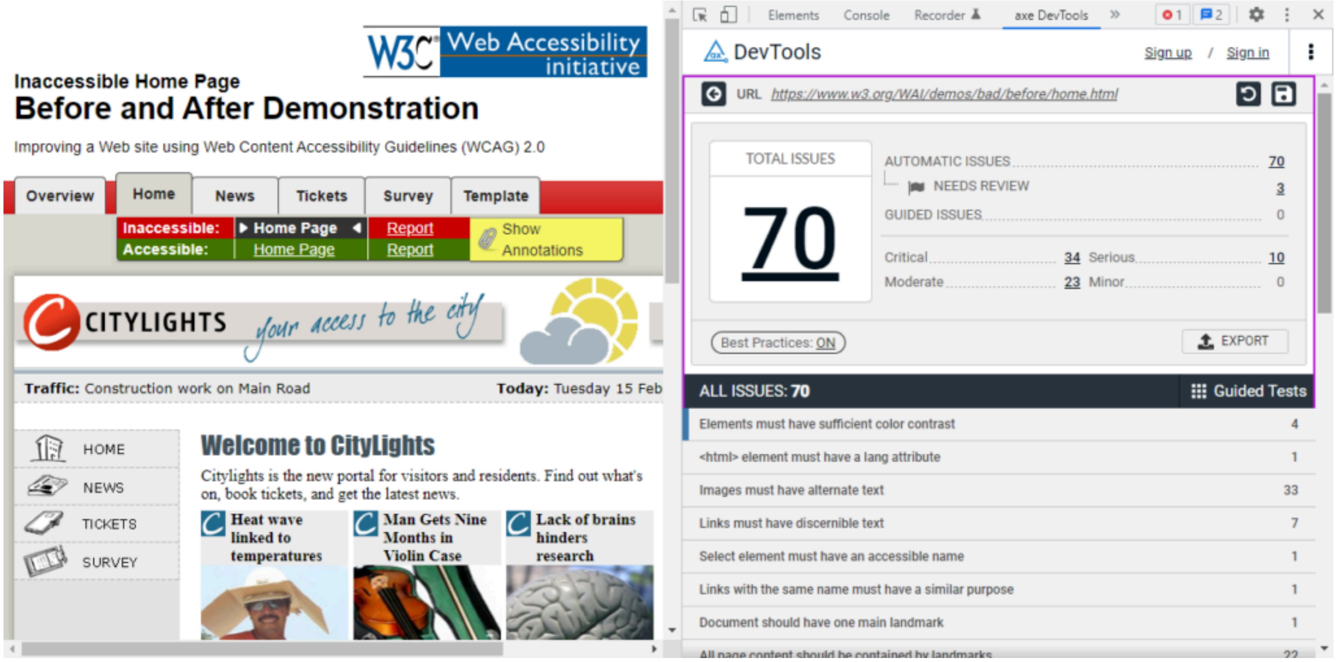
Let's get started

You're one click away from scanning your site with the most widely used and trusted accessibility tool in the world.

Start an Intelligent Guided Test

Select an IGT, we'll scan your entire page, and then you can start testing!

4. The axe test results will appear. In this example, there are 70 defects, 34 of which are Critical.


W3C Web Accessibility initiative
Inaccessible Home Page
Before and After Demonstration
Improving a Web site using Web Content Accessibility Guidelines (WCAG) 2.0
Overview
Home
News
Tickets
Survey
Template
Inaccessible: Home Page Report
Accessible: Home Page Report
Show Annotations
CITYLIGHTS your access to the city
Traffic: Construction work on Main Road
Today: Tuesday 15 Feb
HOME
NEWS
TICKETS
SURVEY
Welcome to CityLights
Heat wave linked to temperatures
Man Gets Nine Months in Violin Case
Lack of brains hinders research
Elements
Console
Recorder
axe DevTools
DevTools
Sign up / Sign in
Best Practices: ON
EXPORT
Guided Tests

Citylights is the new portal for visitors and residents. Find out what's on, book tickets, and get the latest news.

URL https://www.w3.org/WAI/demos/bad/before/home.html

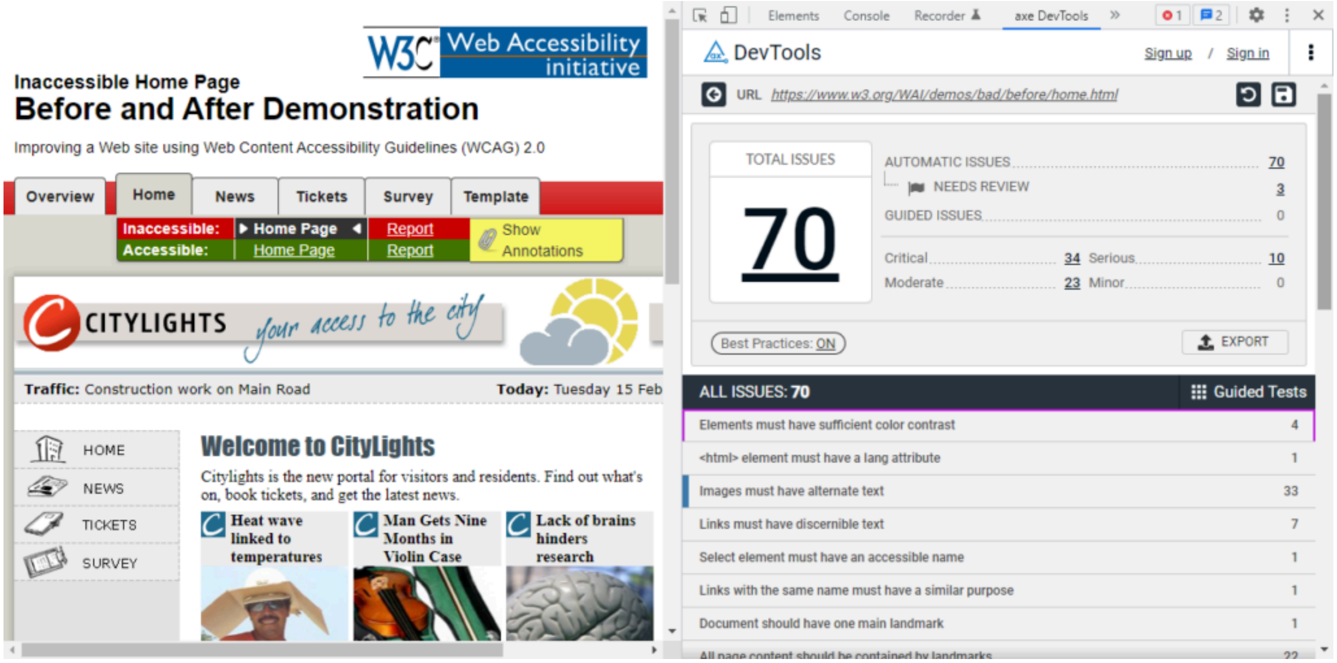
TOTAL ISSUES 70

AUTOMATIC ISSUES 70
NEEDS REVIEW 3
GUIDED ISSUES 0

Critical 34 Serious 10
Moderate 23 Minor 0

ALL ISSUES: 70	
Elements must have sufficient color contrast	4
<html> element must have a lang attribute	1
Images must have alternate text	33
Links must have discernible text	7
Select element must have an accessible name	1
Links with the same name must have a similar purpose	1
Document should have one main landmark	1
All page content should be contained by landmarks	22

5. Next, we can get additional information on a particular issue. We'll select "Images must have alternate text".


W3C Web Accessibility initiative
Overview
Home
News
Tickets
Survey
Template
Inaccessible: Home Page Report
Accessible: Home Page Report
Show Annotations
CITYLIGHTS your access to the city
HOME
NEWS
TICKETS
SURVEY
Heat wave linked to temperatures
Man Gets Nine Months in Violin Case
Lack of brains hinders research
Elements
Console
Recorder
axe DevTools
DevTools
Sign up / Sign in
Best Practices: ON
EXPORT
Guided Tests

Inaccessible Home Page

Before and After Demonstration

Improving a Web site using Web Content Accessibility Guidelines (WCAG) 2.0

Traffic: Construction work on Main Road

Today: Tuesday 15 Feb

Welcome to CityLights

Citylights is the new portal for visitors and residents. Find out what's on, book tickets, and get the latest news.

URL https://www.w3.org/WAI/demos/bad/before/home.html

TOTAL ISSUES 70

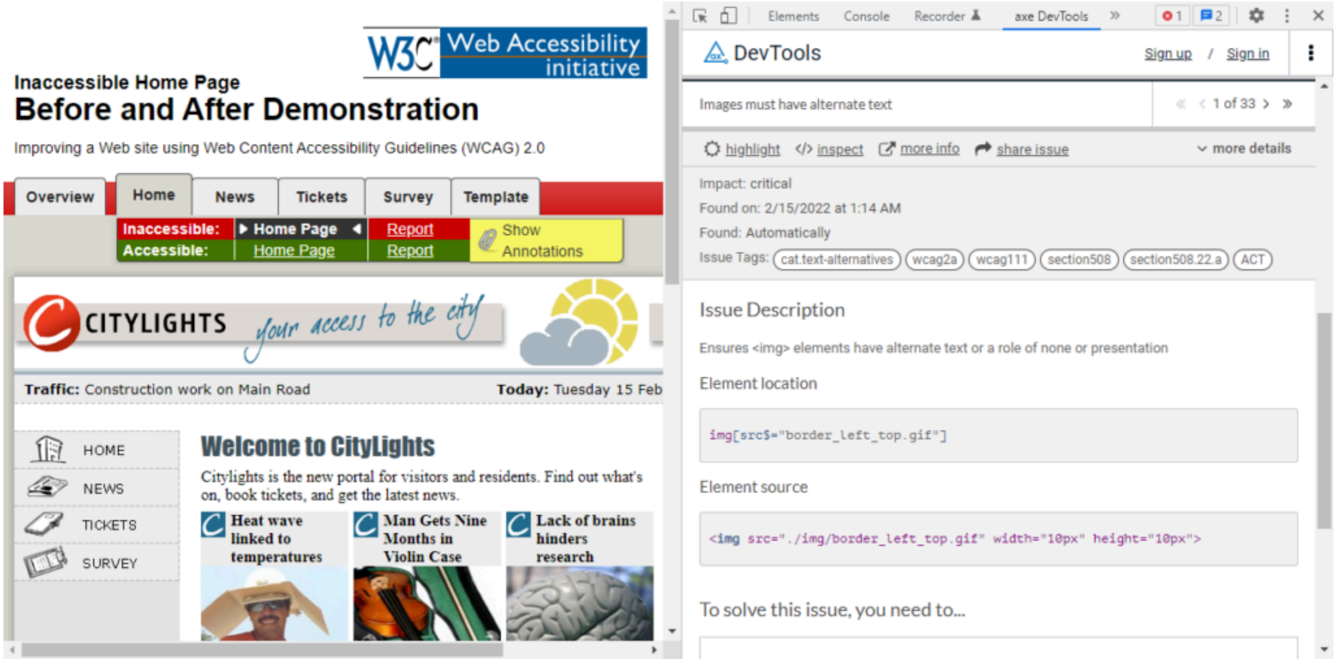
AUTOMATIC ISSUES	70
NEEDS REVIEW	3
GUIDED ISSUES	0

Critical	34	Serious	10
Moderate	23	Minor	0

ALL ISSUES: 70

Elements must have sufficient color contrast	4
<html> element must have a lang attribute	1
Images must have alternate text	33
Links must have discernible text	7
Select element must have an accessible name	1
Links with the same name must have a similar purpose	1
Document should have one main landmark	1
All page content should be contained by landmarks	22

6. A specific issue is displayed, along with Issue Tags. The tag "wcag2a" indicates that this defect relates to a double-A level **WCAG** success criterion. The tag "wcag111" indicates that this defect failed to conform to success criterion 1.1.1. Under the heading "Element source" is the HTML code which caused the defect.


W3C Web Accessibility initiative
Overview
Home
News
Tickets
Survey
Template
Inaccessible: Home Page Report
Accessible: Home Page Report
Show Annotations
CITYLIGHTS your access to the city
HOME
NEWS
TICKETS
SURVEY
Heat wave linked to temperatures
Man Gets Nine Months in Violin Case
Lack of brains hinders research
Elements
Console
Recorder
axe DevTools
DevTools
Sign up / Sign in
1 of 33
highlight
inspect
more info
share issue
more details

Inaccessible Home Page

Before and After Demonstration

Improving a Web site using Web Content Accessibility Guidelines (WCAG) 2.0

Traffic: Construction work on Main Road

Today: Tuesday 15 Feb

Welcome to CityLights

Citylights is the new portal for visitors and residents. Find out what's on, book tickets, and get the latest news.

Images must have alternate text

Impact: critical
Found on: 2/15/2022 at 1:14 AM
Found: Automatically
Issue Tags: cat.text-alternatives wcag2a wcag111 section508 section508.22.a ACT

Issue Description

Ensures <img> elements have alternate text or a role of none or presentation

Element location

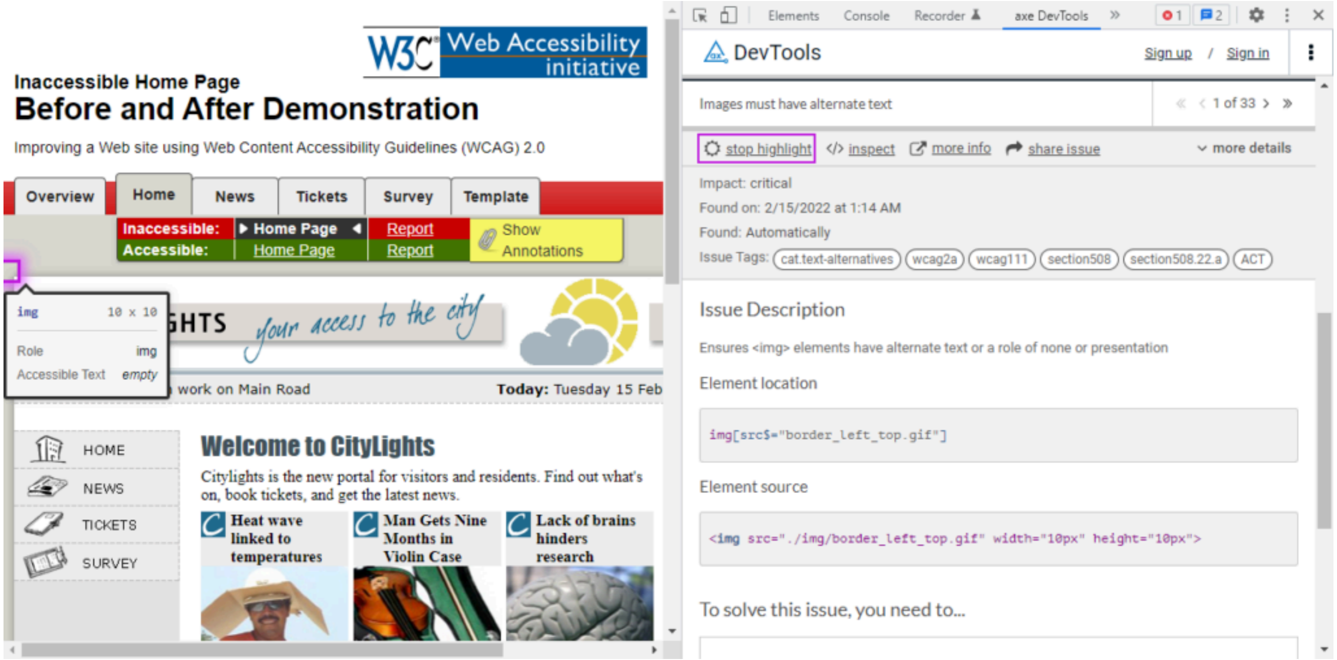
```
img[src$="border_left_top.gif"]
```

Element source

```
<img src="./img/border_left_top.gif" width="10px" height="10px">
```

To solve this issue, you need to...

7. If you click the "highlight" button, the browser will visually highlight the page element which caused the defect.


W3C Web Accessibility initiative
Inaccessible Home Page
Before and After Demonstration
Improving a Web site using Web Content Accessibility Guidelines (WCAG) 2.0
Overview
Home
News
Tickets
Survey
Template
Inaccessible: ▸ Home Page ◂ Report
Accessible: Home Page Report
Show Annotations
img 10 x 10
Role img
Accessible Text empty
your access to the city
work on Main Road
Today: Tuesday 15 Feb
HOME
NEWS
TICKETS
SURVEY
Welcome to CityLights
Citylights is the new portal for visitors and residents. Find out what's on, book tickets, and get the latest news.
Heat wave linked to temperatures
Man Gets Nine Months in Violin Case
Lack of brains hinders research
Elements
Console
Recorder
axe DevTools
DevTools
Sign up / Sign in
1 of 33
stop highlight
inspect
more info
share issue
more details

Images must have alternate text

Impact: critical

Found on: 2/15/2022 at 1:14 AM

Found: Automatically

Issue Tags: cat.text-alternatives wcag2a wcag111 section508 section508.22.a ACT

Issue Description

Ensures <img> elements have alternate text or a role of none or presentation

Element location

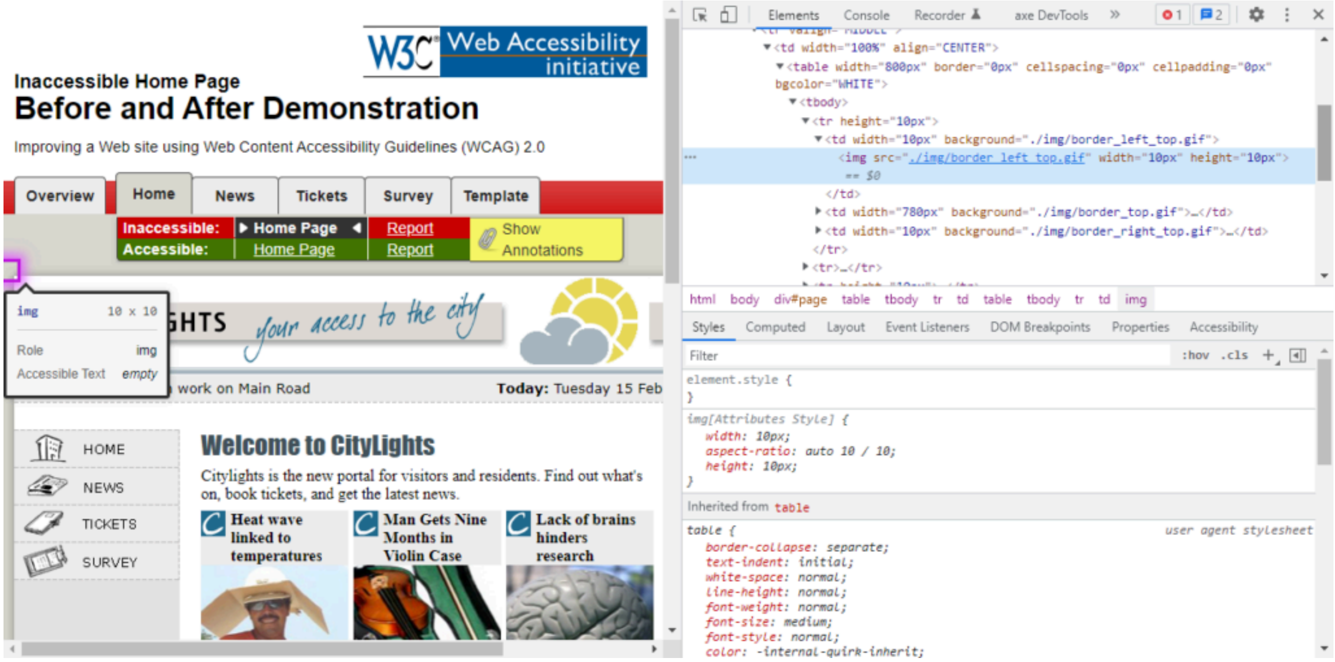
```
img[src$="border_left_top.gif"]
```

Element source

```
<img src="./img/border_left_top.gif" width="10px" height="10px">
```

To solve this issue, you need to...

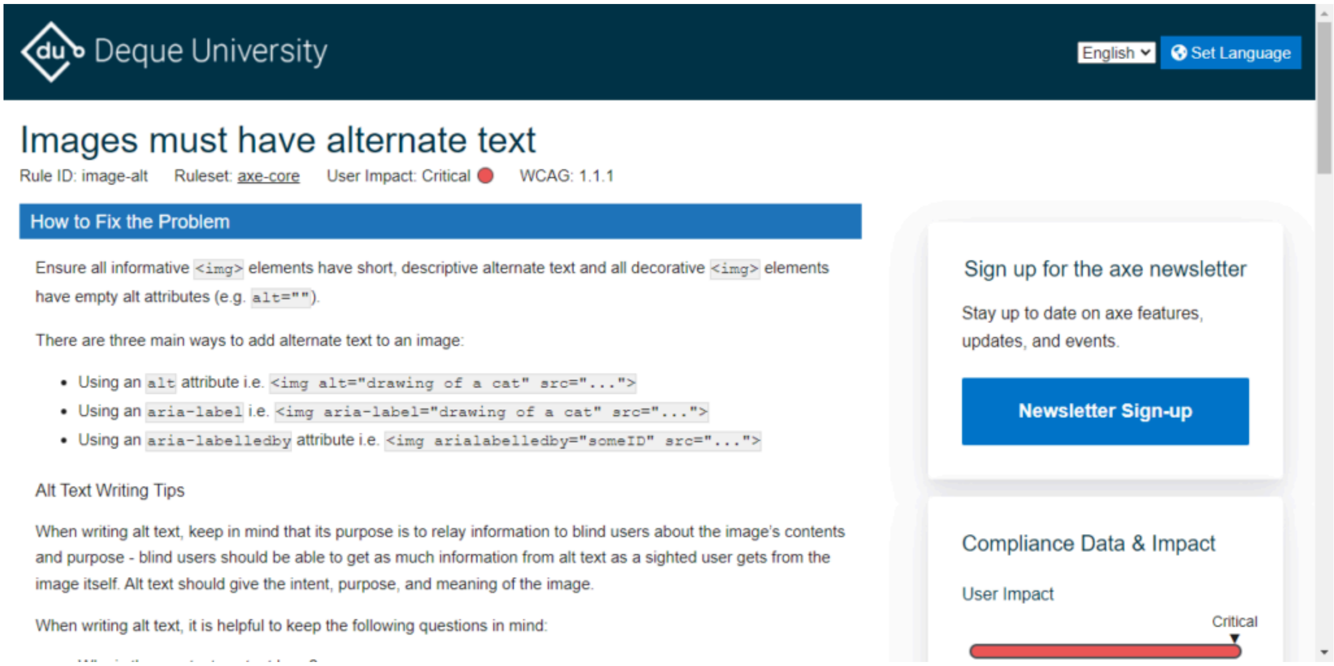
8. If you click the "Inspect" button, the Elements tab will display, and the HTML code that caused the defect will be highlighted. There's an <img> tag without an alt attribute in this example.


W3C Web Accessibility initiative
Inaccessible Home Page
Before and After Demonstration
Improving a Web site using Web Content Accessibility Guidelines (WCAG) 2.0
Overview
Home
News
Tickets
Survey
Template
Inaccessible: ▸ Home Page ◂ Report
Accessible: Home Page Report
Show Annotations
img 10 x 10
Role img
Accessible Text empty
your access to the city
work on Main Road
Today: Tuesday 15 Feb
HOME
NEWS
TICKETS
SURVEY
Welcome to CityLights
Citylights is the new portal for visitors and residents. Find out what's on, book tickets, and get the latest news.
Heat wave linked to temperatures
Man Gets Nine Months in Violin Case
Lack of brains hinders research
Elements
Console
Recorder
axe DevTools
html body div#page table tbody tr td table tbody tr td img
Styles
Computed
Layout
Event Listeners
DOM Breakpoints
Properties
Accessibility
Filter
:hov .cls

```
▼<td width="100%" align="CENTER">
  ▼<table width="800px" border="0px" cellspacing="0px" cellpadding="0px"
  bgcolor="WHITE">
    ▼<tbody>
      ▼<tr height="10px">
        ▼<td width="10px" background="./img/border_left_top.gif">
            <img src="./img/border_left_top.gif" width="10px" height="10px">
             == $0
          </td>
        ▸<td width="780px" background="./img/border_top.gif">…</td>
        ▸<td width="10px" background="./img/border_right_top.gif">…</td>
        </tr>
      ▸<tr>…</tr>
```

```
element.style {
}
img[Attributes Style] {
   width: 10px;
   aspect-ratio: auto 10 / 10;
   height: 10px;
}
Inherited from table
table {                                          user agent stylesheet
   border-collapse: separate;
   text-indent: initial;
   white-space: normal;
   line-height: normal;
   font-weight: normal;
   font-size: medium;
   font-style: normal;
   color: -internal-quirk-inherit;
```

9. If you go back to the "axe DevTools" tab and click the "more info" button, a new browser tab will open and display details about this type of defect. Close this tab when you are done with it.



Images must have alternate text

Rule ID: image-alt Ruleset: axe-core User Impact: Critical ● WCAG: 1.1.1

How to Fix the Problem

Ensure all informative `<img>` elements have short, descriptive alternate text and all decorative `<img>` elements have empty alt attributes (e.g. `alt=""`).

There are three main ways to add alternate text to an image:

- Using an `alt` attribute i.e. `<img alt="drawing of a cat" src="...">`
- Using an `aria-label` i.e. `<img aria-label="drawing of a cat" src="...">`
- Using an `aria-labelledby` attribute i.e. `<img arialabelledby="someID" src="...">`

Alt Text Writing Tips

When writing alt text, keep in mind that its purpose is to relay information to blind users about the image's contents and purpose - blind users should be able to get as much information from alt text as a sighted user gets from the image itself. Alt text should give the intent, purpose, and meaning of the image.

When writing alt text, it is helpful to keep the following questions in mind:

Sign up for the axe newsletter

Stay up to date on axe features, updates, and events.

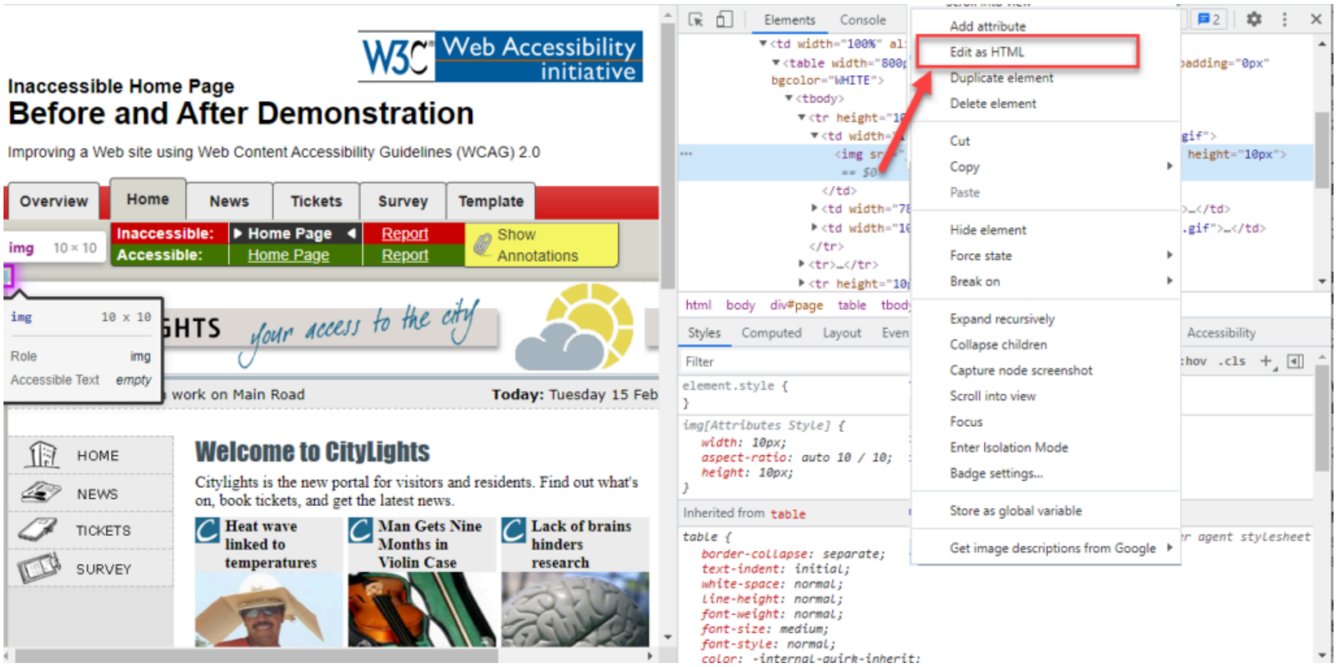
Newsletter Sign-up

Compliance Data & Impact

User Impact

Critical

10. When you write up your test results, you may want to recommend a code fix. You can test your code fix right in the browser – go to the Elements tab, right-click on the line of code you'd like to change, and then select "Edit as HTML".



Inaccessible Home Page

Before and After Demonstration

Improving a Web site using Web Content Accessibility Guidelines (WCAG) 2.0

Welcome to CityLights

Citylights is the new portal for visitors and residents. Find out what's on, book tickets, and get the latest news.

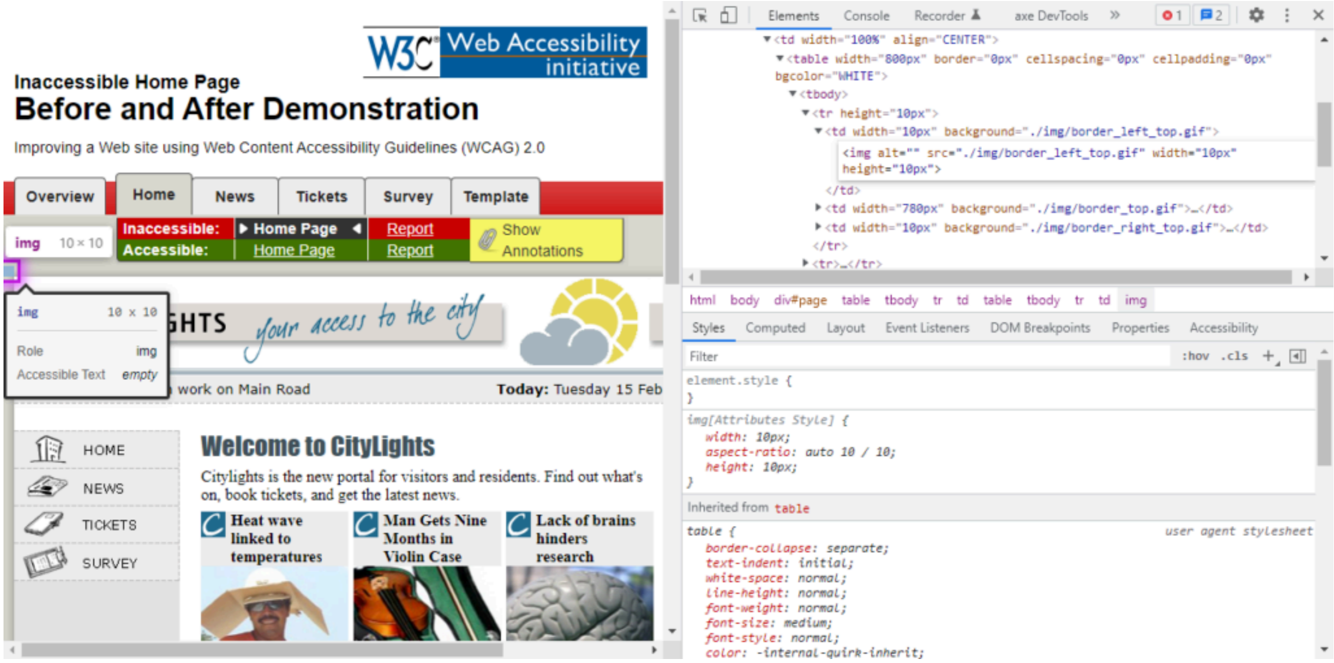
Heat wave linked to temperatures

Man Gets Nine Months in Violin Case

Lack of brains hinders research

```
element.style {
}
img[Attributes Style] {
    width: 10px;
    aspect-ratio: auto 10 / 10;
    height: 10px;
}
Inherited from table
table {
    border-collapse: separate;
    text-indent: initial;
    white-space: normal;
    line-height: normal;
    font-weight: normal;
    font-size: medium;
    font-style: normal;
    color: -internal-quirk-inherit;
```

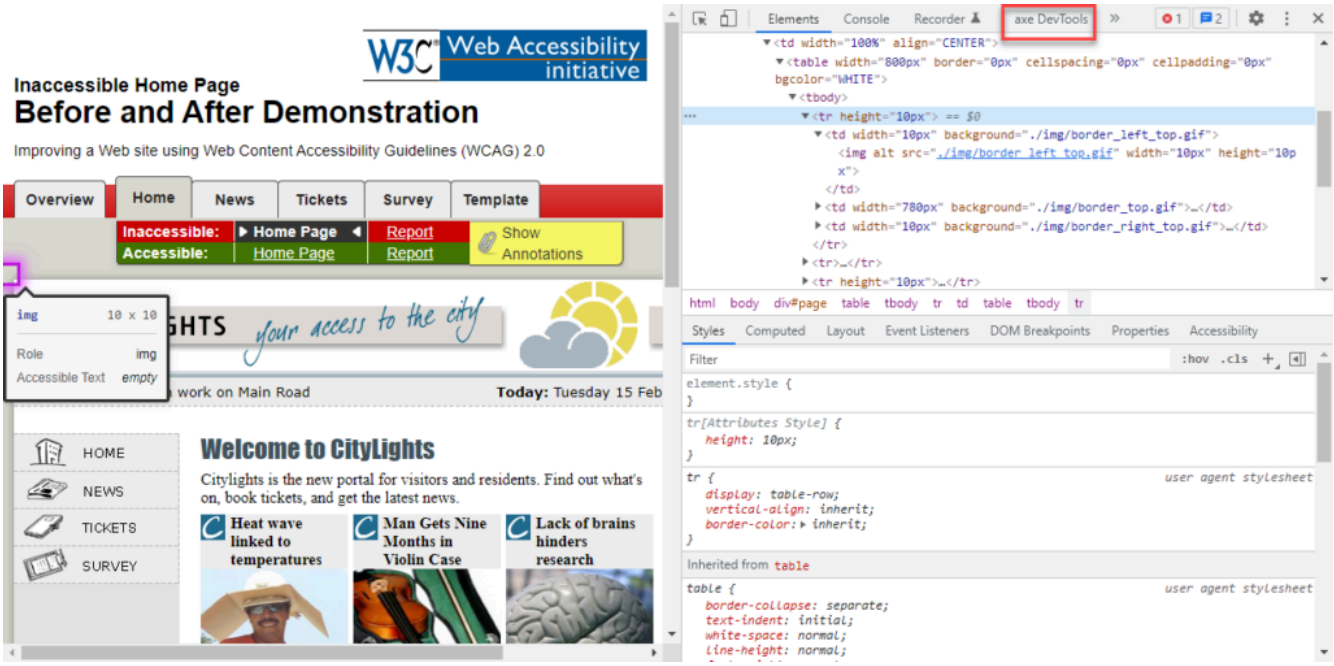
11. For this specific issue that is related to an image not having alternative text, if you add alt="", the defect will be fixed! (In this case the image is part of a border, so it's just decorative, but it still needs an alt attribute.)


W3C Web Accessibility initiative
Inaccessible Home Page
Before and After Demonstration
Improving a Web site using Web Content Accessibility Guidelines (WCAG) 2.0
Overview
Home
News
Tickets
Survey
Template
Inaccessible: Home Page Report
Accessible: Home Page Report
Show Annotations
img 10 × 10
img 10 x 10
Role img
Accessible Text empty
your access to the city
work on Main Road
Today: Tuesday 15 Feb
HOME
NEWS
TICKETS
SURVEY
Welcome to CityLights
Citylights is the new portal for visitors and residents. Find out what's on, book tickets, and get the latest news.
Heat wave linked to temperatures
Man Gets Nine Months in Violin Case
Lack of brains hinders research
Elements
Console
Recorder
axe DevTools
1
2
html body div#page table tbody tr td table tbody tr td img
Styles
Computed
Layout
Event Listeners
DOM Breakpoints
Properties
Accessibility
Filter
:hov .cls +

```
▼<td width="100%" align="CENTER">
  ▼<table width="800px" border="0px" cellspacing="0px" cellpadding="0px"
  bgcolor="WHITE">
    ▼<tbody>
      ▼<tr height="10px">
        ▼<td width="10px" background="./img/border_left_top.gif">
            <img alt="" src="./img/border_left_top.gif" width="10px"
            height="10px">
          </td>
        ▶<td width="780px" background="./img/border_top.gif">…</td>
        ▶<td width="10px" background="./img/border_right_top.gif">…</td>
       </tr>
      ▶<tr>…</tr>
```

```
element.style {
}
img[Attributes Style] {
   width: 10px;
   aspect-ratio: auto 10 / 10;
   height: 10px;
}
Inherited from table
table {                                          user agent stylesheet
   border-collapse: separate;
   text-indent: initial;
   white-space: normal;
   line-height: normal;
   font-weight: normal;
   font-size: medium;
   font-style: normal;
   color: -internal-quirk-inherit;
```

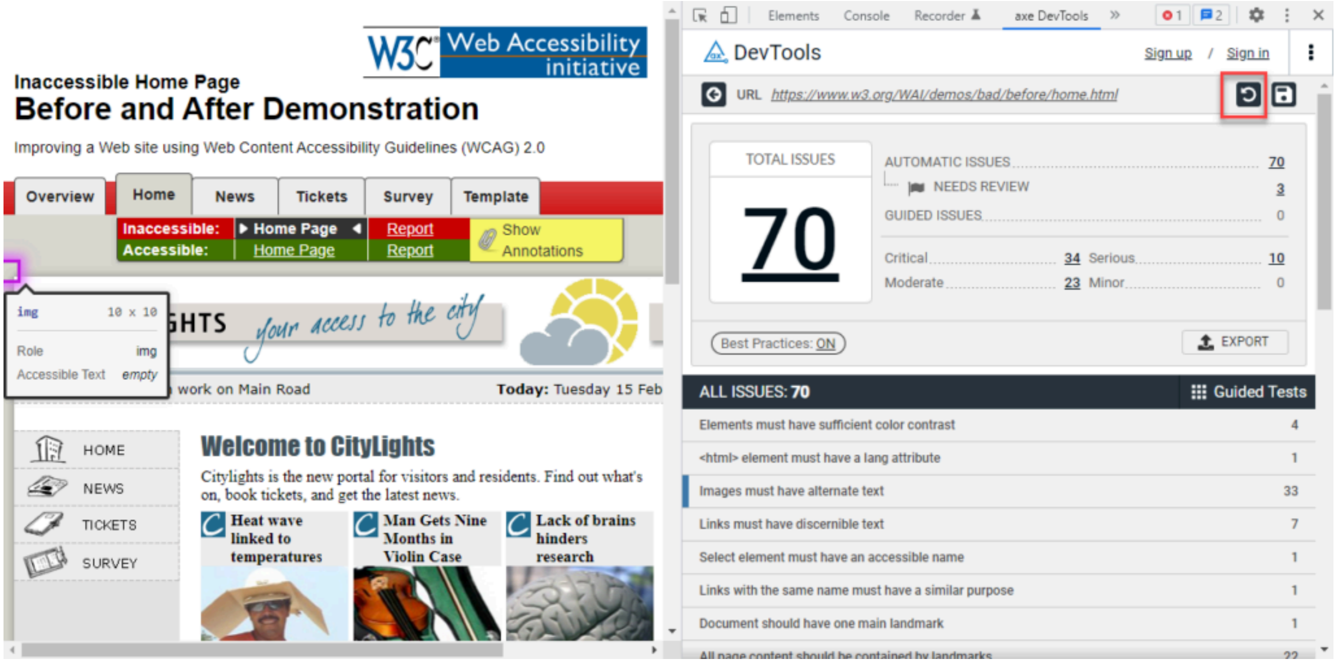
12. Now click on the “axe DevTools” tab to go back to axe.


W3C Web Accessibility initiative
Inaccessible Home Page
Before and After Demonstration
Improving a Web site using Web Content Accessibility Guidelines (WCAG) 2.0
Overview
Home
News
Tickets
Survey
Template
Inaccessible: Home Page Report
Accessible: Home Page Report
Show Annotations
img 10 x 10
Role img
Accessible Text empty
your access to the city
work on Main Road
Today: Tuesday 15 Feb
HOME
NEWS
TICKETS
SURVEY
Welcome to CityLights
Citylights is the new portal for visitors and residents. Find out what's on, book tickets, and get the latest news.
Heat wave linked to temperatures
Man Gets Nine Months in Violin Case
Lack of brains hinders research
Elements
Console
Recorder
axe DevTools
1
2
html body div#page table tbody tr td table tbody tr
Styles
Computed
Layout
Event Listeners
DOM Breakpoints
Properties
Accessibility
Filter
:hov .cls +

```
▼<td width="100%" align="CENTER">
  ▼<table width="800px" border="0px" cellspacing="0px" cellpadding="0px"
  bgcolor="WHITE">
    ▼<tbody>
...   ▼<tr height="10px"> == $0
        ▼<td width="10px" background="./img/border_left_top.gif">
            <img alt src="./img/border_left_top.gif" width="10px" height="10p
            x">
          </td>
        ▶<td width="780px" background="./img/border_top.gif">…</td>
        ▶<td width="10px" background="./img/border_right_top.gif">…</td>
       </tr>
      ▶<tr>…</tr>
      ▶<tr height="10px">…</tr>
```

```
element.style {
}
tr[Attributes Style] {
   height: 10px;
}
tr {                                             user agent stylesheet
   display: table-row;
   vertical-align: inherit;
   border-color: ▶ inherit;
}
Inherited from table
table {                                          user agent stylesheet
   border-collapse: separate;
   text-indent: initial;
   white-space: normal;
   line-height: normal;
```

13. Click the button that looks like an arrow pointing in a circle (if you are using a screen reader, it is the “Re-run automatic scan” button). This will re-run your scan.



Elements
Console
Recorder
axe DevTools
DevTools
Sign up / Sign in
W3C Web Accessibility initiative
Inaccessible Home Page
Before and After Demonstration
Improving a Web site using Web Content Accessibility Guidelines (WCAG) 2.0
Overview
Home
News
Tickets
Survey
Template
Inaccessible: Home Page Report
Accessible: Home Page Report
Show Annotations
img 10 x 10
Role img
Accessible Text empty
your access to the city
work on Main Road
Today: Tuesday 15 Feb
HOME
NEWS
TICKETS
SURVEY
Welcome to CityLights
Citylights is the new portal for visitors and residents. Find out what's on, book tickets, and get the latest news.
Heat wave linked to temperatures
Man Gets Nine Months in Violin Case
Lack of brains hinders research
Best Practices: ON
EXPORT
Guided Tests

URL https://www.w3.org/WAI/demos/bad/before/home.html

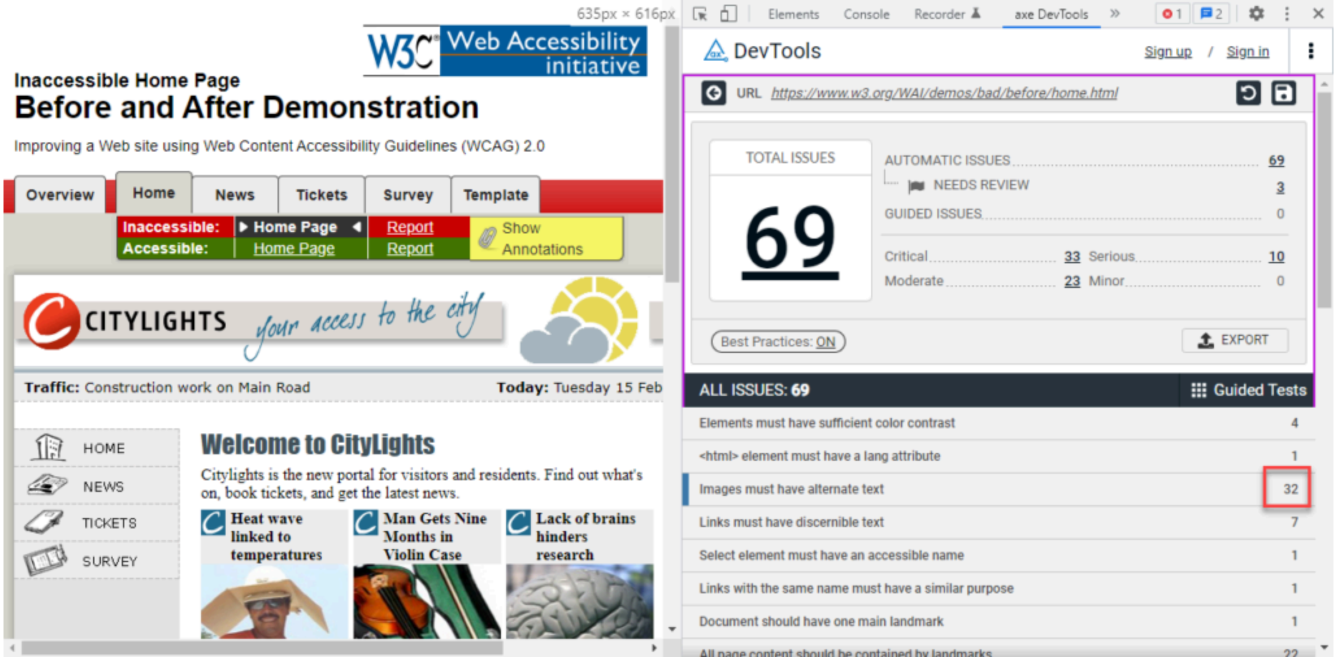
TOTAL ISSUES 70

AUTOMATIC ISSUES 70
NEEDS REVIEW 3
GUIDED ISSUES 0
Critical 34 Serious 10
Moderate 23 Minor 0

ALL ISSUES: 70

Elements must have sufficient color contrast	4
<html> element must have a lang attribute	1
Images must have alternate text	33
Links must have discernible text	7
Select element must have an accessible name	1
Links with the same name must have a similar purpose	1
Document should have one main landmark	1
All page content should be contained by landmarks	22

14. Now the defect is fixed, and the number of defects has decreased by one. Remember to specify your code fix in the audit report, now that you've verified it works!



635px × 616px
Elements
Console
Recorder
axe DevTools
DevTools
Sign up / Sign in
W3C Web Accessibility initiative
Inaccessible Home Page
Before and After Demonstration
Improving a Web site using Web Content Accessibility Guidelines (WCAG) 2.0
Overview
Home
News
Tickets
Survey
Template
Inaccessible: Home Page Report
Accessible: Home Page Report
Show Annotations
CITYLIGHTS
your access to the city
Traffic: Construction work on Main Road
Today: Tuesday 15 Feb
HOME
NEWS
TICKETS
SURVEY
Welcome to CityLights
Citylights is the new portal for visitors and residents. Find out what's on, book tickets, and get the latest news.
Heat wave linked to temperatures
Man Gets Nine Months in Violin Case
Lack of brains hinders research
Best Practices: ON
EXPORT
Guided Tests

URL https://www.w3.org/WAI/demos/bad/before/home.html

TOTAL ISSUES 69

AUTOMATIC ISSUES 69
NEEDS REVIEW 3
GUIDED ISSUES 0
Critical 33 Serious 10
Moderate 23 Minor 0

ALL ISSUES: 69

Elements must have sufficient color contrast	4
<html> element must have a lang attribute	1
Images must have alternate text	32
Links must have discernible text	7
Select element must have an accessible name	1
Links with the same name must have a similar purpose	1
Document should have one main landmark	1
All page content should be contained by landmarks	22

Testing tips

Here are some additional testing tips you may find helpful:

- axe usually reports accurate results, although it may occasionally report a false positive. For example: if you have white text on a coloured background, axe sometimes assumes that the background is white and therefore thinks you're using white text on a white background. Then it will incorrectly flag this as a colour contrast defect.
- Content must be displayed to be scanned by axe. To thoroughly test all parts of the page, you may need to open accordions, open modal dialogs, and try all responsive design layouts (by changing your browser window's width).

Resources:

- axe browser extensions (for Chrome, Firefox, and Edge)
- Pages to test axe on – both Before and After they've been fixed

SCREEN READER TESTING

Screen readers allow people who have difficulty seeing or reading to access information. The screen reader will identify text in a website, app, or document and read it out loud with an electronic voice to the user. Screen readers are Artificial Intelligence (AI) and so they do not intuitively understand what order text should be read in and rely on features like headers and other clues built into documents, websites, and applications for context. Using a screen reader on your content is one way to test its accessibility.

Introduction to NVDA

NonVisual Desktop Access (**NVDA**) is a free screen reader for Windows. NVDA is popular for testing web pages and documents because:

- It's available for free (pay what you can) and is well-maintained.
- It's lightweight (i.e., it has a relatively small install file) and loads quickly.
- It adheres to web standards, so it's a very accurate testing tool. Another Windows screen reader, **JAWS**, applies heuristic rules and guesses how to announce page elements. For example, if an input field has no label, NVDA will tell you that, but JAWS will use nearby text to guess what the label is.

Install NVDA

You can download NVDA from the NVDA Install page.

NVDA commands

Basic commands

These commands are the very first ones you'll need to get you started.

Command	Function
Ctrl	Stop talking until another key is pressed
Insert+s	Toggle speech on/off
Insert+q	Close NVDA

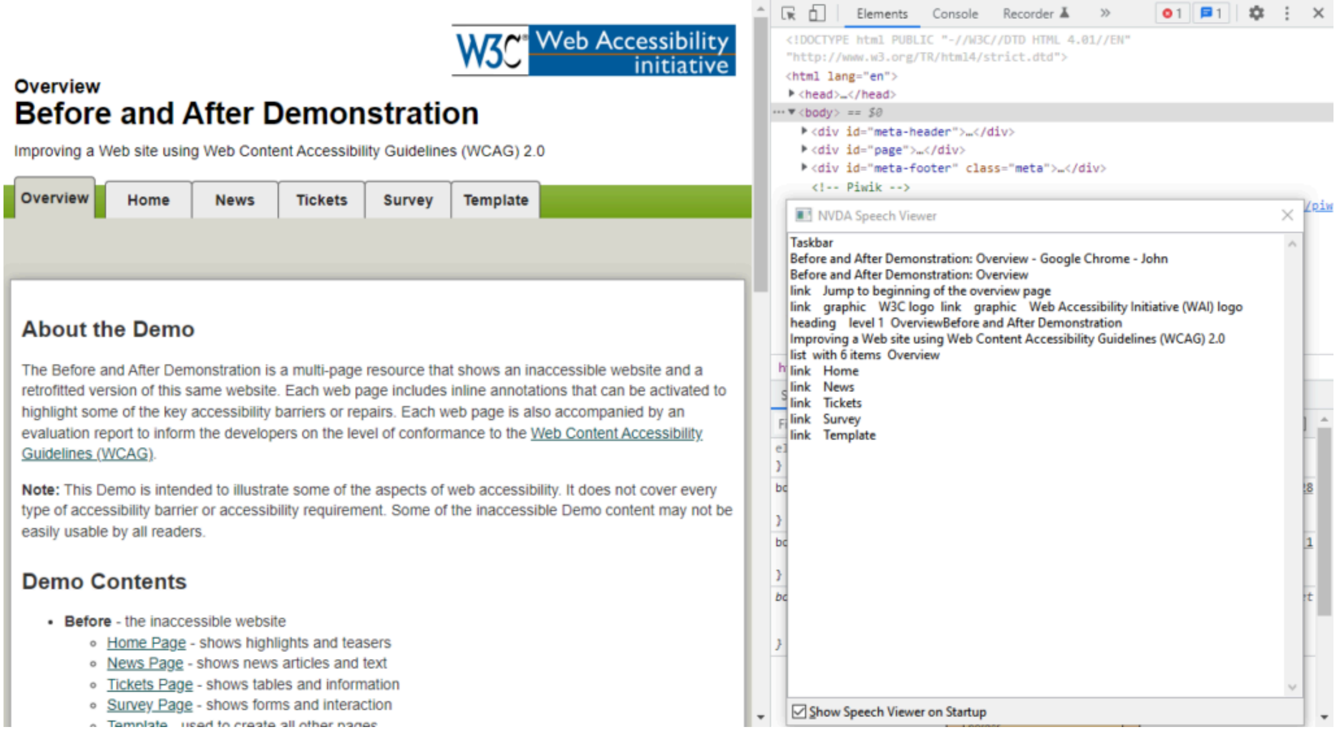
Settings

Here are some basic setup commands you'll want to set based on your preference.

Command	Function
Insert+m	Toggle mouse tracking on/off (Turn this off so moving the mouse won't announce content your mouse pointer touches.)
NVDA > Tools > Speech Viewer	Speech Viewer (Turn this on to see what NVDA is announcing. Include this in your screenshots.)
NVDA > Preferences > Settings > Speech	Select a voice Synthesizer that you like, and then set the speaking Rate (how fast text will be read).

Speech Viewer

The Speech Viewer is a floating window that displays everything that NVDA announces. You can position the window wherever you like to include screenshots with logged defects. To open the viewer, click the NVDA logo in the Windows taskbar, select Tools, then Speech Viewer.



Overview
Before and After Demonstration

Improving a Web site using Web Content Accessibility Guidelines (WCAG) 2.0

About the Demo

The Before and After Demonstration is a multi-page resource that shows an inaccessible website and a retrofitted version of this same website. Each web page includes inline annotations that can be activated to highlight some of the key accessibility barriers or repairs. Each web page is also accompanied by an evaluation report to inform the developers on the level of conformance to the Web Content Accessibility Guidelines (WCAG).

Note: This Demo is intended to illustrate some of the aspects of web accessibility. It does not cover every type of accessibility barrier or accessibility requirement. Some of the inaccessible Demo content may not be easily usable by all readers.

Demo Contents

- **Before** - the inaccessible website
  - Home Page - shows highlights and teasers
  - News Page - shows news articles and text
  - Tickets Page - shows tables and information
  - Survey Page - shows forms and interaction
  - Template - used to create all other pages

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
 ▸<head>…</head>
…▾<body> == $0
   ▸<div id="meta-header">…</div>
   ▸<div id="page">…</div>
   ▸<div id="meta-footer" class="meta">…</div>
    <!-- Piwik -->
```

```
Taskbar
Before and After Demonstration: Overview - Google Chrome - John
Before and After Demonstration: Overview
link   Jump to beginning of the overview page
link   graphic   W3C logo  link   graphic   Web Accessibility Initiative (WAI) logo
heading   level 1  OverviewBefore and After Demonstration
Improving a Web site using Web Content Accessibility Guidelines (WCAG) 2.0
list  with 6 items  Overview
link   Home
link   News
link   Tickets
link   Survey
link   Template
```

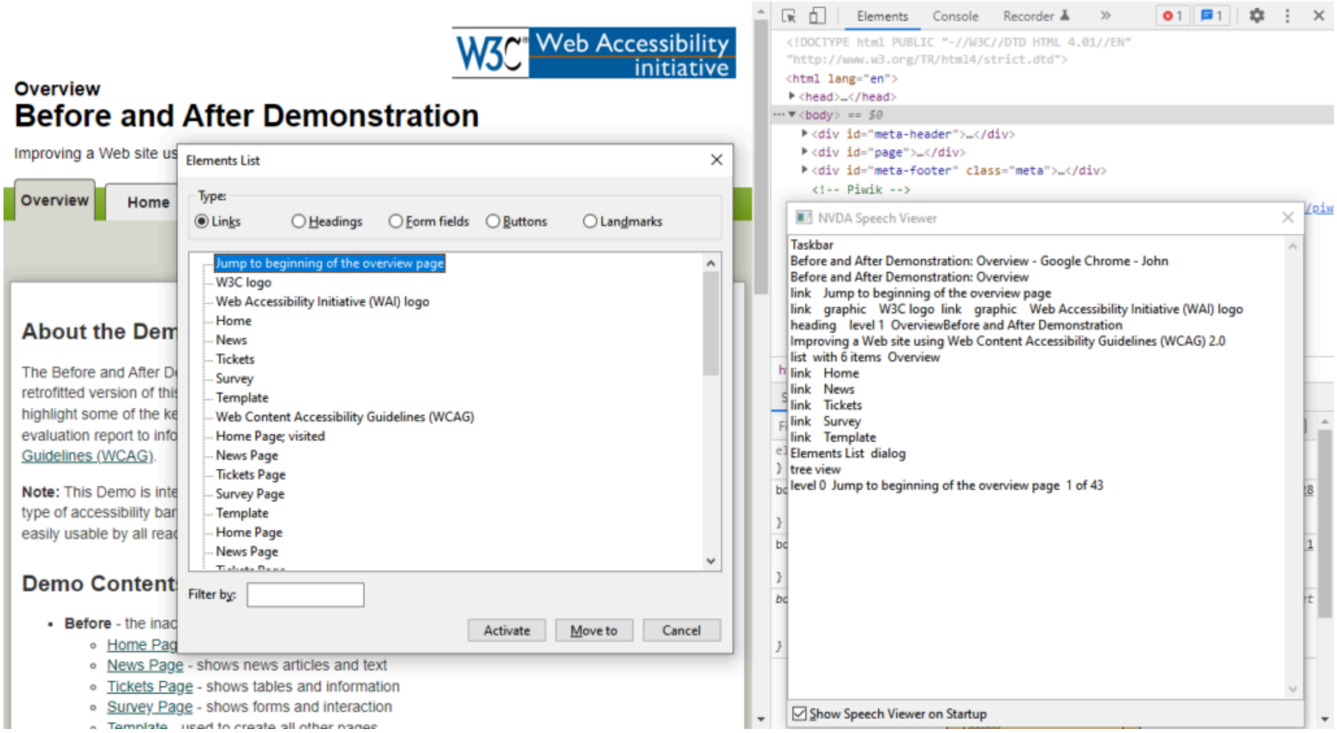
Main navigation commands

Here are the main commands you'll use to move around a web page.

Command	Function
Up arrow key	Go up a line
Down arrow key	Go down a line
Tab	Go to next clickable element
Shift+Tab	Go to previous clickable element
Insert+F7	Open Elements List – to list Links, Headings, Form fields, Buttons, and Landmarks
Ctrl+Home	Go to the top of the page

Elements List

The Elements list feature displays all Links, Headings, Form fields, Buttons, and Landmarks on the current page. It gives a good overview of the page at a glance.


W3C Web Accessibility initiative
Overview
Before and After Demonstration
Overview
Home
Elements List
Type:
● Links ○ Headings ○ Form fields ○ Buttons ○ Landmarks
Filter by:
Activate
Move to
Cancel
Elements
Console
Recorder
NVDA Speech Viewer
☑ Show Speech Viewer on Startup

- Jump to beginning of the overview page
- W3C logo
- Web Accessibility Initiative (WAI) logo
- Home
- News
- Tickets
- Survey
- Template
- Web Content Accessibility Guidelines (WCAG)
- Home Page; visited
- News Page
- Tickets Page
- Survey Page
- Template
- Home Page
- News Page

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
 ▶<head>…</head>
▼<body> == $0
   ▶<div id="meta-header">…</div>
   ▶<div id="page">…</div>
   ▶<div id="meta-footer" class="meta">…</div>
    <!-- Piwik -->
```

```
Taskbar
Before and After Demonstration: Overview - Google Chrome - John
Before and After Demonstration: Overview
link   Jump to beginning of the overview page
link   graphic   W3C logo  link   graphic   Web Accessibility Initiative (WAI) logo
heading   level 1  OverviewBefore and After Demonstration
Improving a Web site using Web Content Accessibility Guidelines (WCAG) 2.0
list  with 6 items  Overview
link   Home
link   News
link   Tickets
link   Survey
link   Template
Elements List  dialog
tree view
level 0  Jump to beginning of the overview page  1 of 43
```

About the Dem

The Before and After D retrofitted version of this highlight some of the ke evaluation report to info Guidelines (WCAG).

Note: This Demo is inte type of accessibility bar easily usable by all read

Demo Content

- **Before** - the inac
  - Home Pag
  - News Page - shows news articles and text
  - Tickets Page - shows tables and information
  - Survey Page - shows forms and interaction

Other navigation commands

Here are some commands that you may use less often, but are still important.

Command	Function
Insert+space or Escape	Exit forms mode (when in an input field) – so you can press the down arrow key and continue reading
h	Go to next heading
Shift+h	Go to previous heading
Right arrow key	Announce next character
Left arrow key	Announce previous character
d	Go to next landmark
Shift+d	Go to previous landmark

Data table commands

When testing a data table, NVDA has commands to move up, down, left, and right.

Command	Function
Ctrl+Alt+Up arrow	Move up one cell
Ctrl+Alt+Down arrow	Move down one cell
Ctrl+Alt+Left arrow	Move left one cell
Ctrl+Alt+Right arrow	Move right one cell

Browse mode vs. Focus mode

One feature of NVDA is that there are two distinct modes it operates in depending on what type of webpage layout is interpreted. In most layouts, NVDA will operate in Browse mode, unless NVDA detects that you have moved to a form (meaning the user is expected to enter in some information). Within a form, NVDA will switch to Focus mode.

Browse mode

When you press the down arrow key (and go down one line in the page), NVDA will announce all types of web page content, including text, links, and form controls. This is known as Browse mode. Using Browse mode is an excellent way to read all content without missing anything from top to bottom.

Focus mode

Once you tab to a form control, pressing the arrow keys will have different results. This is known as Focus mode. For example, you can select the radio button you want using arrow keys. Or, when you're in an input field, pressing the up/down arrow keys will have no effect. When you're ready to exit Focus mode, press Insert+space or the Escape key, and then you will be back in Browse mode.

Tip: If you get stuck in an input field, press Insert+space to exit Focus mode, and then you can use the down arrow key to continue reading.

How to test a web page with NVDA

To test a web page with NVDA, try these steps:

1. Open the web page you're going to test.
2. Run NVDA.
3. Click anywhere on your browser window, so NVDA is focused on it.
4. Tab into the web page so NVDA can read the web page contents.
5. Set the mouse aside, so it doesn't interfere with your test.
6. Press Ctrl to stop NVDA from reading. (You will use this a lot.)
7. Press Ctrl+Home to jump to the top of the page. Then, repeatedly press the down arrow key, reading the page contents from top to bottom.
8. Once you've reached the bottom of the page, press Ctrl+Home to jump to the top of the page again. This time, tab through the page – you should reach each link and form control. Test each one (or at least a good sampling of them) by pressing Enter when you reach it.
9. Without using the mouse, you should be able to have NVDA announce anything on the page, and you should be able to use all links and form controls.
10. Press Insert+F7 – this will display a convenient menu of Links, Headings, Form fields, Buttons, and Landmarks.
11. Press Insert+q to close NVDA.

Resources:

- Download NVDA
- NVDA User Guide

    - NVDA – Browse mode and Focus mode

ACCESSIBILITY INSIGHTS FOR WINDOWS

There is an additional **accessibility** checking tool available for Windows applications called Microsoft Accessibility Insights. This tool can inspect Windows applications using a standard called User Interface Automation (UIA), which is the accessibility framework for Microsoft Windows. UIA provides access to user interface (UI) elements which is what enables **assistive technology** **(AT)** products like **screen readers** to provide information about the UI to users.

Installing Accessibility Insights

To download Accessibility Insights for Windows, go to the Accessibility Insights for Windows page and click the “Download for Windows” button. Follow the prompts to install the software.

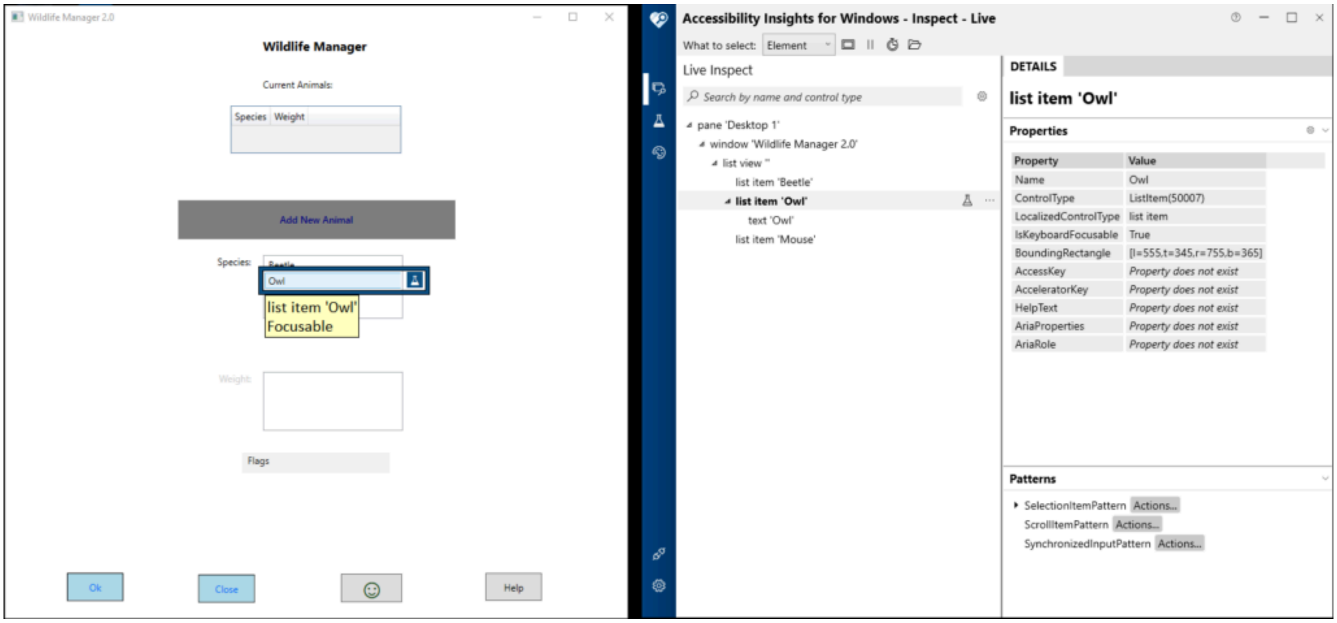
Wildlife Manager (demo app)

Accessibility Insights includes a demo app: Wildlife Manager, with which you can learn to use Accessibility Insights. To get the demo app, go to the WildlifeManager page and download WildlifeManager.exe. You can open the .exe file right away (no install is needed).

Live Inspect

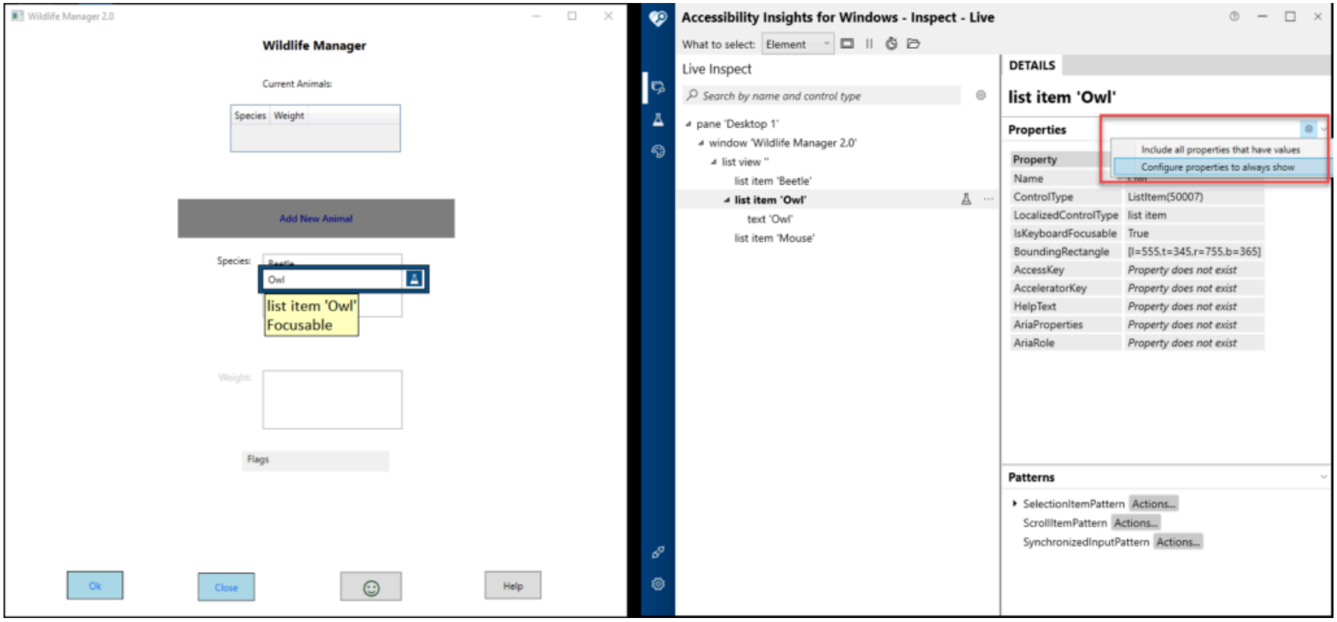
You can use the Live Inspect feature to inspect UI elements by placing your mouse over them.

To use it, open the demo app and Accessibility Insights side-by-side. Move your mouse pointer or tab to each UI element. Accessibility Insights will display the element’s name and properties.



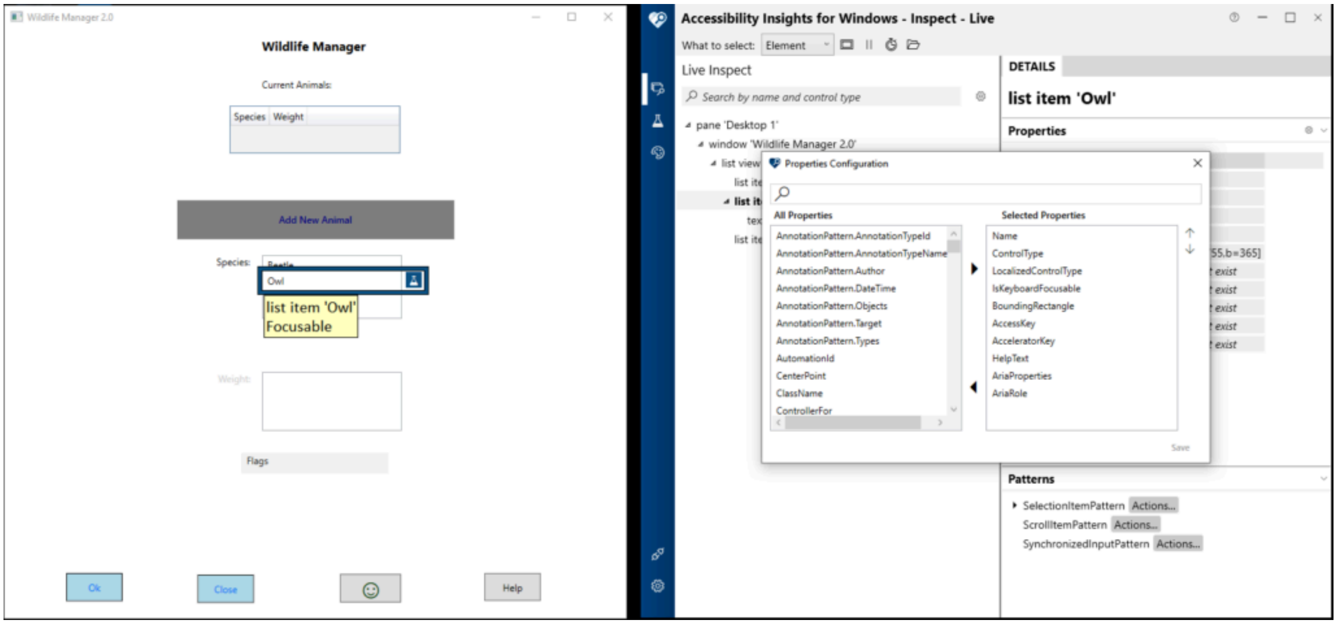
Property	Value
Name	Owl
ControlType	ListItem(50007)
LocalizedControlType	list item
IsKeyboardFocusable	True
BoundingRectangle	[l=555,t=345,r=755,b=365]
AccessKey	*Property does not exist*
AcceleratorKey	*Property does not exist*
HelpText	*Property does not exist*
AriaProperties	*Property does not exist*
AriaRole	*Property does not exist*

You can customize the list of properties that are shown by clicking the “Properties Settings” button (which looks like a gear icon).



Property	Value
Name	Owl
ControlType	ListItem(50007)
LocalizedControlType	list item
IsKeyboardFocusable	True
BoundingRectangle	[l=555,t=345,r=755,b=365]
AccessKey	*Property does not exist*
AcceleratorKey	*Property does not exist*
HelpText	*Property does not exist*
AriaProperties	*Property does not exist*
AriaRole	*Property does not exist*

Then you can manage your list of Selected Properties.



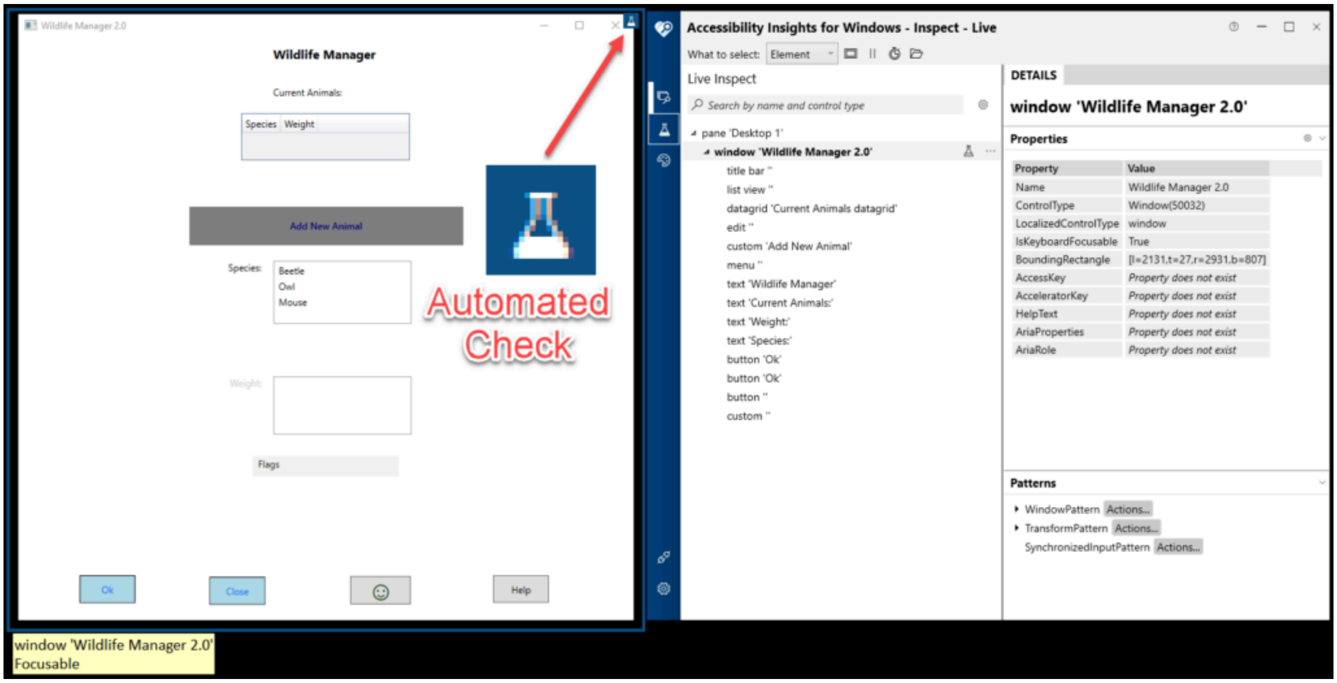
FastPass

When developers are editing code, they can run a FastPass test before each code check-in to ensure no accessibility issues have occurred as a result of their changes. FastPass consists of two main features – Automated Checks and Tab Stops.

Automated Checks

The Automated Checks feature checks an application's elements for **conformance** with dozens of accessibility requirements.

To use this feature, position your mouse over the element(s) you'd like to test (a blue box will appear around them) and click the Test icon (which looks like a laboratory beaker).

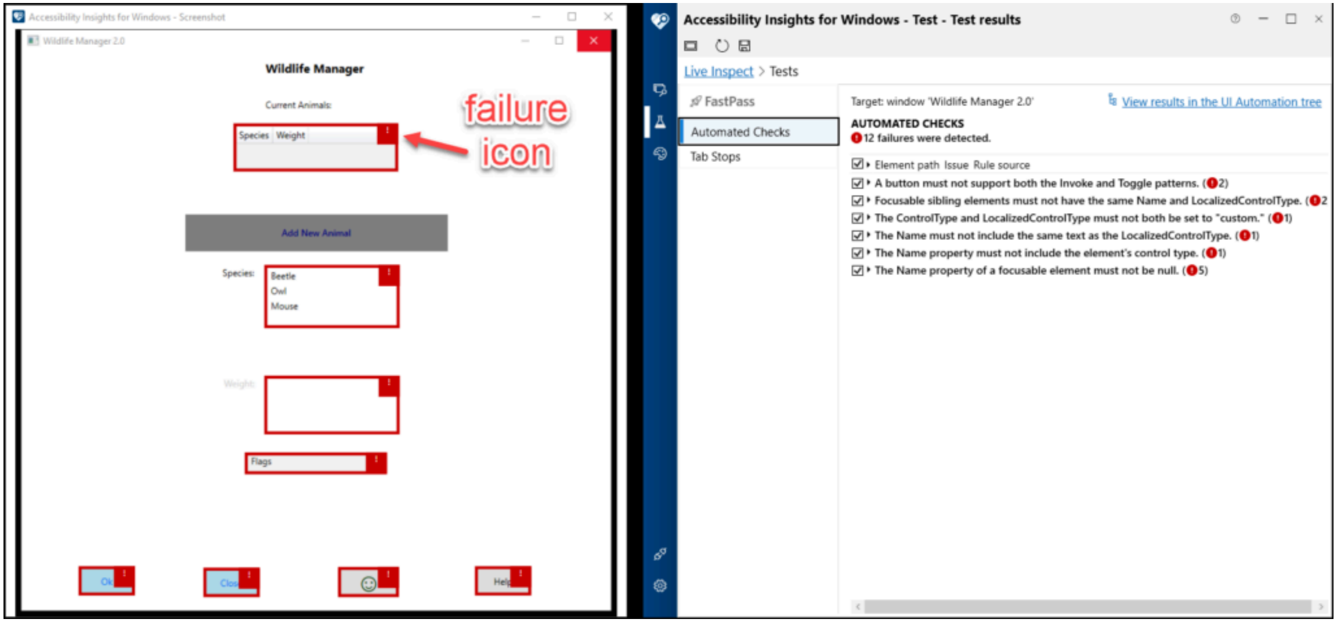


window 'Wildlife Manager 2.0'

Properties

Property	Value
Name	Wildlife Manager 2.0
ControlType	Window(50032)
LocalizedControlType	window
IsKeyboardFocusable	True
BoundingRectangle	[l=2131,t=27,r=2931,b=807]
AccessKey	*Property does not exist*
AcceleratorKey	*Property does not exist*
HelpText	*Property does not exist*
AriaProperties	*Property does not exist*
AriaRole	*Property does not exist*

Some elements will be flagged with issues. To select one, click on its failure icon (!).



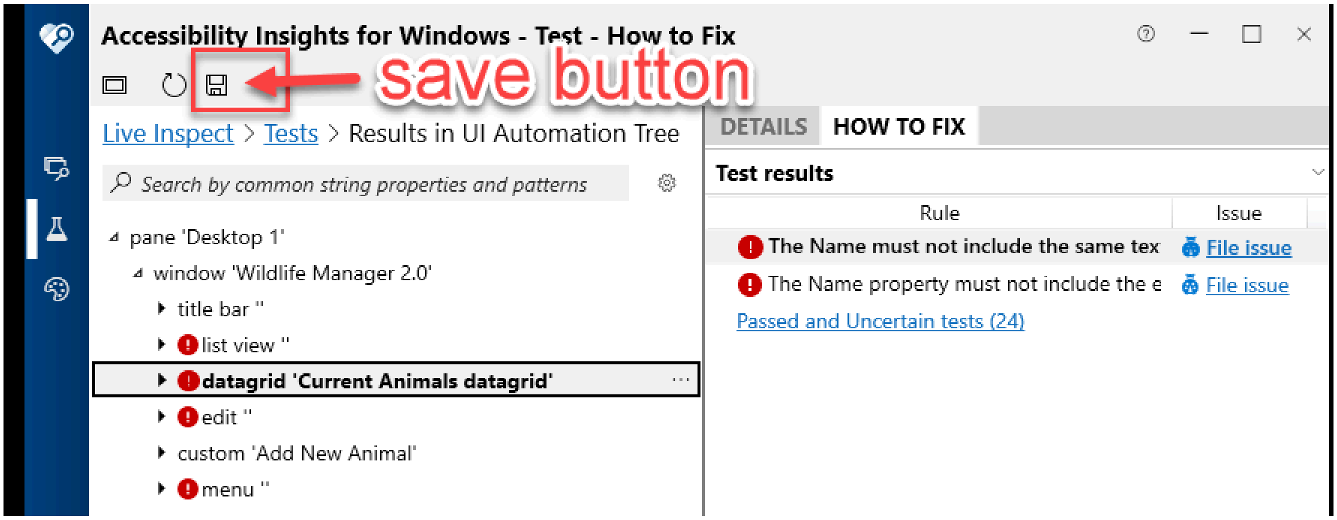
Target: window 'Wildlife Manager 2.0'

AUTOMATED CHECKS

12 failures were detected.

☑ Element path Issue Rule source
☑ A button must not support both the Invoke and Toggle patterns. (2)
☑ Focusable sibling elements must not have the same Name and LocalizedControlType. (2
☑ The ControlType and LocalizedControlType must not both be set to "custom." (1)
☑ The Name must not include the same text as the LocalizedControlType. (1)
☑ The Name property must not include the element's control type. (1)
☑ The Name property of a focusable element must not be null. (5)

Details will be displayed about the highlighted issue along with tips on how to fix it.



save button

Accessibility Insights for Windows - Test - How to Fix

Live Inspect > Tests > Results in UI Automation Tree

Search by common string properties and patterns

- pane 'Desktop 1'
  - window 'Wildlife Manager 2.0'
    - title bar ''
    - list view ''
    - datagrid 'Current Animals datagrid'
    - edit ''
    - custom 'Add New Animal'
    - menu ''

DETAILS HOW TO FIX

Test results

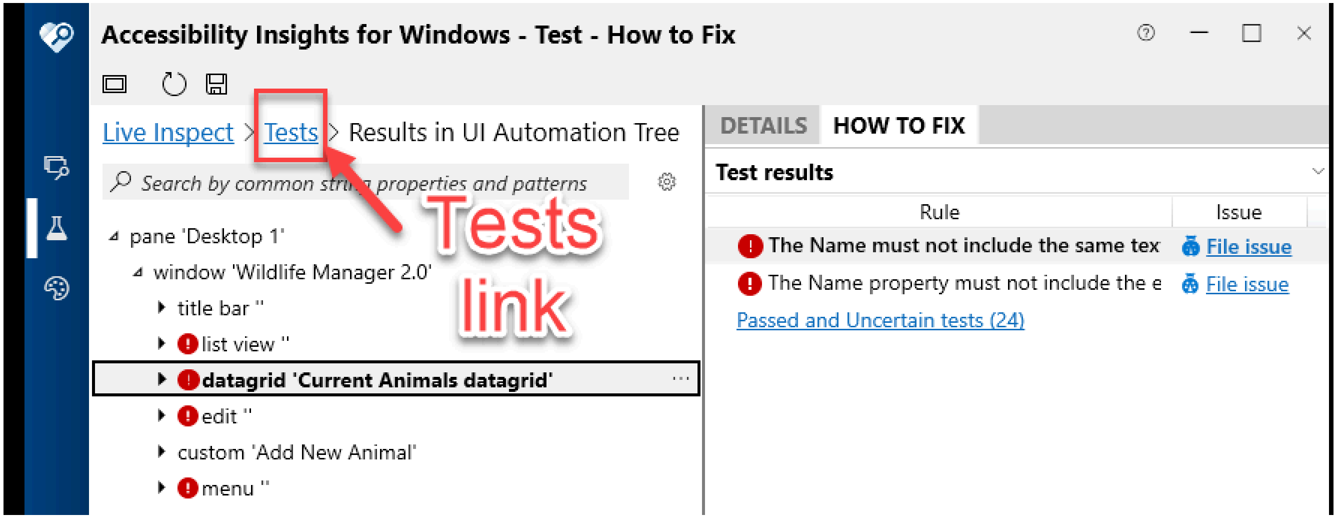
Rule	Issue
The Name must not include the same tex	File issue
The Name property must not include the e	File issue

Passed and Uncertain tests (24)

You can click the Save button (which looks like a diskette) to save a snapshot of the issue.

Tab Stops

The Tab Stops feature of the FastPass tests supports you in testing whether the tab progression in the application is accessible. To test the Tab Stops, click the Tests link.



Tests link

Accessibility Insights for Windows - Test - How to Fix

Live Inspect > Tests > Results in UI Automation Tree

Search by common string properties and patterns

- pane 'Desktop 1'
  - window 'Wildlife Manager 2.0'
    - title bar ''
    - list view ''
    - datagrid 'Current Animals datagrid'
    - edit ''
    - custom 'Add New Animal'
    - menu ''

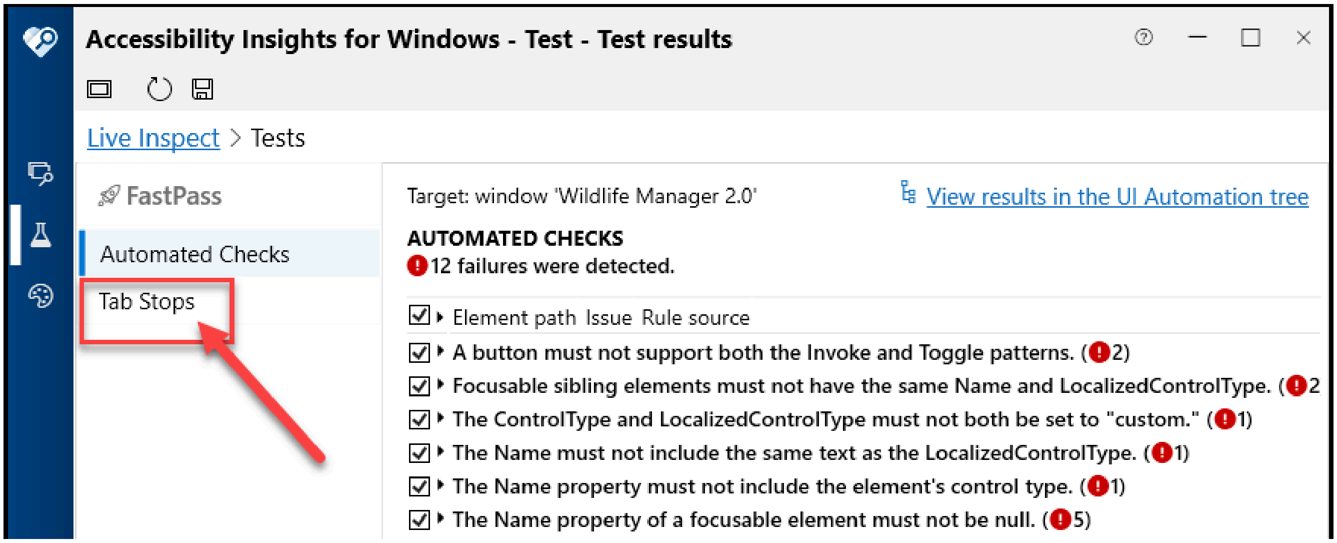
DETAILS HOW TO FIX

Test results

Rule	Issue
The Name must not include the same tex	File issue
The Name property must not include the e	File issue

Passed and Uncertain tests (24)

Then select Tab Stops.


Accessibility Insights for Windows - Test - Test results
Live Inspect > Tests
FastPass
Automated Checks
Tab Stops
View results in the UI Automation tree

Target: window 'Wildlife Manager 2.0'

AUTOMATED CHECKS

❗12 failures were detected.

☑ ▸ Element path Issue Rule source

☑ ▸ A button must not support both the Invoke and Toggle patterns. (❗2)

☑ ▸ Focusable sibling elements must not have the same Name and LocalizedControlType. (❗2

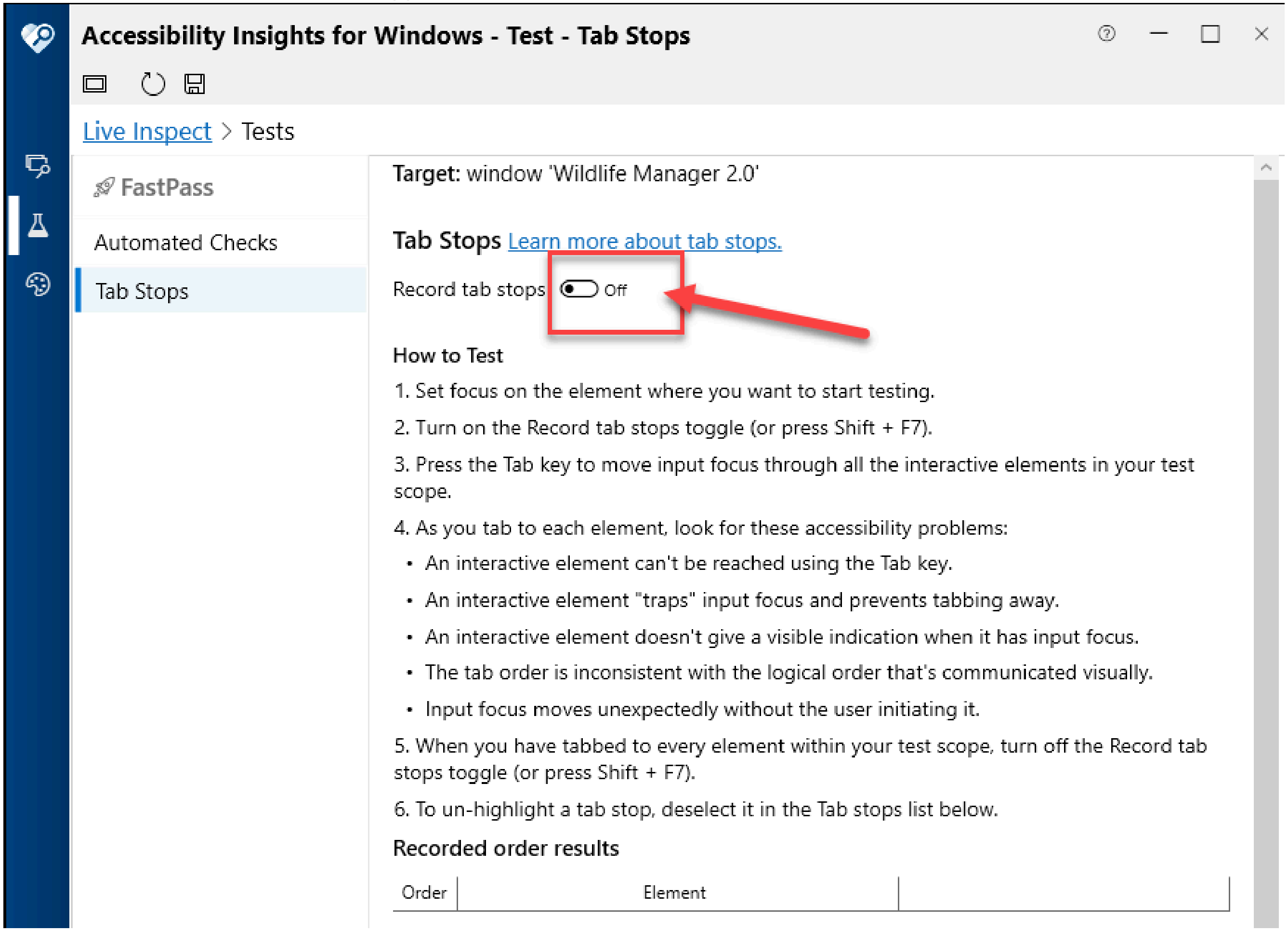
☑ ▸ The ControlType and LocalizedControlType must not both be set to "custom." (❗1)

☑ ▸ The Name must not include the same text as the LocalizedControlType. (❗1)

☑ ▸ The Name property must not include the element's control type. (❗1)

☑ ▸ The Name property of a focusable element must not be null. (❗5)

Switch the "Record tab stops" toggle to On.



Target: window 'Wildlife Manager 2.0'

Tab Stops Learn more about tab stops.

Record tab stops ○ Off

How to Test

1. Set focus on the element where you want to start testing.

2. Turn on the Record tab stops toggle (or press Shift + F7).

3. Press the Tab key to move input focus through all the interactive elements in your test scope.

4. As you tab to each element, look for these accessibility problems:

- An interactive element can't be reached using the Tab key.
- An interactive element "traps" input focus and prevents tabbing away.
- An interactive element doesn't give a visible indication when it has input focus.
- The tab order is inconsistent with the logical order that's communicated visually.
- Input focus moves unexpectedly without the user initiating it.

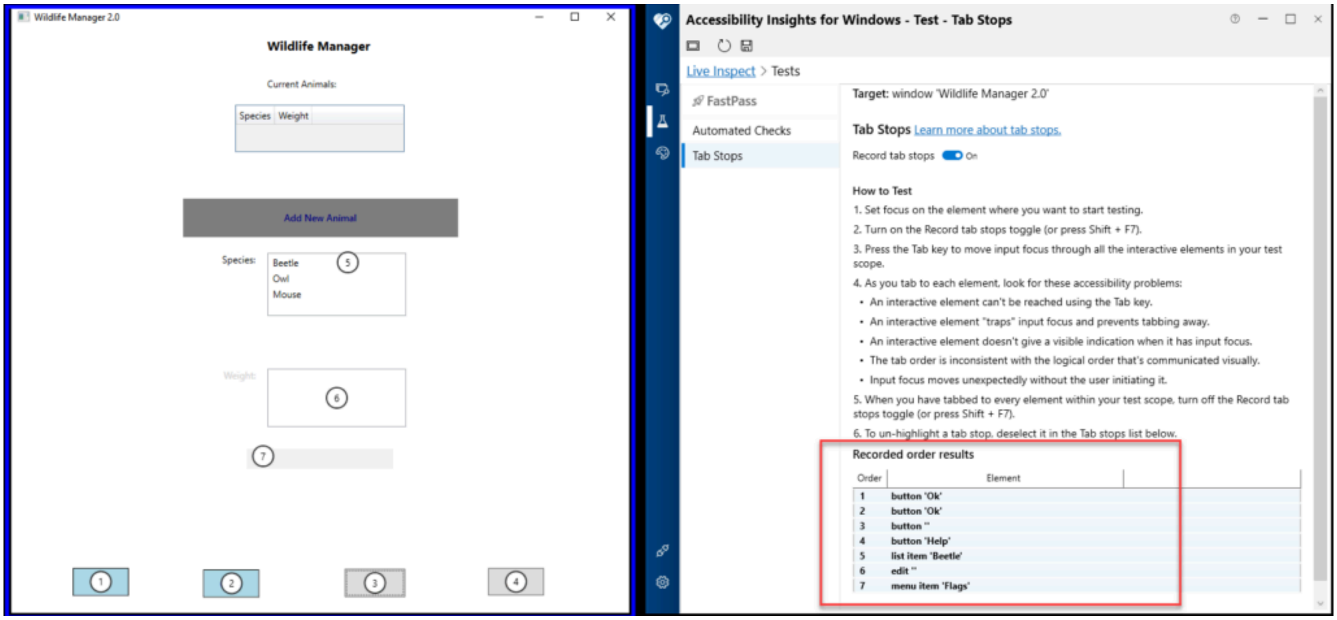
5. When you have tabbed to every element within your test scope, turn off the Record tab stops toggle (or press Shift + F7).

6. To un-highlight a tab stop, deselect it in the Tab stops list below.

Recorded order results

Order	Element	

Then press the Tab key several times, and each tab stop will be listed in the "Recorded order results" table. This will make it easy to identify missing tab stops or incorrect tab order.



Target: window 'Wildlife Manager 2.0'

Tab Stops Learn more about tab stops.

Record tab stops On

How to Test

1. Set focus on the element where you want to start testing.
2. Turn on the Record tab stops toggle (or press Shift + F7).
3. Press the Tab key to move input focus through all the interactive elements in your test scope.
4. As you tab to each element, look for these accessibility problems:
   - An interactive element can't be reached using the Tab key.
   - An interactive element "traps" input focus and prevents tabbing away.
   - An interactive element doesn't give a visible indication when it has input focus.
   - The tab order is inconsistent with the logical order that's communicated visually.
   - Input focus moves unexpectedly without the user initiating it.
5. When you have tabbed to every element within your test scope, turn off the Record tab stops toggle (or press Shift + F7).
6. To un-highlight a tab stop, deselect it in the Tab stops list below.

Recorded order results

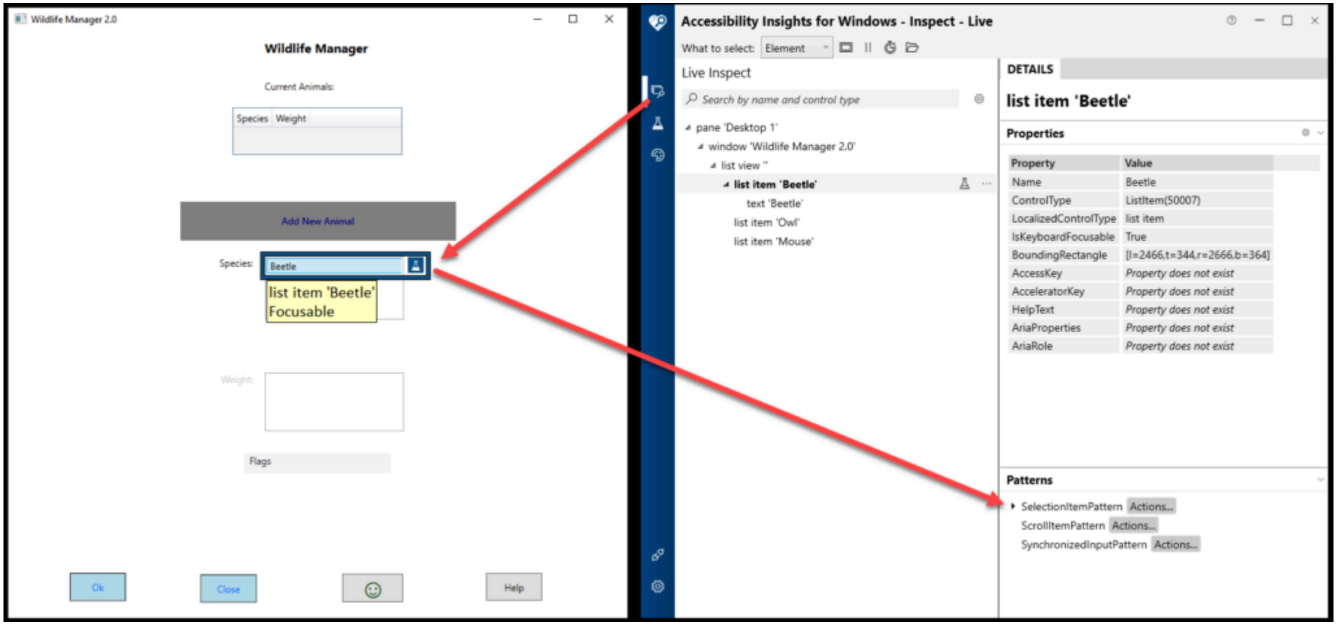
Order	Element
1	button 'Ok'
2	button 'Ok'
3	button ''
4	button 'Help'
5	list item 'Beetle'
6	edit ''
7	menu item 'Flags'

Troubleshooting

There are two components of the Troubleshooting feature that are relatively important to keep in mind. The first allows you to see the patterns supported by a control and invoke their UIA methods to see whether it responds correctly to user input. The second records application events.

Patterns

To use this feature, select the Inspect button in Accessibility Insights. Then select a UI element in the application. Then select a Pattern you'd like to inspect in Accessibility Insights.



- pane 'Desktop 1'
  - window 'Wildlife Manager 2.0'
    - list view ''
      - list item 'Beetle'
        - text 'Beetle'
      - list item 'Owl'
      - list item 'Mouse'

list item 'Beetle'

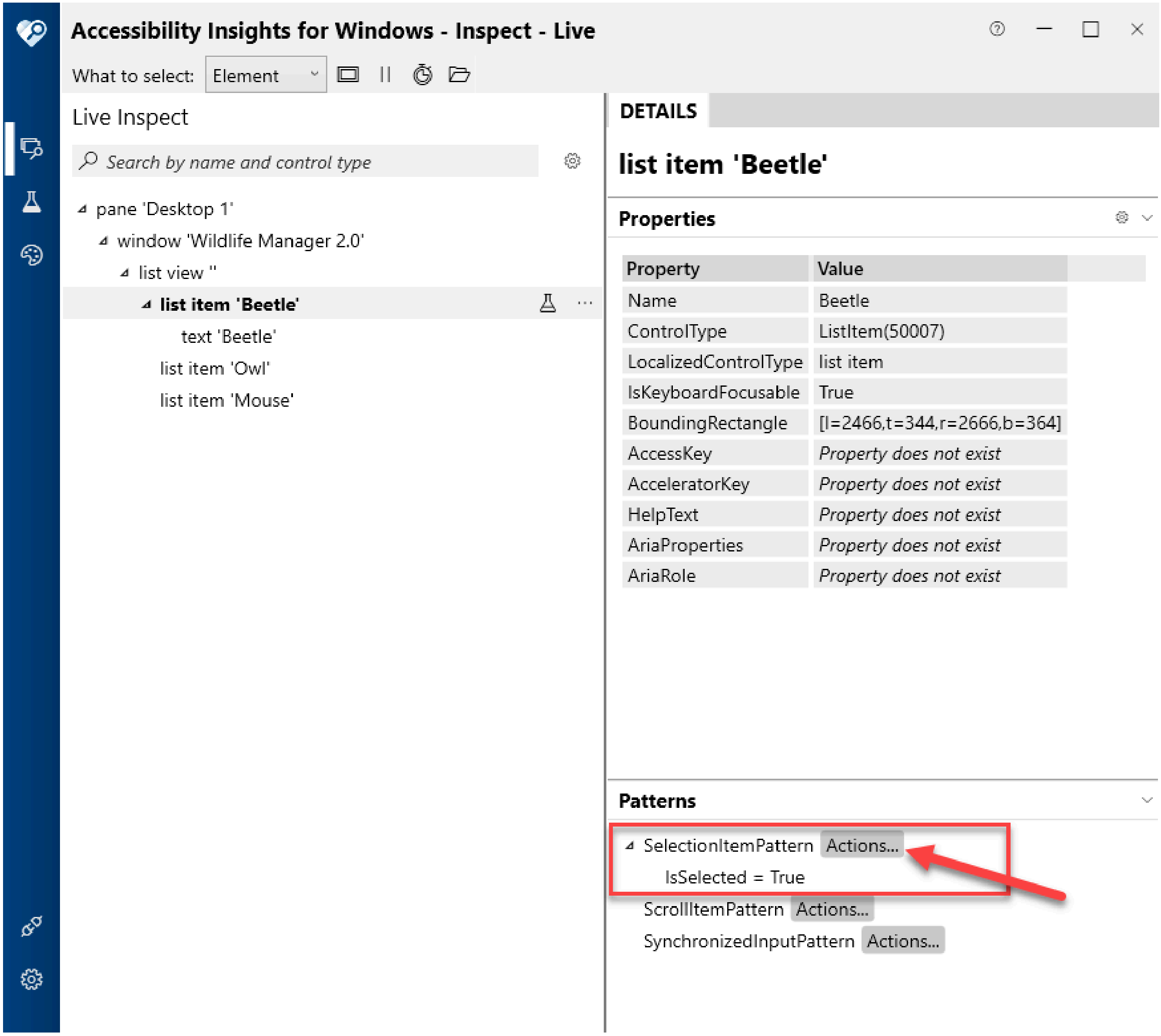
Properties

Property	Value
Name	Beetle
ControlType	ListItem(50007)
LocalizedControlType	list item
IsKeyboardFocusable	True
BoundingRectangle	[l=2466,t=344,r=2666,b=364]
AccessKey	*Property does not exist*
AcceleratorKey	*Property does not exist*
HelpText	*Property does not exist*
AriaProperties	*Property does not exist*
AriaRole	*Property does not exist*

Patterns

- SelectionItemPattern Actions...
- ScrollItemPattern Actions...
- SynchronizedInputPattern Actions...

Then click on the pattern's "Actions…" button.



Accessibility Insights for Windows - Inspect - Live

What to select: Element

Live Inspect

Search by name and control type

- pane 'Desktop 1'
  - window 'Wildlife Manager 2.0'
    - list view ''
      - **list item 'Beetle'**
        - text 'Beetle'
      - list item 'Owl'
      - list item 'Mouse'

DETAILS

list item 'Beetle'

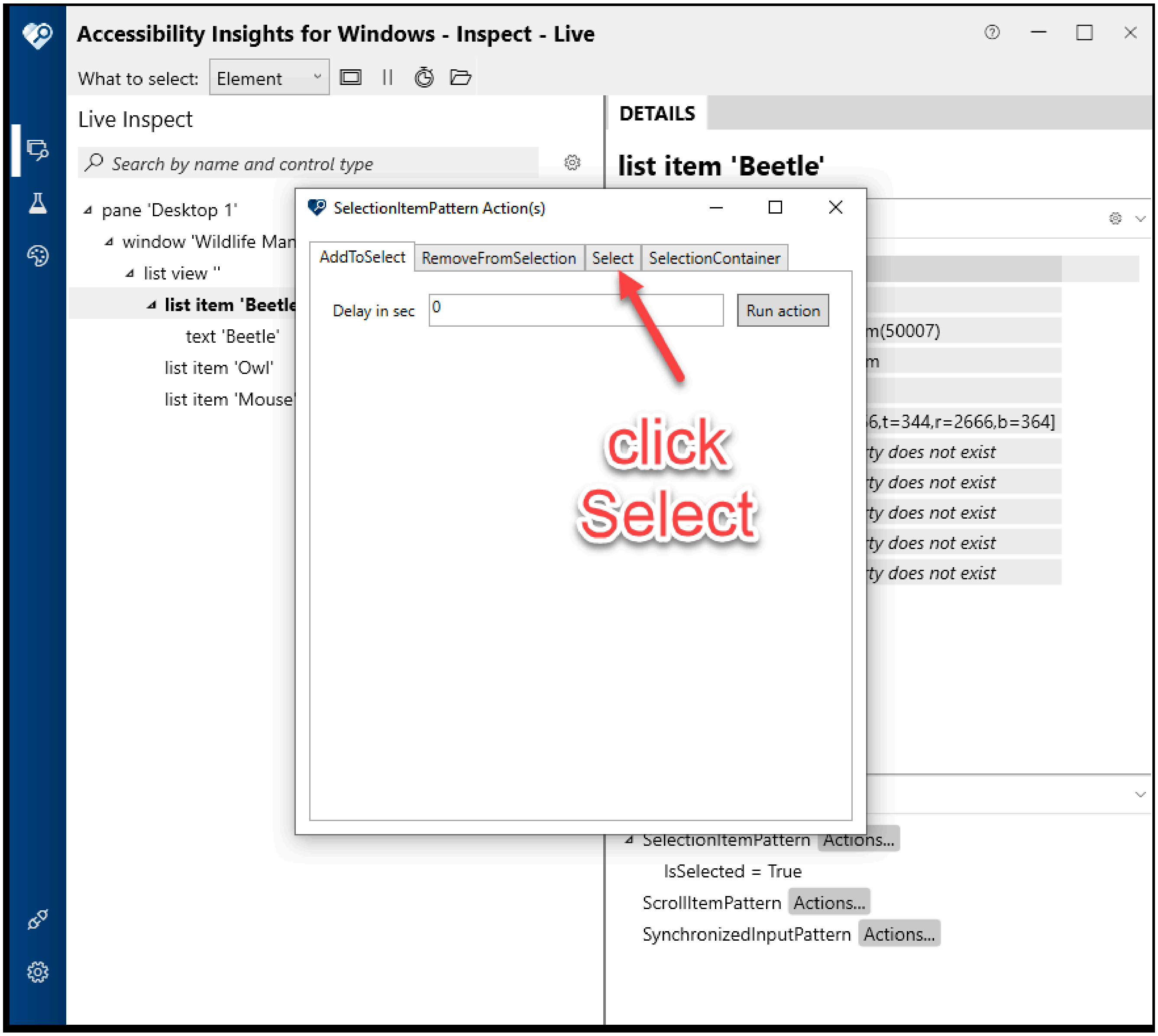
Properties

Property	Value
Name	Beetle
ControlType	ListItem(50007)
LocalizedControlType	list item
IsKeyboardFocusable	True
BoundingRectangle	[l=2466,t=344,r=2666,b=364]
AccessKey	*Property does not exist*
AcceleratorKey	*Property does not exist*
HelpText	*Property does not exist*
AriaProperties	*Property does not exist*
AriaRole	*Property does not exist*

Patterns

- SelectionItemPattern Actions...
  - IsSelected = True
- ScrollItemPattern Actions...
- SynchronizedInputPattern Actions...

An Actions dialogue will open. Click the Select tab.

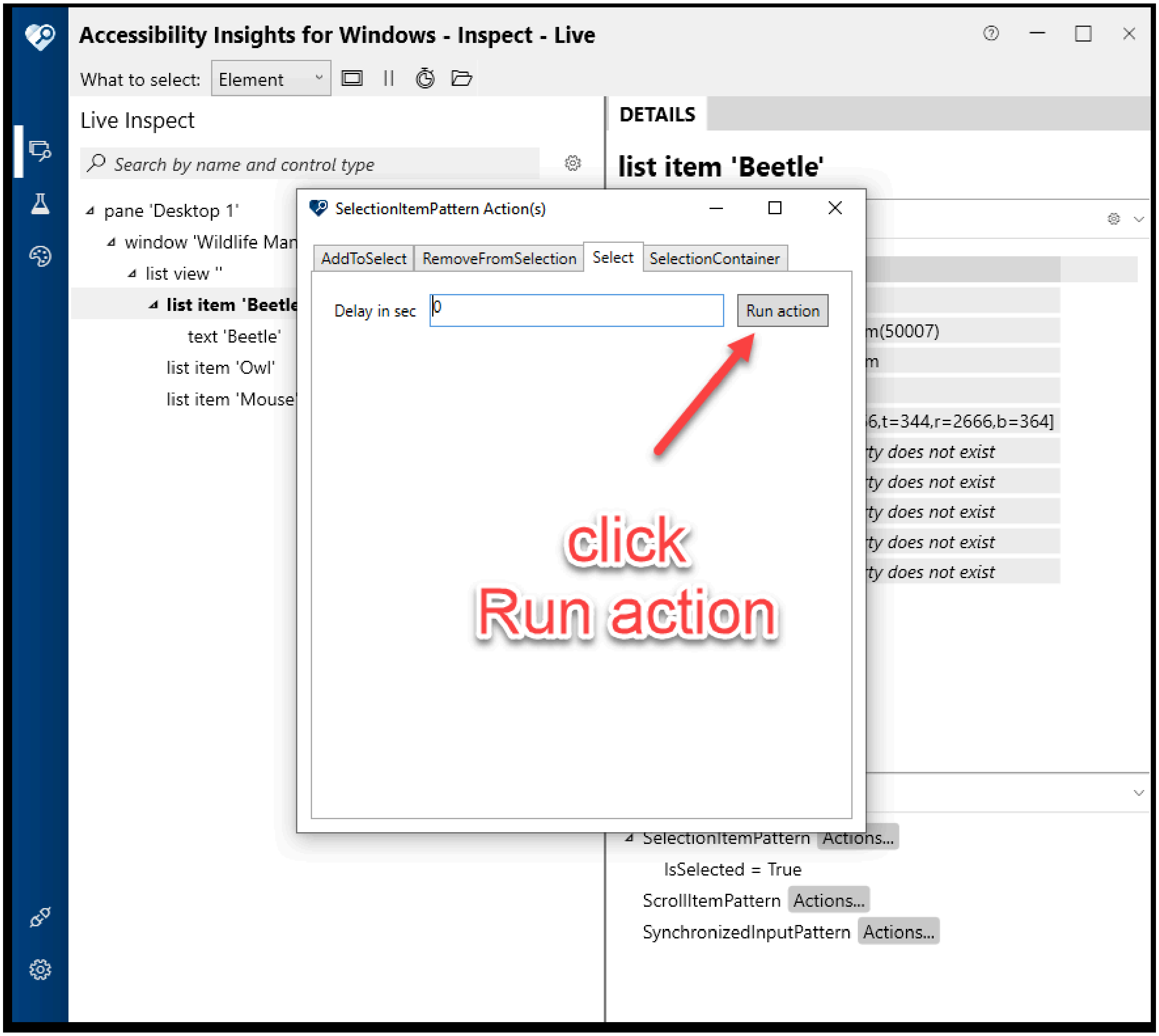


What to select: Element

Delay in sec 0

click Select

Then click the “Run action” button.



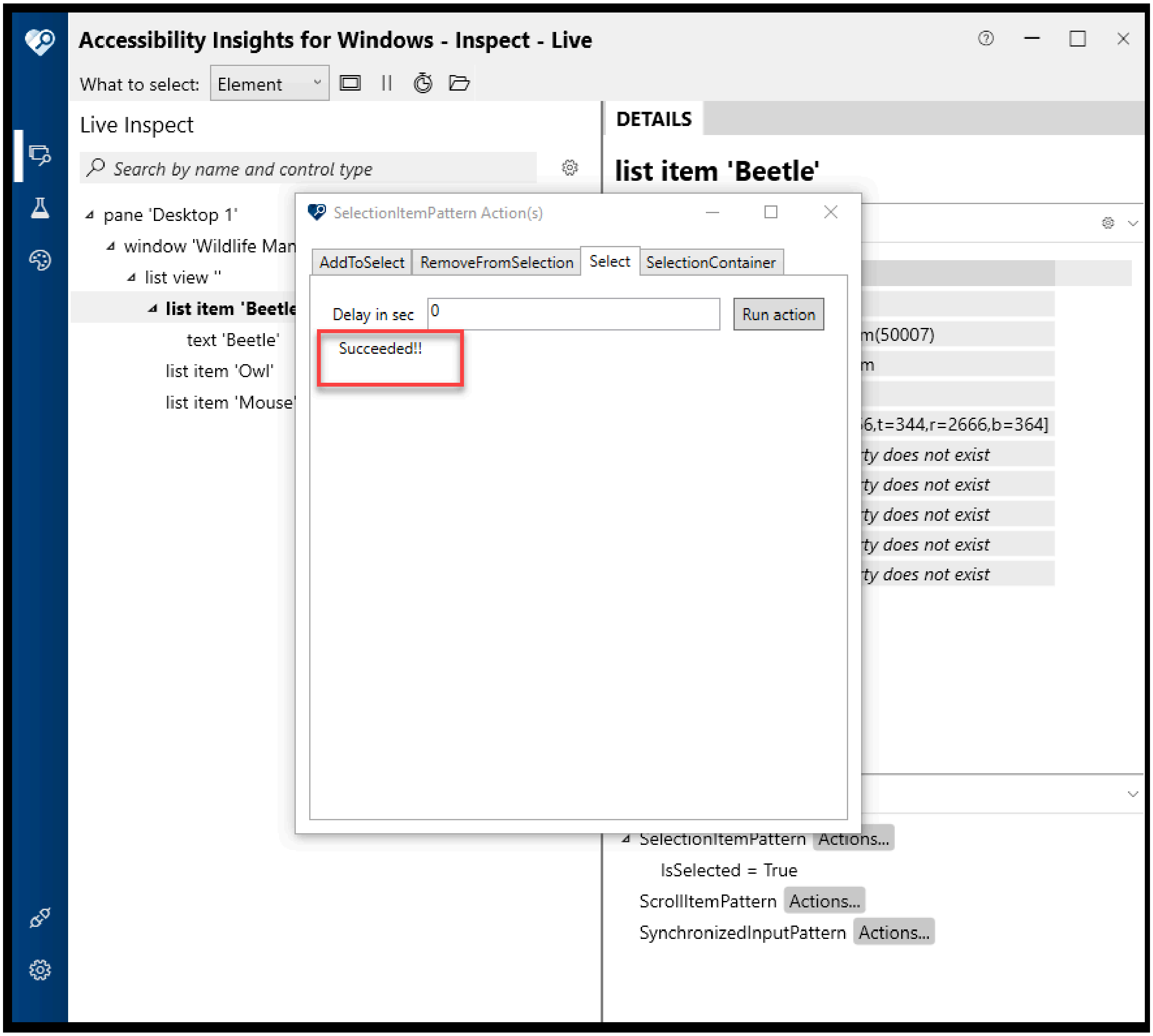
What to select: Element

Delay in sec 0

IsSelected = True

click
Run action

And the “Succeeded” message appears.

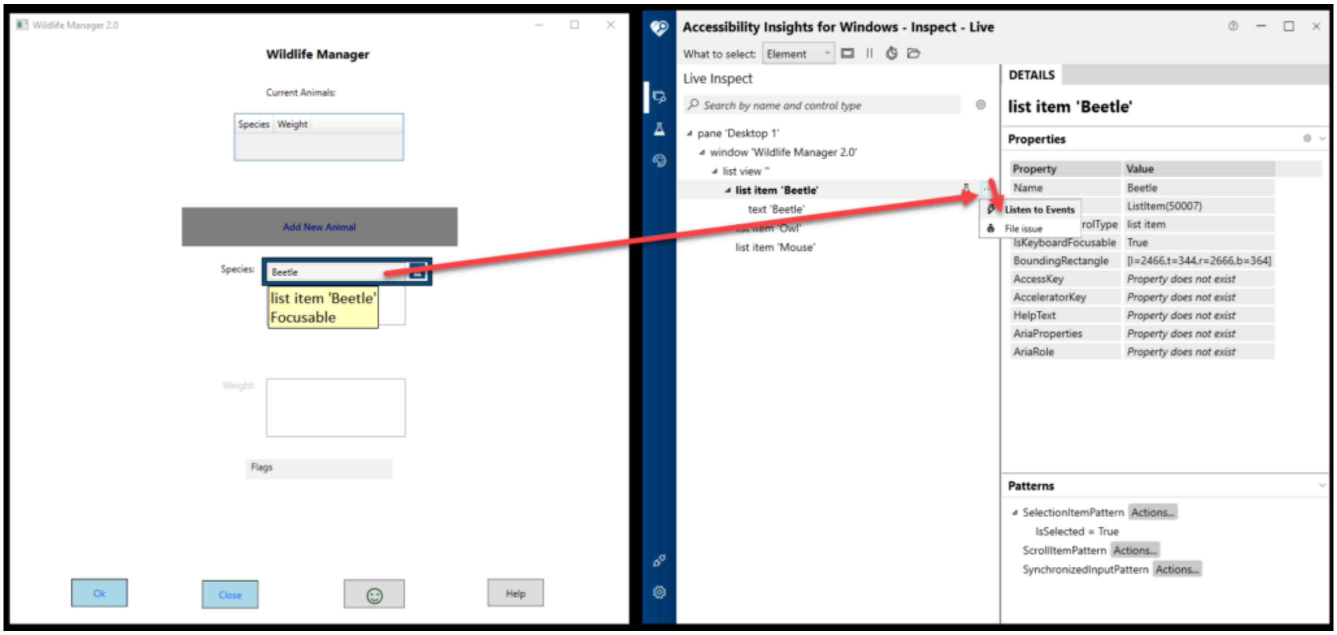


Delay in sec 0

Succeeded!!

Record application events

Another part of troubleshooting is to record application events. To do this, select an element in the application. This element will become highlighted in Insights, click the “More options” (...) button, and select “Listen to Events”



pane 'Desktop 1'
window 'Wildlife Manager 2.0'
list view ''
list item 'Beetle'
text 'Beetle'
list item 'Owl'
list item 'Mouse'

list item 'Beetle'

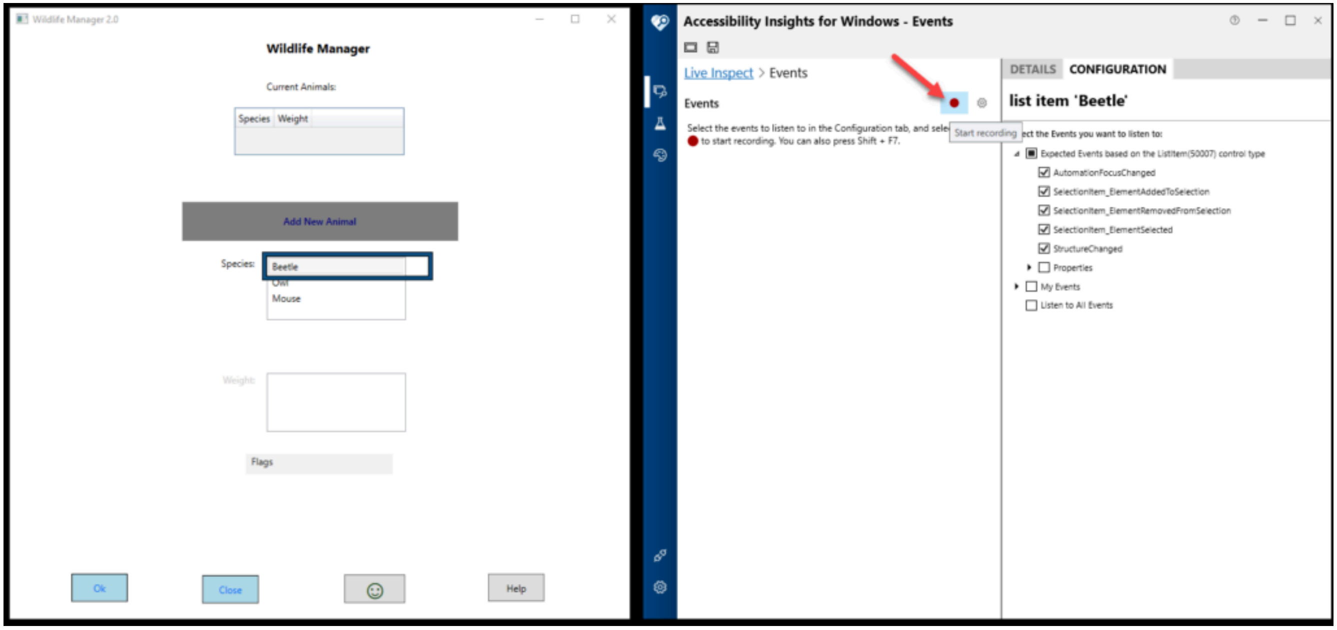
Properties

Property	Value
Name	Beetle
	ListItem(50007)
rolType	list item
IsKeyboardFocusable	True
BoundingRectangle	[l=2466,t=344,r=2666,b=364]
AccessKey	*Property does not exist*
AcceleratorKey	*Property does not exist*
HelpText	*Property does not exist*
AriaProperties	*Property does not exist*
AriaRole	*Property does not exist*

Patterns

SelectionItemPattern Actions...
IsSelected = True
ScrollItemPattern Actions...
SynchronizedInputPattern Actions...

Then click the “Start recording” button.



Events

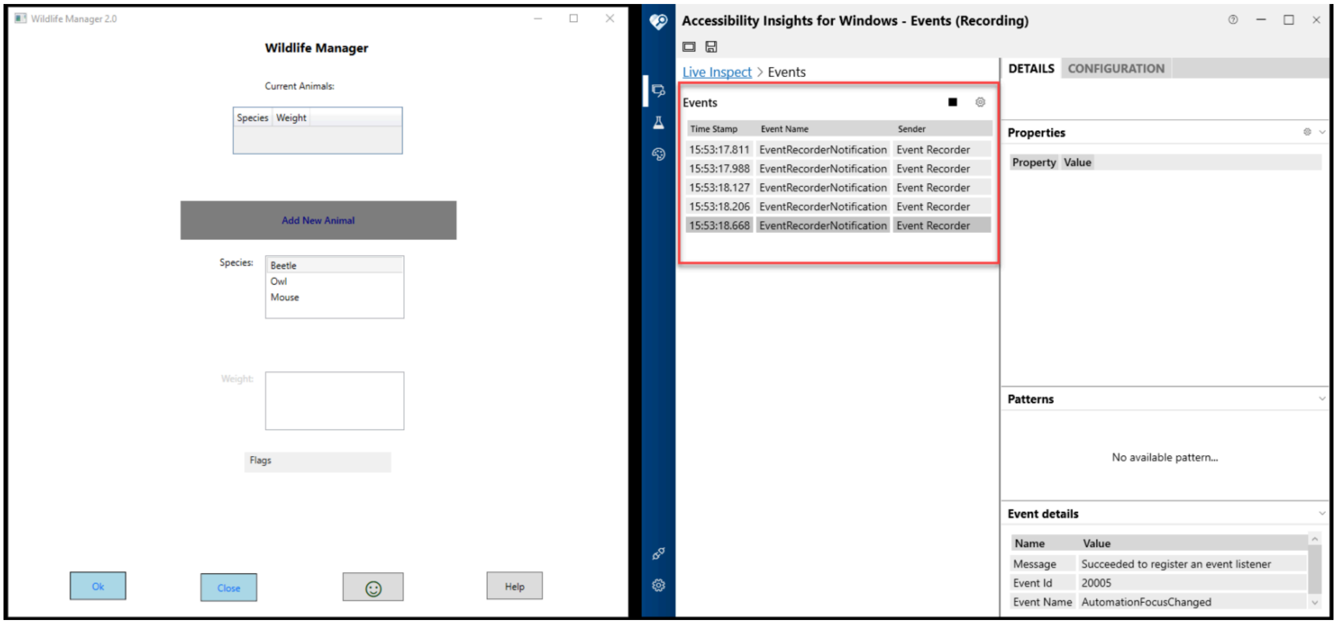
Select the events to listen to in the Configuration tab, and sele
● to start recording. You can also press Shift + F7.

list item 'Beetle'

ect the Events you want to listen to:

- ■ Expected Events based on the ListItem(50007) control type
  - ☑ AutomationFocusChanged
  - ☑ SelectionItem_ElementAddedToSelection
  - ☑ SelectionItem_ElementRemovedFromSelection
  - ☑ SelectionItem_ElementSelected
  - ☑ StructureChanged
  - □ Properties
- □ My Events
- □ Listen to All Events

The invoked events will be logged under the Events heading.



Events

Time Stamp	Event Name	Sender
15:53:17.811	EventRecorderNotification	Event Recorder
15:53:17.988	EventRecorderNotification	Event Recorder
15:53:18.127	EventRecorderNotification	Event Recorder
15:53:18.206	EventRecorderNotification	Event Recorder
15:53:18.668	EventRecorderNotification	Event Recorder

No available pattern...

Name	Value
Message	Succeeded to register an event listener
Event Id	20005
Event Name	AutomationFocusChanged

Resources:

- Introduction to Accessibility Insights for Windows
- Accessibility Insights for Windows
- Inspect (aka Live Inspect)
- FastPass in Accessibility Insights for Windows
- Issue filing in Accessibility Insights for Windows
- UI Automation Overview

INSTALLATION-FREE TESTING

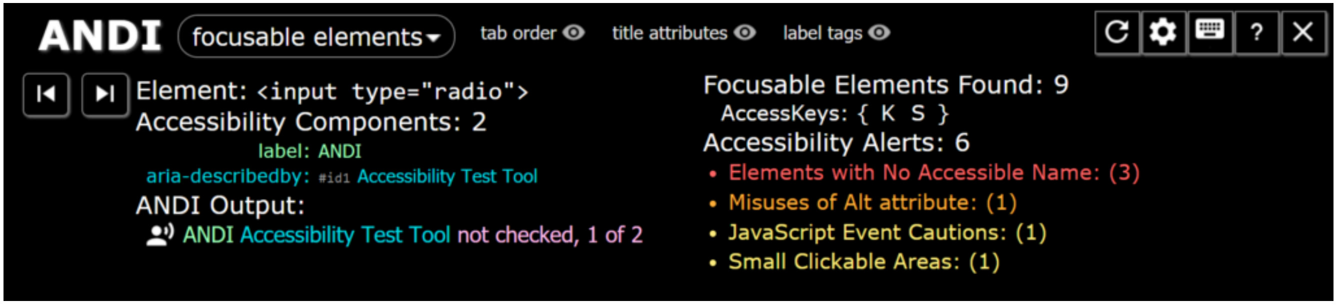
In some organizations, you may not have permission to install some applications on your computer. Fortunately, there are accessibility testing tools that do not require install access.

ANDI bookmarklet

ANDI (Accessible Name & Description Inspector) is a set of tools organized into menus, developed by the US Social Security Administration. It includes several tools for visually highlighting page features, like tab order, heading levels, and alt text for images.

ANDI can be ‘installed’ by dragging a link to your browser’s bookmarks bar – see the ANDI Install page for instructions.

Learn about ANDI’s features on this Tutorial page.



Element: <input type="radio">
Accessibility Components: 2
label: ANDI
aria-describedby: #id1 Accessibility Test Tool
ANDI Output:
ANDI Accessibility Test Tool not checked, 1 of 2

Focusable Elements Found: 9
AccessKeys: { K S }
Accessibility Alerts: 6
- Elements with No Accessible Name: (3)
- Misuses of Alt attribute: (1)
- JavaScript Event Cautions: (1)
- Small Clickable Areas: (1)

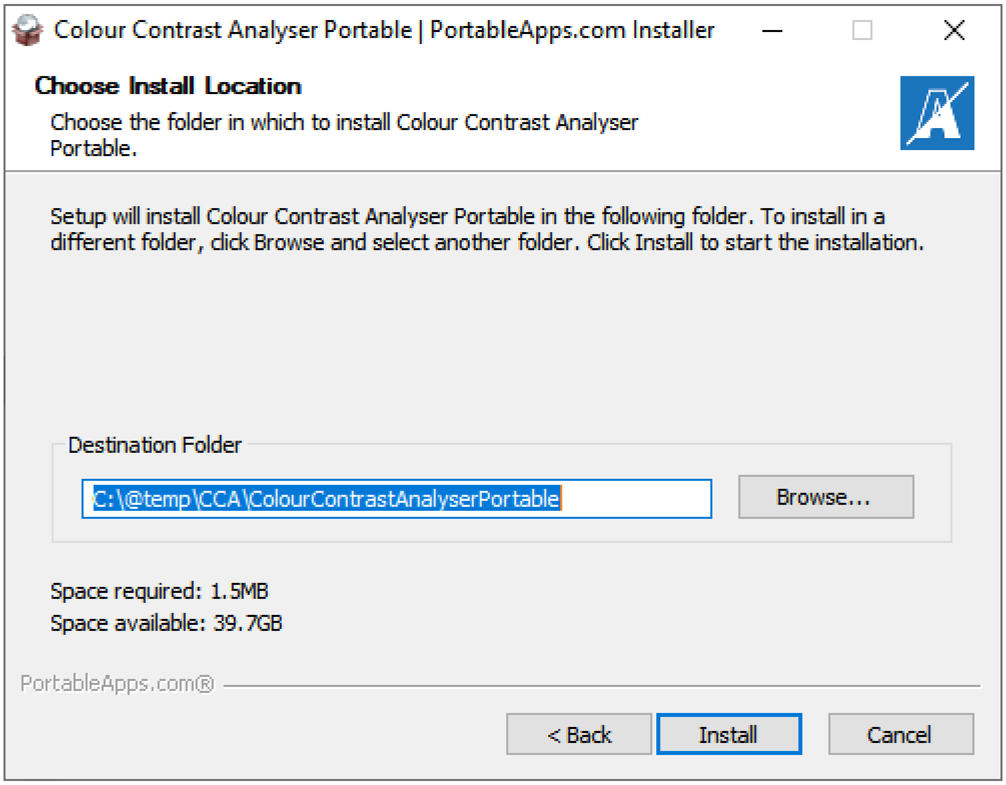
Portable apps

Some versions of applications are specifically designed to be “portable”, meaning they can run on your computer from a folder or USB drive.

Colour Contrast Analyser (CCA) as a portable app

The portable version of CCA can be downloaded from PortableApps.com.

Open the “installer” and select a folder to place the program files. You can delete them later if you like.



Colour Contrast Analyser Portable | PortableApps.com Installer

Choose Install Location

Choose the folder in which to install Colour Contrast Analyser Portable.

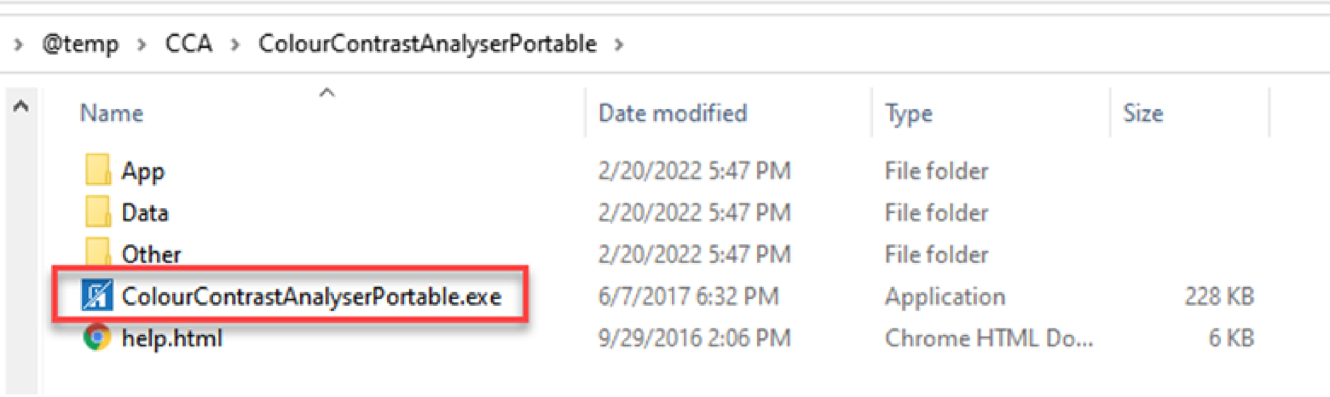
Setup will install Colour Contrast Analyser Portable in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder: C:\@temp\CCA\ColourContrastAnalyserPortable

Space required: 1.5MB

Space available: 39.7GB

After the "install" process is complete, you can run CCA by double-clicking on the file ColourContrastAnalyser.exe. You can create a shortcut to this file and place it on your Desktop if you like.



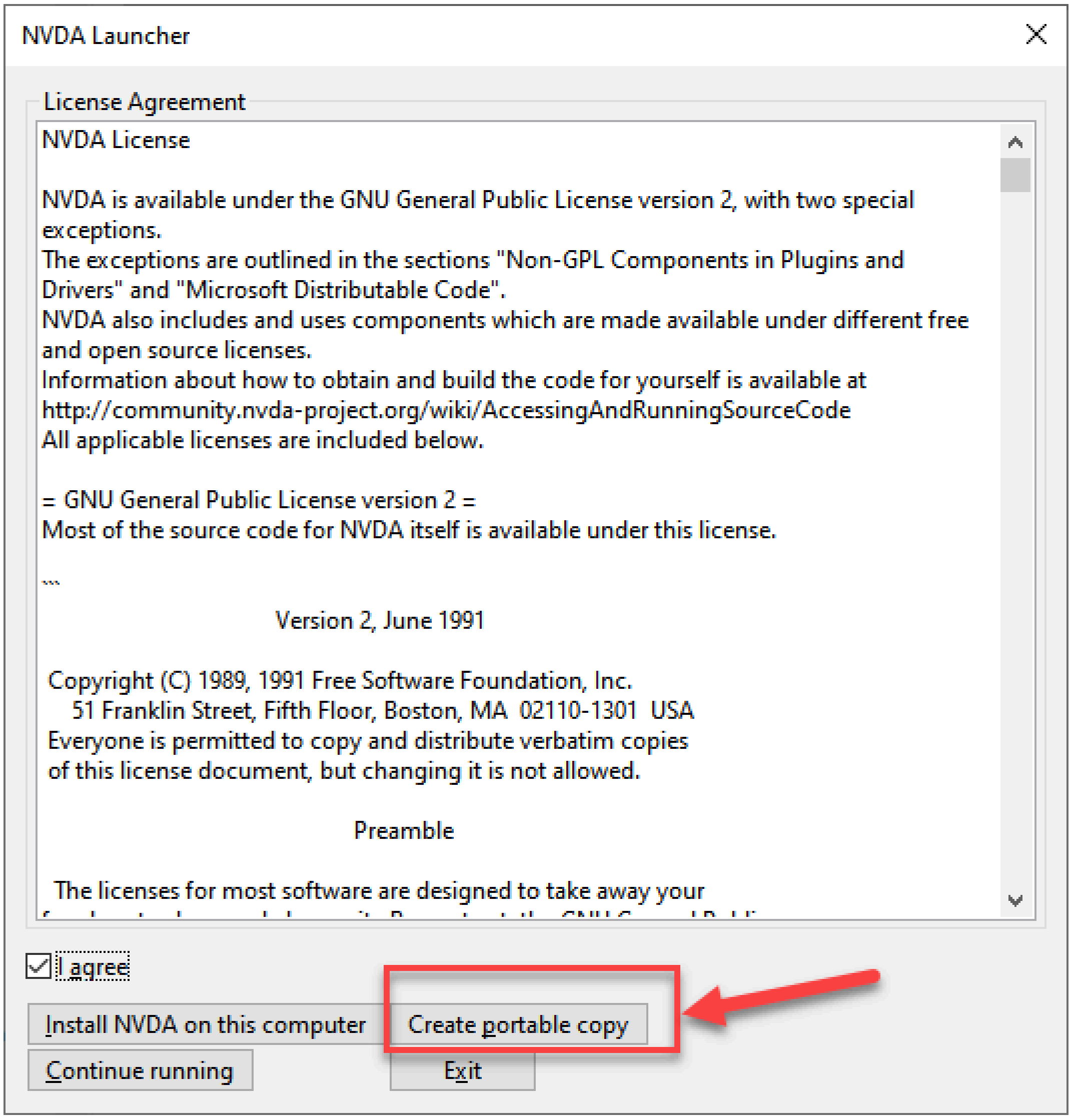
› @temp › CCA › ColourContrastAnalyserPortable ›

Name	Date modified	Type	Size
App	2/20/2022 5:47 PM	File folder	
Data	2/20/2022 5:47 PM	File folder	
Other	2/20/2022 5:47 PM	File folder	
ColourContrastAnalyserPortable.exe	6/7/2017 6:32 PM	Application	228 KB
help.html	9/29/2016 2:06 PM	Chrome HTML Do...	6 KB

NVDA as a portable app

NonVisual Desktop Access (**NVDA**) is a free screen reader for Windows. To install NVDA as a portable app, begin by downloading the software from the NVDA download page.

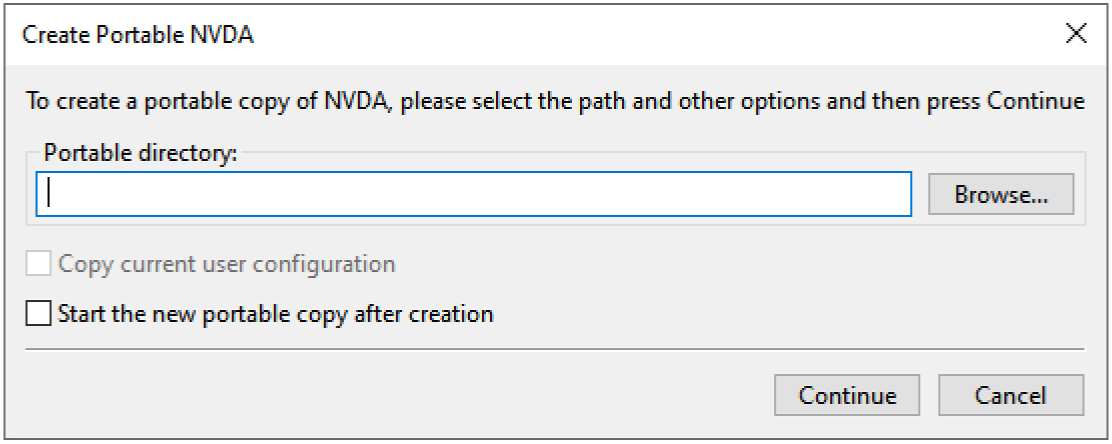
Open the install file and click the "Create portable copy" button instead of the "Install..." button.



NVDA License

NVDA is available under the GNU General Public License version 2, with two special exceptions.
The exceptions are outlined in the sections "Non-GPL Components in Plugins and Drivers" and "Microsoft Distributable Code".
NVDA also includes and uses components which are made available under different free and open source licenses.
Information about how to obtain and build the code for yourself is available at http://community.nvda-project.org/wiki/AccessingAndRunningSourceCode
All applicable licenses are included below.

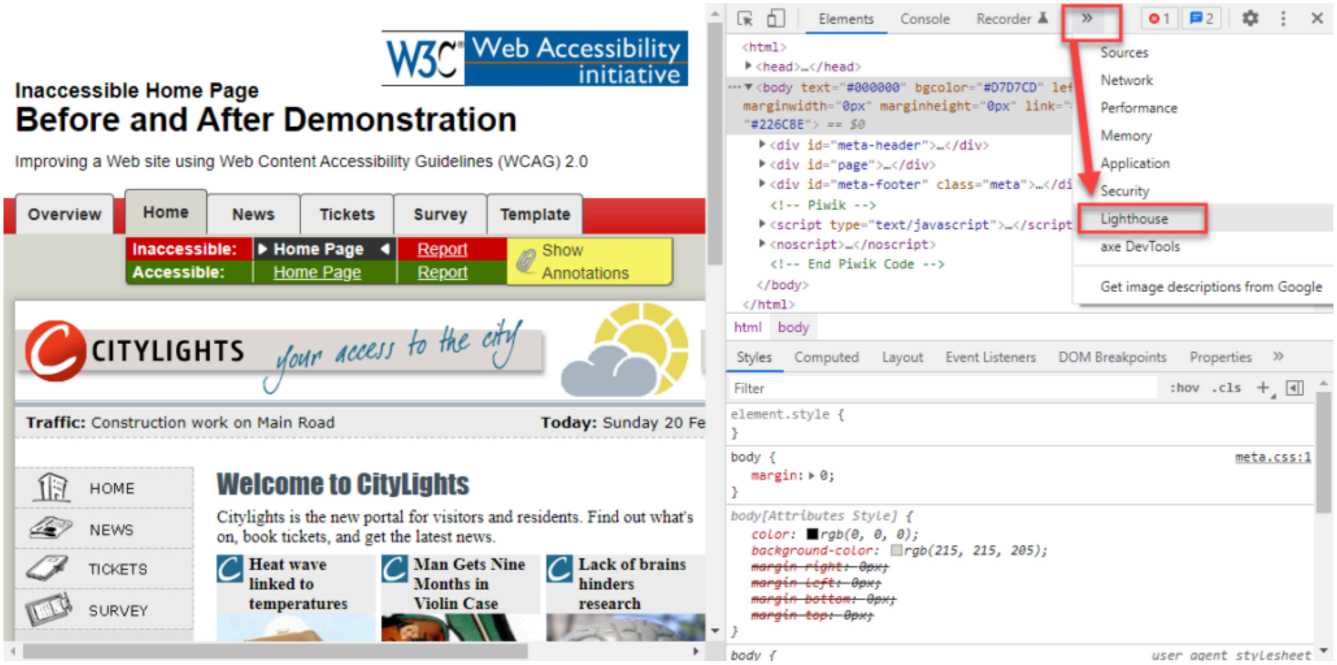
= GNU General Public License version 2 =
Most of the source code for NVDA itself is available under this license.

```

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your

☑ I agree

Then enter (or browse to) a directory where you would like the program files to be stored.



To create a portable copy of NVDA, please select the path and other options and then press Continue

☐ Copy current user configuration

☐ Start the new portable copy after creation

Then navigate to that folder containing the program files and open the file nvda.exe.

You can also make a Desktop shortcut.
```

Lighthouse for Chrome

The Chrome Browser includes Lighthouse. Lighthouse uses Deque's accessibility engine, axe, to scan for accessibility issues.

To scan a web page with Lighthouse, load the web page first. Then press F12 or Control+Shift+i (Command+Shift+i on MacOS), click the "More tabs" (») button, and then select Lighthouse.

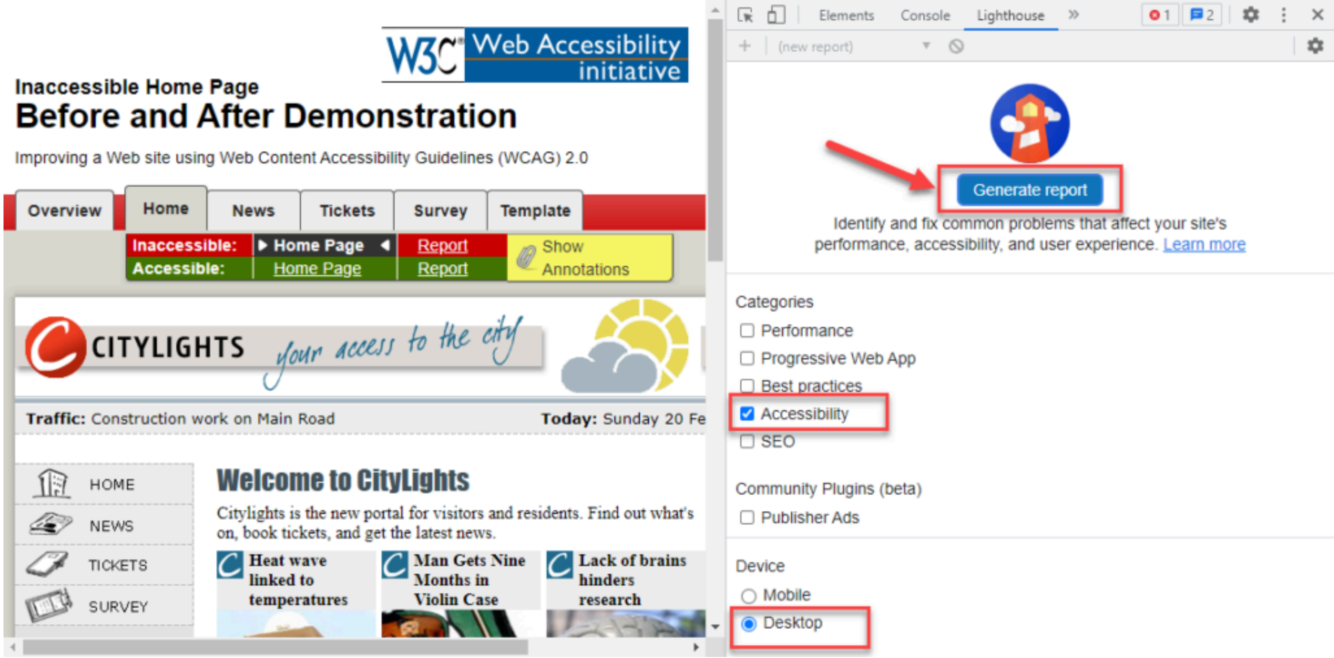


Citylights is the new portal for visitors and residents. Find out what's on, book tickets, and get the latest news.

```
<html>
 <head>…</head>
…<body text="#000000" bgcolor="#D7D7CD" lef
  marginwidth="0px" marginheight="0px" link="
  "#226C8E"> == $0
   <div id="meta-header">…</div>
   <div id="page">…</div>
   <div id="meta-footer" class="meta">…</di
    <!-- Piwik -->
   <script type="text/javascript">…</script
   <noscript>…</noscript>
    <!-- End Piwik Code -->
  </body>
</html>
```

```
element.style {
}
body {
   margin: ▸ 0;
}
body[Attributes Style] {
   color: ■rgb(0, 0, 0);
   background-color: □rgb(215, 215, 205);
   ~~margin-right: 0px;~~
   ~~margin-left: 0px;~~
   ~~margin-bottom: 0px;~~
   ~~margin-top: 0px;~~
}
body {
```

Lighthouse can run many tests. You can select or deselect the tests that you want to include. Select a device (Desktop or Mobile) and click the "Generate report" button.



Citylights is the new portal for visitors and residents. Find out what's on, book tickets, and get the latest news.

Identify and fix common problems that affect your site's performance, accessibility, and user experience. Learn more

Categories

- ☐ Performance
- ☐ Progressive Web App
- ☐ Best practices
- ☑ Accessibility
- ☐ SEO

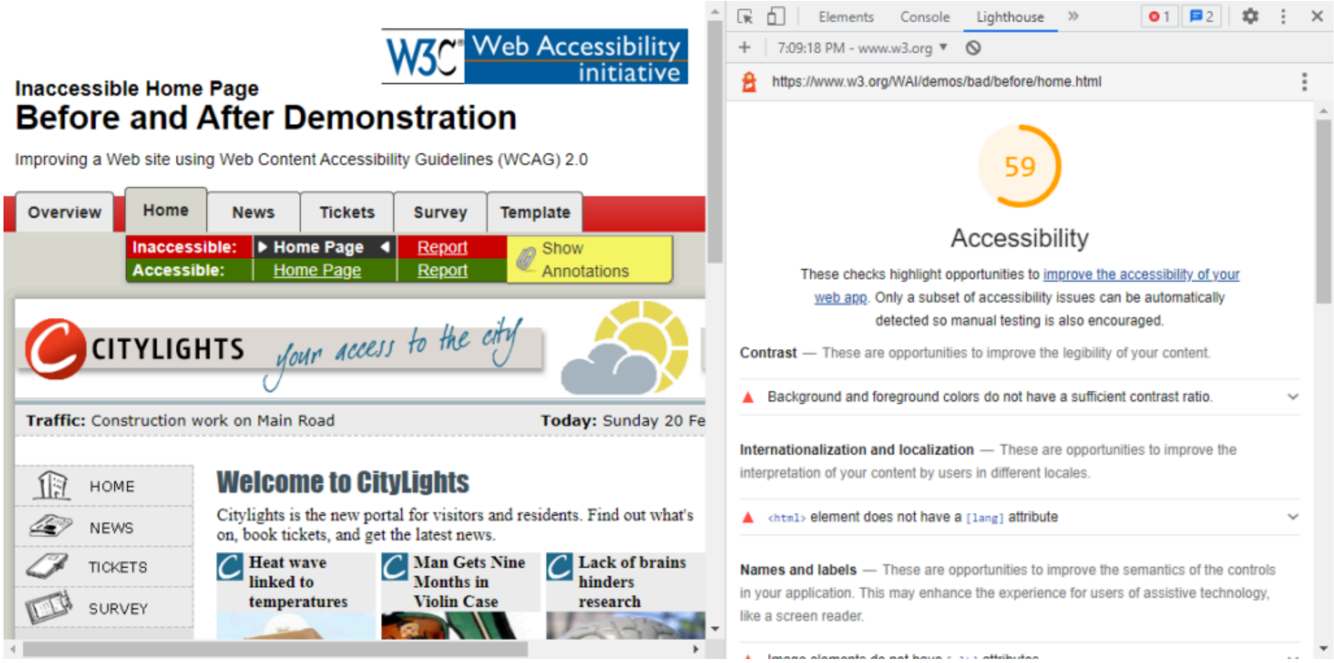
Community Plugins (beta)

- ☐ Publisher Ads

Device

- ○ Mobile
- ● Desktop

Lighthouse will display a report with an overall grade and a list of defects.


Elements
Console
Lighthouse
7:09:18 PM - www.w3.org
https://www.w3.org/WAI/demos/bad/before/home.html
W3C Web Accessibility initiative
Overview
Home
News
Tickets
Survey
Template
Inaccessible: Home Page Report
Accessible: Home Page Report
Show Annotations
CITYLIGHTS your access to the city
HOME
NEWS
TICKETS
SURVEY
59

Inaccessible Home Page

Before and After Demonstration

Improving a Web site using Web Content Accessibility Guidelines (WCAG) 2.0

Traffic: Construction work on Main Road   Today: Sunday 20 Fe

Welcome to CityLights

Citylights is the new portal for visitors and residents. Find out what's on, book tickets, and get the latest news.

Heat wave linked to temperatures | Man Gets Nine Months in Violin Case | Lack of brains hinders research

Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

Contrast — These are opportunities to improve the legibility of your content.

▲ Background and foreground colors do not have a sufficient contrast ratio.

Internationalization and localization — These are opportunities to improve the interpretation of your content by users in different locales.

▲ `<html>` element does not have a `[lang]` attribute

Names and labels — These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

Sa11y

Sa11y is an accessibility quality assurance tool that visually highlights common accessibility and **usability** issues. Sa11y helps content authors to identify errors or warnings by providing a tooltip on how to fix them. Sa11y works best in a templated content management system (CMS), although it's also available as a bookmarklet.

Resources:

- ANDI bookmarklet
- Colour Contrast Analyser (portable version)
- Lighthouse for Chrome
- NVDA screen reader
- Sa11y

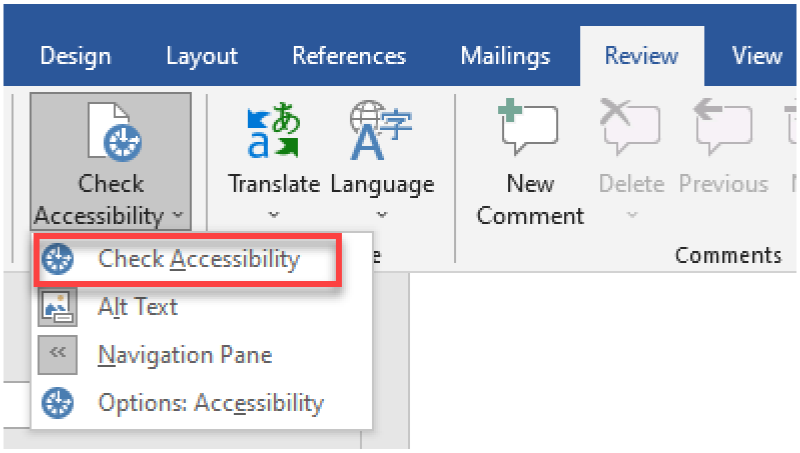
DOCUMENT ACCESSIBILITY CHECKERS

Some document applications include **accessibility** checkers and require specific techniques to make documents accessible.

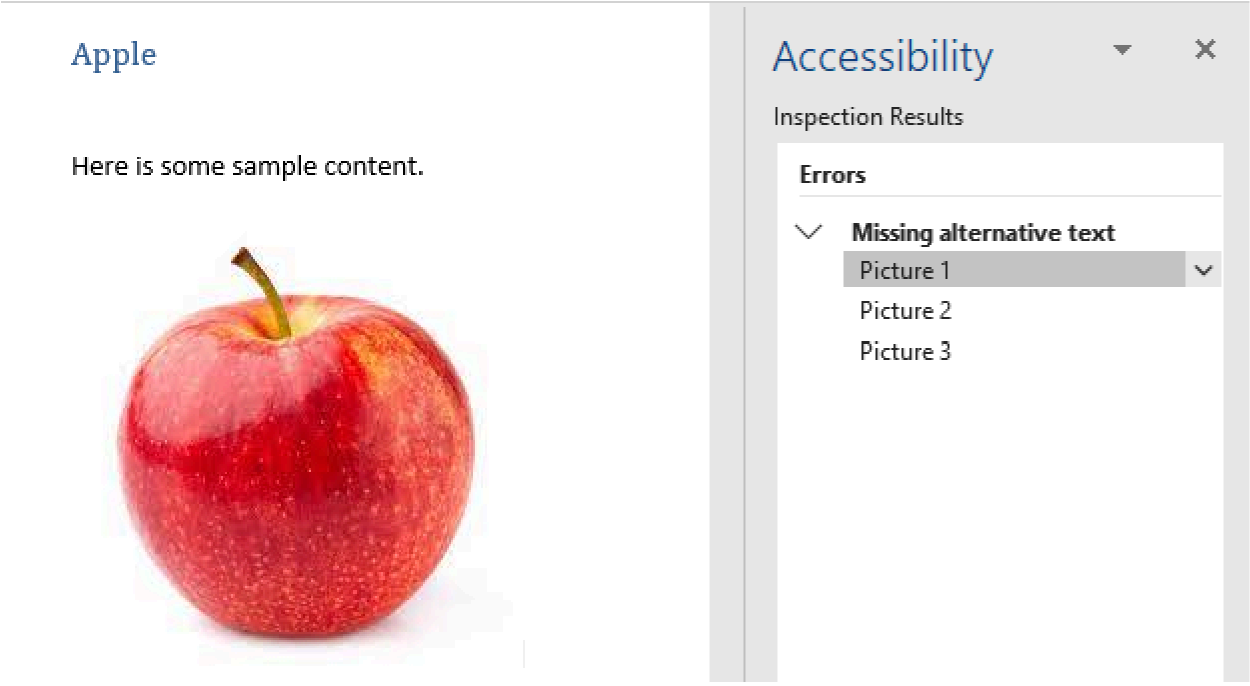
Microsoft Word

To run the accessibility checker in Word, try the following:

1. On the ribbon, select the Review tab.
2. Select Check Accessibility. A dropdown will appear – choose Check Accessibility again. This will run a test for accessibility and the results will pop up.



3. Review your results. A list of errors, warnings, and tips with how-to-fix recommendations will be listed. See Rules for the Accessibility Checker for more information.



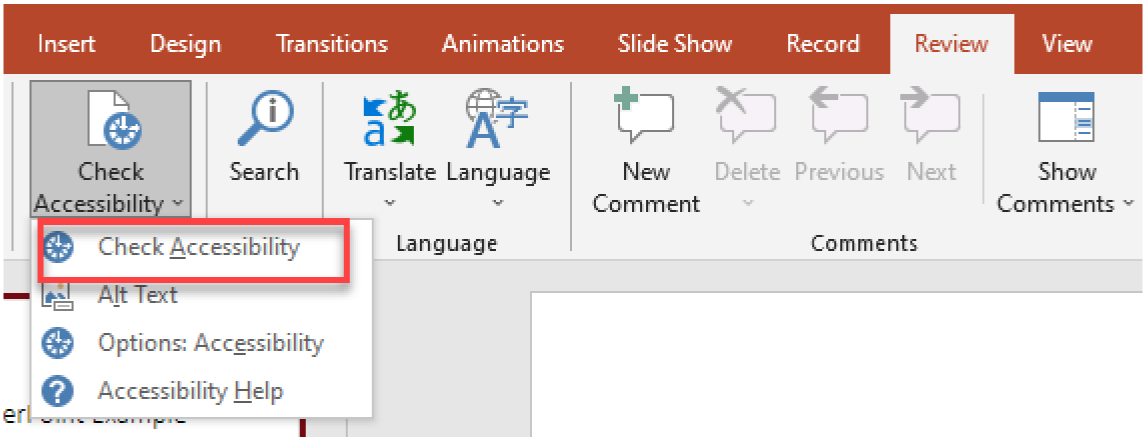
Apple

Here is some sample content.

Microsoft PowerPoint

To run the accessibility checker in PowerPoint, try the following:

1. On the ribbon, select the Review tab.
2. Select Check Accessibility. A dropdown will appear – choose Check Accessibility again. This will run a test for accessibility and the results will pop up.

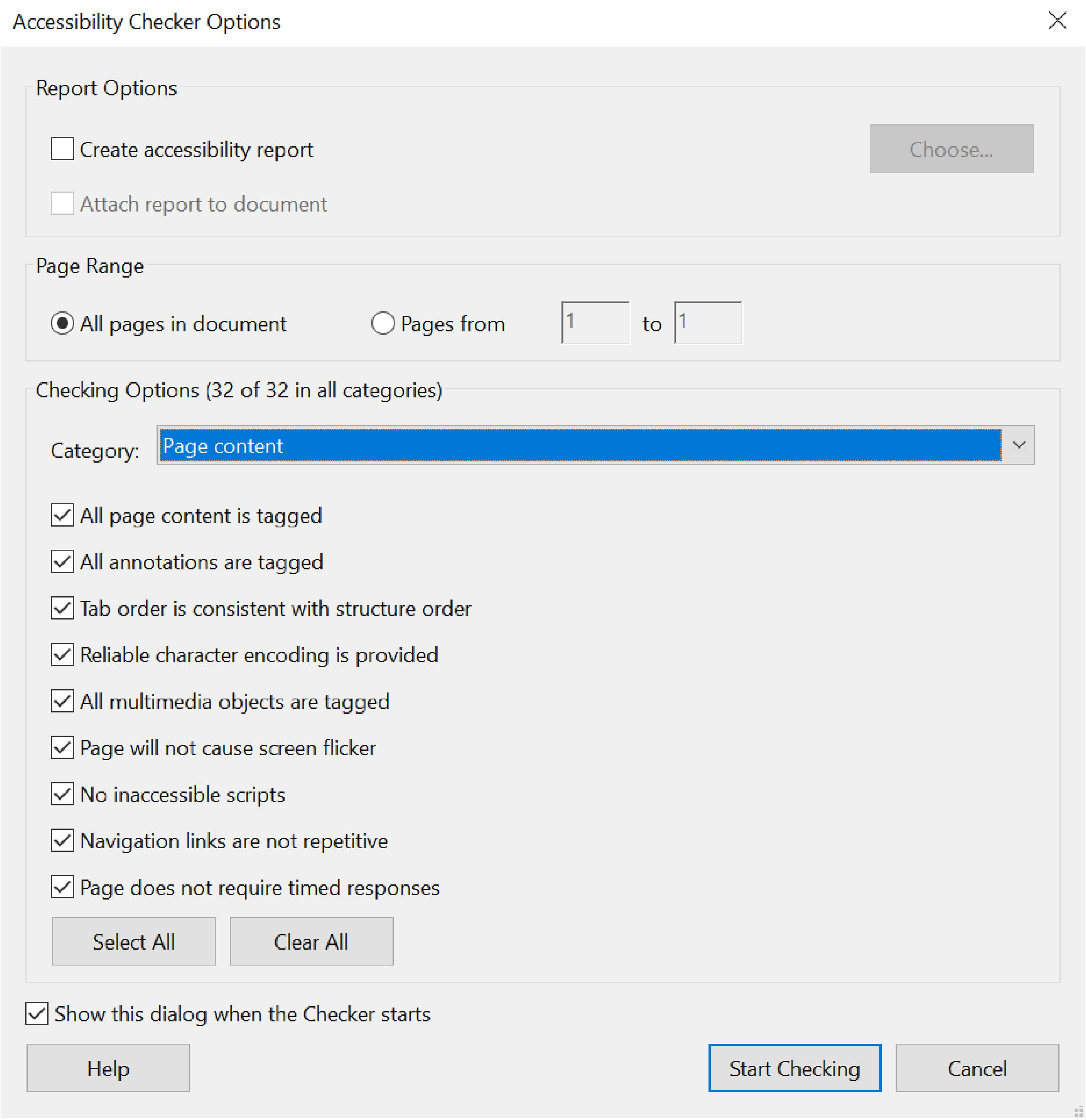


3. Review your results. A list of errors, warnings, and tips with how-to-fix recommendations will be displayed. See Rules for the Accessibility Checker for more information.

Adobe PDF

Only some of the Adobe products have an accessibility checker. Adobe Acrobat **Reader** does not have an accessibility checker, but **Adobe Acrobat Pro** does. To run the accessibility checker in **Acrobat Pro**, try the following:

1. Select Tools > Accessibility.
2. The Accessibility toolset is displayed in the secondary toolbar.
3. In the secondary toolbar, click Full Check/Accessibility Check.
4. The Accessibility Checker Options dialog box is displayed.
5. In the Report Options section, select options for how you want to view the results. You can save the results as an HTML file or attach the results file to the document.
6. Select a page range if you prefer to check individual pages of a document.
   Note: When you have a large document, you can either run a full check/accessibility check or check one page at a time, depending on your work style.
7. Select one or more of the Checking Options. These correspond to different accessibility features you want checked (page content, annotations, tab order).



Accessibility Checker Options

Report Options

□ Create accessibility report   Choose...

□ Attach report to document

Page Range

● All pages in document   ○ Pages from 1 to 1

Checking Options (32 of 32 in all categories)

Category: Page content

☑ All page content is tagged

☑ All annotations are tagged

☑ Tab order is consistent with structure order

☑ Reliable character encoding is provided

☑ All multimedia objects are tagged

☑ Page will not cause screen flicker

☑ No inaccessible scripts

☑ Navigation links are not repetitive

☑ Page does not require timed responses

Select All   Clear All

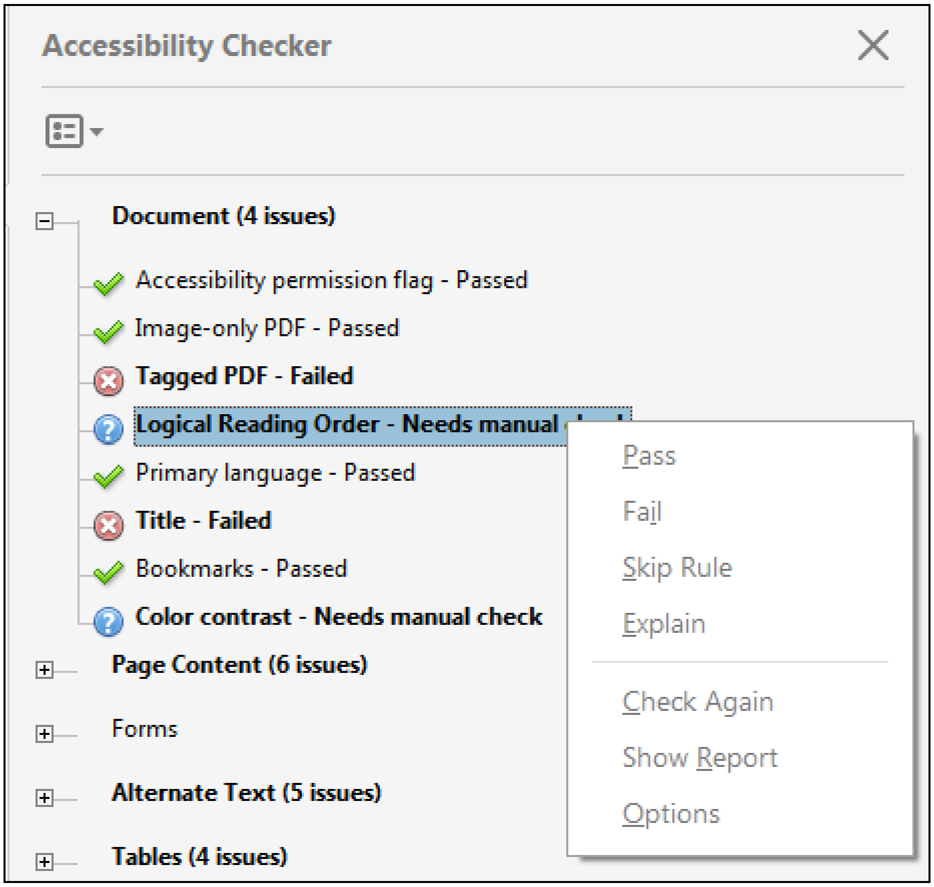
☑ Show this dialog when the Checker starts

Help   Start Checking   Cancel

8. Click Start Checking. The results are displayed in the Accessibility Checker panel on the left, which also has helpful links and hints for repairing issues. If you created a report in step 5, the results are available in the selected folder.
9. Because the Full Check/Accessibility Check feature cannot distinguish between essential and nonessential content types (such as decorative borders), some reported issues don't affect readability. It's a good idea to review all issues to determine which ones require correction.

10. The report displays one of the following statuses for each rule check:

    - **Passed:** The item is accessible.
    - **Skipped By User:** Rule was not checked because it wasn't selected in the **Accessibility Checker Options** dialog box.
    - **Needs Manual Check:** The **Full Check/Accessibility Check** feature couldn't check the item automatically. Verify the item manually.
    - **Failed:** The item didn't pass the accessibility check.



- Document (4 issues)
  - ✓ Accessibility permission flag - Passed
  - ✓ Image-only PDF - Passed
  - ☒ Tagged PDF - Failed
  - ? Logical Reading Order - Needs manual
  - ✓ Primary language - Passed
  - ☒ Title - Failed
  - ✓ Bookmarks - Passed
  - ? Color contrast - Needs manual check
- Page Content (6 issues)
- Forms
- Alternate Text (5 issues)
- Tables (4 issues)

Resources:

- Microsoft Word and PowerPoint:
  - Improve accessibility with the Accessibility Checker
  - Rules for the Accessibility Checker

  - Make your content accessible to everyone with the Accessibility Checker
- Adobe Acrobat Pro:
  - Create and verify PDF accessibility (Acrobat Pro)
- Adobe Acrobat Reader:
  - Accessibility features

PART V

GLOSSARY

Acronyms

A11y

An abbreviation that refers to the eleven letters between the A and the Y in Accessibility. Usually refers to Digital Accessibility.

ACA

Accessible Canada Act

ADHD

Attention deficit hyperactivity disorder

ASD

Autism spectrum disorder

AODA

Accessibility for Ontarians with Disabilities Act

API

Application Program Interface

DevTools

Developer Tools

ETSI

European Telecommunications Standards Institute

FARs

Functional Accessibility Requirements

IASR

Integrated Accessibility Standards Regulation

ICT

Information and communications technology

IT

Information technology

JAWS

Job Access With Speech: A screen reader

NVDA

Non-visual display access: A screen reader

OHRC

Ontario Human Rights Code

OS

Operating System

POUR

An acronym for the four WCAG principles: Perceptive, Operable, Understandable, Robust

SC

Success Criterion

UI

User Interface

W3C

World Wide Web Consortium

WCAG

Web Content Accessibility Guidelines

WAI

Web Accessibility Initiative

Terms

Accessibility

The design of products, devices, services, environments, technologies, policies and rules in a way that allows all people, including people with a variety of disabilities, to access them.

Accommodation

A change that is made so that a person with a disability is able to fully participate or access information.

Adaptive Technology

Special versions of already existing technologies or tools, for example large print books, keyboard modifications, or Operating System (OS) level accessibility tools. The adaptation helps individuals to accomplish specific tasks.

Assistive Technology

Specialized hardware or software that can assist people with disabilities to perceive and interact with digital content.

Barrier

Anything that might hinder people with disabilities' full and equal participation. Barriers can be architectural, technological, attitudinal, based on information or communications, or can be the result of a policy or procedure. Barriers can be financial, knowledge based, or directly related to an individual's disability (e.g., no descriptive alternative text accompanying a photo would be a barrier for someone with a vision impairment).

Conformance / conformant

How well something, such as a product, service or a system, meets a specified standard.

Conversion-ready

Describes information that is in an electronic format that can be easily converted into an accessible format.

Compliance / compliant

The act of following requirements for a product, service or system.

Component

An identifiable part of a larger program or construction.

Disability

Any impairment, or difference in physical, mental, intellectual, cognitive, learning, or communication ability. Disabilities can be permanent, temporary, or episodic (meaning that the impact of the disability can change over time). There are different types of disabilities, including physical, vision-related, hearing-related, and cognitive disabilities. The specifics of a disability vary by person and a person can have more than one disability.

Equality and Equity

Equality is when each individual or group of people is given the same resources or opportunities. Equity is when each individual or group of people is given resources or opportunities that account for imbalance in social systems. Equity aims for equal outcomes.

Inclusivity

The practice or policy of providing equal access to opportunities and resources for people who might otherwise be excluded or marginalized, such as those having physical or mental disabilities or belonging to other minority groups.

Inclusive design

Design that considers the full range of human diversity with respect to ability, language, culture, gender, age and other forms of human difference.

Persona

A fictional user-type created to represent real people that might use your service, product, or site. Creating multiple and diverse personas can help you recognize different needs and expectations of potential users.

Usability

A principle that ensures that a product, service or a system is not only accessible but also easy to use and understand.

Widget

A component added to a website or application as a stand-alone feature.

Assistive and Adaptive Technologies

Screen reader

A technology that helps people who have difficulties seeing to access and interact with digital content, like websites or applications via audio or touch (e.g. Braille). Screen readers include **JAWS**, **NVDA**, and VoiceOver.

Screen magnifier

Software that presents enlarged content, by enlarging part (or all) of a screen.

Switch control

Software that interacts with a digital device using one or more switches instead of the mouse-click or tap on a touchscreen.

Speech-to-Text

Software that converts spoken words to electronic text or commands.

Text-to-speech

Same technology as a Screen Reader. See Screen reader

ADDITIONAL RESOURCES

Specific Audiences

Libraries:

- Ontario Council of University Libraries, Accessibility Information Toolkit for Libraries (May 2014)
- Ontario Council of University Libraries, ROAM: Report on Accessible Media identifying options to ensure equitable access to video collections (2015)

Educators:

- Council of Ontario Universities, Understanding Barriers to Accessibility: An Educator's Perspective (undated)
- Council of Ontario Universities, Teaching Tips (undated)
- Brock University, Creating Accessible Classroom Resources (undated)
- Ryerson University, Tips for Inclusive Teaching (undated)
- University of Guelph, Accessible Instruction – Resources (including an educators' accessibility resource toolkit) (undated)

E-Learning:

- Accessibility in e-Learning: What You Need to Know (July 2014) and its companion document: Easy to Use Resources

Procurement

- Government of Canada, ICT Accessibility Requirements Wizard (prototype) (2022)
- University of Toronto, Accessibility Language for Procurement Agreements (2010)
- Council of Ontario Universities, Accessible Procurement Guide
- Global Initiative for Inclusive ICTs (G3ICT), 9 Steps to Procuring Accessible ICTs for Inclusive Education
- University of Montana, Electronic and Information Technology Accessibility Policy and Procedures
- Government of Canada, Guideline on Making Information Technology Usable by All

- New York University, Procedures and Procurement for Digital Accessibility
- University of Wisconsin – Madison Information Technology, Procure Accessible Technology
- O. Dosis, George Brown College, Ontario Colleges: AODA Procurement Toolkit (2014)

Accessibility in Different Platforms

Documents

- Government of Canada, Digital Accessibility Toolkit, including accessible document guides: Office 365 and Office 2016 (2022) https://a11y.canada.ca/en/guides/
- IDRC, Accessible Digital Office Document Project (2010) https://adod.idrc.ocadu.ca/
- Council of Ontario Universities, Accessible Digital Documents and Websites (undated) https://accessiblecampus.ca/reference-library/accessible-digital-documents-websites/
- McMaster University, Accessible Communications and Accessible Digital Media Resources (undated) https://accessibility.mcmaster.ca/digital-accessibility/
- Ontario Tech University, Guide to Accessible Documents (2021) https://accessibility.ontariotechu.ca/resources/ontariotechguidetoaccessibledocuments.pdf
- George Brown College, Accessible Information and Communications, including numerous accessible document tutorials (undated) https://www.georgebrown.ca/about/aoda/accessibility/accessible-information-communication

Websites

- Trent University, Basic Web Accessibility Guide (undated)
- Council of Ontario Universities, Accessible Digital Documents and Websites (undated)
- Ryerson University, Tips for Creating Accessible Websites (undated)
- Association of Registered Graphic Designers, AccessAbility: A Practical Handbook on Accessible Web Design (2015)

Graphic Design

- Association of Registered Graphic Designers of Ontario, Accessibility: A Practical Guide on Accessible Graphic Design (2010)

Social Media:

- Queen's University, Social Media Accessibility (undated)

E-Learning:

- University of Western Ontario, Accessibility in e-Learning: What You Need to Know (July 2014) and its companion document: Easy to Use Resources

Videos and Digital Media:

- McMaster University, Accessible Communications and Accessible Digital Media Resources (undated)
- Queen's University, Video Accessibility (undated)
- Described and Captioned Media Program, Guidelines and Best Practices for Captioning Educational Video (undated)

Virtual Events and Meetings:

- Ryerson University, Conducting Accessible Virtual Events and Meetings (undated)

Managing an Accessibility Program:

- National Center on Disability and Access to Education, Institutional Tips (undated)
- National Center on Disability and Access to Education, Indicators for Institutional Web Accessibility (undated)

Government of Ontario and Ontario Laws:

- Accessibility Rules for Procurement, (2021)
- Postsecondary Education Standards Development Committee, Development of proposed postsecondary education standards – 2021 initial recommendations report (2021)
- Information and Communications Standards Development Committee, Review of the Information and Communications Standards: 2020 Final Recommendations Report (2020)
- Integrated Accessibility Standards, Accessibility for Ontarians with Disabilities Act

ACCESSIBILITY STATEMENT

eCampusOntario, believes that education should be readily available to everyone, which means supporting the creation of accessible, open, and free educational resources. Wherever possible, the eCampusOntario's Digital Accessibility Toolkit adheres to levels A and AA of the Web Content Accessibility Guidelines (WCAG 2.0, 2.1) of perceivable, operable, understandable, robust, and conformance (https://www.w3.org/TR/WCAG21/).

Pressbooks was chosen for its commitment to built-in accessibility, outlined at https://pressbooks.org/accessibility/.