

Checking Web Accessibility

Ožbej Golob, Alexander Thien, Olli-Pekka Riikola, Robin Karlsson

706.041 Information Architecture and Web Usability 3VU WS 2021/2022
Graz University of Technology

10 Nov 2021

Abstract

Writing a survey can be a traumatic endeavour. It might be a student's first foray into academic research. There are often obstacles and false dawns along the way. This survey paper takes a fresh look at the process and addresses new ways of accomplishing this daunting goal.

The abstract should concisely describe what the survey is about. State the areas which are covered and also those which are not covered. Market your survey to your readership. Also, make sure you mention all relevant keywords in the abstract, since many readers read *only* the abstract and many search engines index *only* the title and the abstract.

This survey explores the issues concerning the writing of an academic survey paper and presents numerous novel insights. Special attention is paid to the use of clear and simple English for an international audience, and advice is given regarding the use of technical aids to production.

© Copyright 2021 by the author(s), except as otherwise noted.

This work is placed under a Creative Commons Attribution 4.0 International (CC BY 4.0) licence.

Contents

| | |
|--|------------|
| Contents | ii |
| List of Figures | iii |
| List of Tables | v |
| List of Listings | vii |
| 1 Introduction | 1 |
| 1.1 Not a Series of Summaries | 1 |
| 1.2 Read All the Papers and Research Some More | 1 |
| 1.3 Dividing up the Field | 1 |
| 1.4 Composing a Title and Abstract | 1 |
| 1.5 Double-Sided Printing. | 2 |
| 1.6 Single Children | 2 |
| 1.7 Make Captions Carry the Story Too | 2 |
| 1.8 Avoid Orphan Floats | 2 |
| 2 Text-Only Browsers | 3 |
| 2.1 tb-history | 3 |
| 2.2 tb-known | 3 |
| 2.2.1 tb-problems | 3 |
| 2.3 tb-browsers. | 4 |
| 2.3.1 tb-lynx | 4 |
| 2.3.2 tb-webbie | 4 |
| 2.3.3 tb-w3m | 4 |
| 2.3.4 tb-links | 4 |
| 2.3.5 tb-others | 4 |
| 2.4 tb-general | 4 |
| 2.4.1 tb-showcase | 6 |
| 2.4.2 tb-screenshots | 6 |
| 3 Screen Readers | 7 |
| 3.1 JAWS. | 7 |
| 3.2 NVDA | 7 |
| 3.3 VoiceOver | 7 |
| 3.4 Narrator | 9 |

| | | |
|----------|--|-----------|
| 3.5 | TalkBack | 9 |
| 3.6 | Showcase videos. | 9 |
| 3.7 | Screen Readers Conclusion | 9 |
| 4 | Browser Extensions for Accessibility Auditing | 11 |
| 4.1 | axe DevTools | 11 |
| 4.2 | Accessibility Insights for Web | 11 |
| 4.3 | Google Lighthouse | 12 |
| 4.4 | Siteimprove Accessibility Checker | 12 |
| 4.5 | WAVE Evaluation Tool | 12 |
| 4.6 | Showcase videos. | 12 |
| 4.7 | Browser Extensions Conclusion | 13 |
| 5 | Web Tools For Accessibility Audits | 15 |
| 5.1 | Accessi | 16 |
| 5.2 | WAVE | 17 |
| 5.3 | Page Speed Insights aka. Lighthouse | 18 |
| 5.4 | Showcase videos. | 18 |
| 5.5 | Web Tools For Accesibility Audits Conclusion | 19 |
| 6 | Concluding Remarks | 21 |
| | Bibliography | 23 |

List of Figures

| | | |
|-----|---|----|
| 2.1 | Browsh grahical CLI browser | 5 |
| 2.2 | Firefox Reader-View | 5 |
| 3.1 | Primary screen reader | 8 |
| 3.2 | Historical trends for primary screen reader | 8 |
| 5.1 | Accessi Overview | 16 |
| 5.2 | Accessi Overview | 17 |
| 5.3 | Page Speed Insights Overview | 18 |

List of Tables

| | | |
|-----|--|----|
| 3.1 | Screen Readers Information | 9 |
| 4.1 | Browser Extensions Information | 12 |

List of Listings

Chapter 1

Introduction

An academic survey paper presents a survey or overview of the state of the art in a particular field. Every chapter and every section should have some introductory text at the beginning, like this text. Never jump straight in to the first section or subsection without one or more paragraphs of introductory text.

1.1 Not a Series of Summaries

A survey is *not* simply a series of summaries of papers. If I have given you say 8 papers to start you off, what you should *not* do is: divide up the papers (read two each) and produce a series of 8 unconnected paper summaries.

1.2 Read All the Papers and Research Some More

Each of you should read *all* the papers and resources: both those I gave you and those you found yourselves. Make sure you search for more papers and resources yourselves. Not just a Google search. Search the ACM [ACM 2021] and IEEE [IEEE 2021] digital libraries. You may want to use Mendeley to collect your resources or maybe maintain a (shared) .bib file within a Git repository.

Include a list of *all* the relevant papers and resources you have found and mark those you have chosen to focus on. Make sure *all* the papers and resources you found or were given appear in the bibliography.

1.3 Dividing up the Field

The hardest part of any survey is dividing up the field. Look for common concepts and threads in the papers and resources. Do they report similar or dissimilar results? Does one paper or resource support or contradict another?

Once you have all read all the papers: you need to construct a small hierarchy (taxonomy) to classify the concepts appearing in the papers and resources. Structure your survey into chapters and sections based on your taxonomy.

1.4 Composing a Title and Abstract

One useful strategy for composing a good title and abstract involves brainstorming for a list of keywords. Start by writing down a list of all the words and phrases describing important topics covered in the thesis and which potential interested readers might use as search terms to find the thesis. Then construct a title containing the most important of these keywords. Finally, compose the abstract and make sure most of the rest of the keywords are contained somewhere in the abstract. Search engines and library systems will usually index the title and the abstract, so anyone searching for any of the keywords should now be

able to find the thesis. When the thesis is approaching completion, revisit the title and abstract, an extra extra keywords and make any necessary adaptations.

1.5 Double-Sided Printing

Create and print your survey in colour and for two-sided (duplex) printing. Modern laser printers can easily handle printing out in colour and double-sided. A survey paper printed one-sided will be (unnecessarily) twice as thick and twice as heavy.

Sections, including the bibliography and any appendices, should usually (as far as possible) start on a new right-hand (odd-numbered) page. This is what the `\cleardoublepage` command does.

1.6 Single Children

As in real life, a single child is not a good idea. A chapter with only one section makes no sense. A section with only one subsection makes no sense. A subsection with only one subsubsection makes no sense either. If a structural unit has subunits, then there should always be at least two subunits.

1.7 Make Captions Carry the Story Too

Some readers like to scan through your work from figure to figure, gaining an impression of what it is about by reading the captions. Support these readers by:

- *Writing self-contained captions:* the caption should describe the figure or table as completely as possible, without assuming knowledge of material in the running text.
- *Writing longish captions:* it is fine for captions to contain two or three sentences.
- *Stringing captions together:* Reading successive captions should also tell an abridged version of the entire story.

1.8 Avoid Orphan Floats

Every floating element (figure, table, or listing) which appears in the thesis and is given its own number such as Figure 3.1, Table 4.1, or Listing 5.1 *must* be discussed and referenced somewhere in the running text. An orphan float is a float which appears and has a number, but is never referenced in the flowing text.

Chapter 2

Text-Only Browsers

2.1 A Quick Look to the History of Text-Only Browsers

Originally, text browsers were the only way to browse the web. Also, back in the days, internet connection speeds used to be much slower than today but similarly web sites were much simpler and it was possible to access them with even with the narrow bandwidth. After that there is now several powerful graphical web browsers for web users, but the text browsers still have some important usecases even they are quite old solution for web browsing. [Memon 2022]. In addition, there is still a few promising text browser remaining and even receiving recent updates at least annually, and they are worth for have a look.

2.2 Text-Only Browsers in use

The header above might seems to be a bit exaggerating, but it is totally possible and sensible to use text browsers in daily basis. This is, because text browsers are lightweight softwares and still efficient on poor internet connection and they could actually help user to reduce distraction while browsing the web and that is why they could be a good choice for a browser even nowadays. The text browsers are designed to logically organize the content of the website rather than just to be ordinary web browser which gives the access to the site for a user. More than that, persons with no sight or who are partially impaired can more efficiently access website with text browser in combination with screen reader. And when talking about blind persons as a web users it is important to consider, that they obviously can't see or otherwise access the actual graphical pieces of art, images, videos or other similar mediaformats in the website. [King 2004a, Chapter 2] So, text browsers reduces the content that screen readers cannot read and that way they will make the web browsing more convenient for blind people.

2.2.1 Common Problems

Event the layout of the site would be simple in text browser, it still is not simple in practice. According King [2004a, Chapter 5] the reason is presentation. People with sight can have advantage of very versatile navigation link list, but for a blind user it might be a struggle. It becomes a struggle at the point, when text browser try to present navigation bar as a list of links, and screen reader starts to read it - link by link, one by one. Luckily, there is an option to skip this navigation bar and go straight to the content, but only if the element is designed properly in the website. Otherwise text browser does not realize that there is a navigation bar, but it just present it like a list of links and user needs to handle it.

Another issue is empty attributes of element, like ALT attribute in IMAGE element. And of course this problem would quite easily be fixed by the programmer, specially with powerful accessibility checking tools, but sadly it is not always like that. So, since images are not relevant content for a person with no sight, it is very important that the ALT tag will tell, what is going on in that part of the website. Also, if image is for hyperlink, then it should also tell with ALT tag, where it will bring the user. [King 2004a, Chapter 5].

Perhaps, the most difficult issue in the terms of text browser, is dynamic content. Since most text browsers do not support JavaScript, some functionality is not available for the user in the site. This essentially means, that text browsers are not any more so convenient way to browse the web. But still, like mentioned in ??, there is few interesting text browser softwares available in the internet.

2.3 Known Text-Only Browsers

2.3.1 Lynx

Lynx is quite old text browser and it appears to be probably most well known one. But as it is in the Lynx [no date] website, Lynx is a command line interface WWW client for Unix systems. Interesting sidenote could also be, that back in the years the Lynx has been a solution to build Campus Wide Information Systems.

2.3.2 WebbIE

WebbIE is the one, which is designed to help visually impaired persons to access the web. They are dedicated to work in combination with screenreaders, like JAWS, WindowEyes, Thunder, NVDA and Narrator as mentioned in the King [2004b] website. WebbIE runs on Windows, and it seems to be pretty much only choice for Windows users.

2.3.3 W3M

W3M is also a very handy text browser and it works as pager too, like 'more' or 'less' as it says in W3M [no date] website. It also works as text formatting tool from HTML to plain text.

2.3.4 Links

Another interesting and very nicely working text browser is Links which is school project from the days in 1999. But since then it has been kept up to date and many people are involved in the project and put effort to make Links more diverse. [W3M 2002]. It runs on Unix.

2.3.5 Others

There is also few cases worth to mention, perhaps even if they are not in the same class with these ones mentioned above. However, there is a project called Browsh, which is a graphical text browser running on Linux. It supports JavaScript, so all functionality in the site should be accessed but it is still very early stage in the development. But it might be useful on poor internet connection.

Another ones are more like plugins, but it is good to know, that there is still such. Chrome offers a plugin, which makes website like text site, but it seems to be somehow outdated and poor. More interesting and very well working built-in function is Firefox Reader-View. It just makes the website to be easier to read with removing distracting elements and making the content to be more clearly seen. This function can be activated just by pressing F9 in the Firefox browser and it works in several website.

2.4 Showcases and Data of Text Browsers

In this section there is provided practical examples and an interesting data of these text browsers. Due to simplicity of the text browsers, it is easy to get familiar with them to only watching a showcase video and comparing the screenshots below in the section ??.

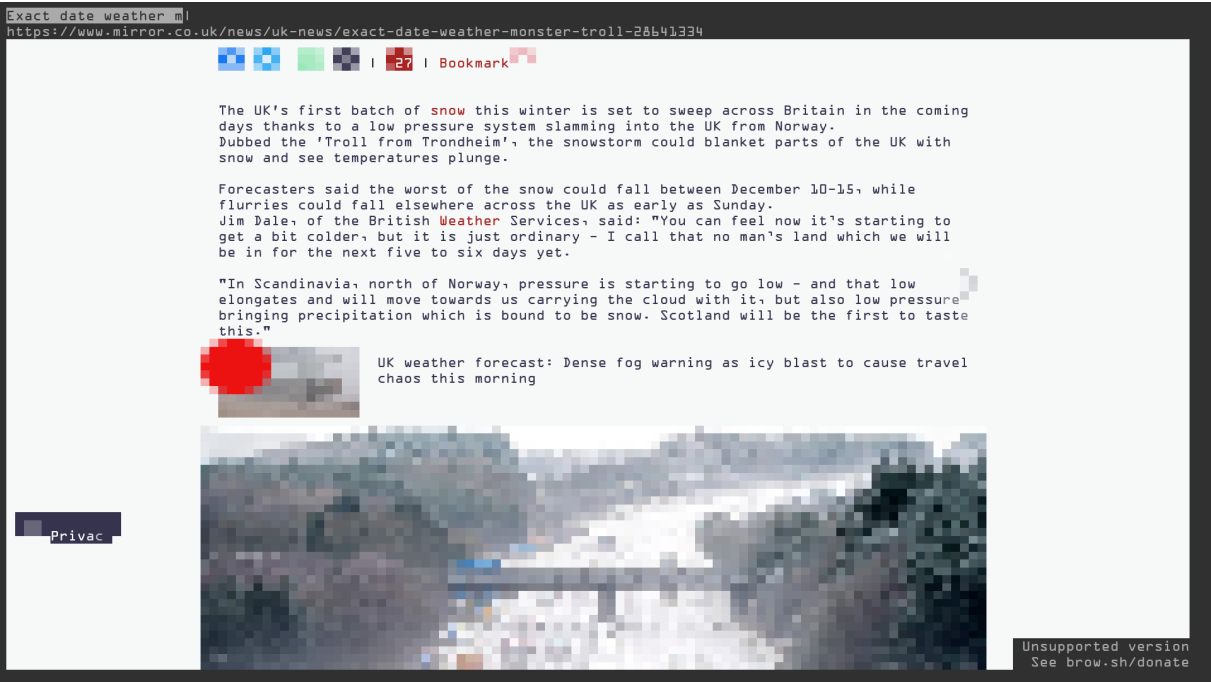


Figure 2.1: Browsing a web with Browsh in website gov.uk.

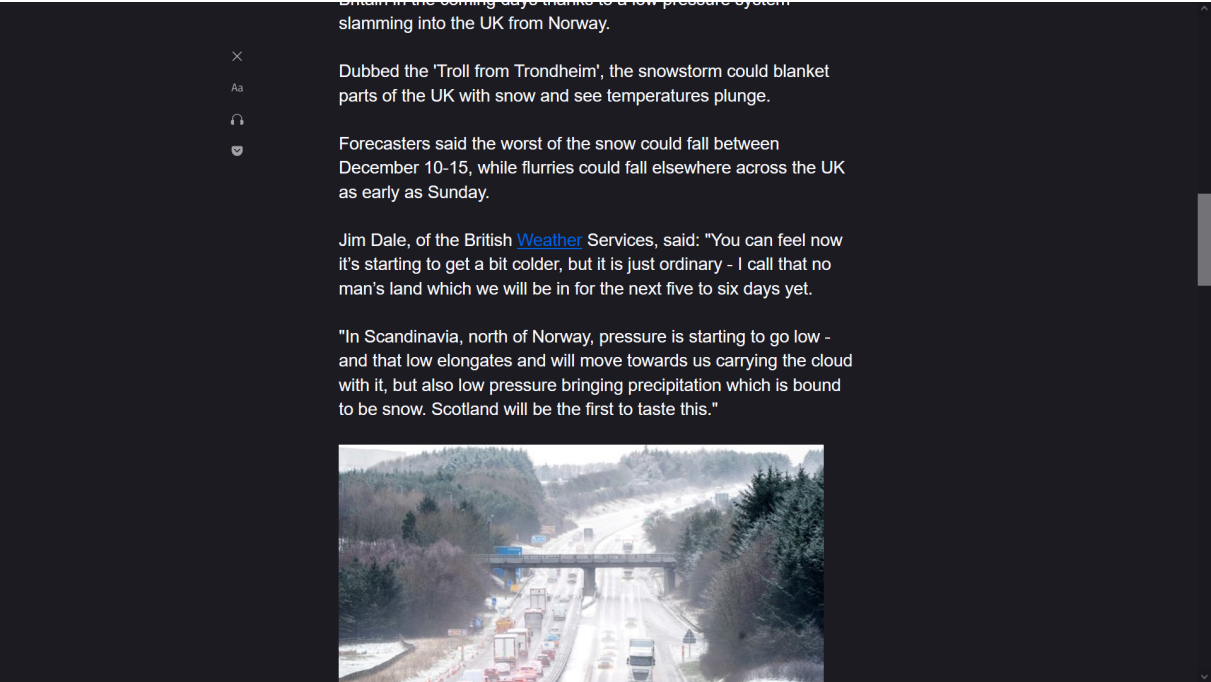


Figure 2.2: Browsing a web with Firefox Reader-View

2.4.1

There is showcase video in YouTube, where is shown what it looks like when browsing the web with the Lynx text browser. In the video it is shown two examples gov.uk and mirror.co.uk, which represents an examples of good and bad web accessibility. Both videos are quite similar in structure, just tabbing through the sites and entering the links, but that way it is easy to recognize the probles with the first glance in the mirror.co.uk website. Link to the showcase video is provided in reference Riikola [no date].

2.4.2 Screenshots of Text Browsers

Following images presents how efficient a text browsers actually could be. In the case of mirror.co.uk, even if there is plenty of images and advertisements, the text browsers are able to present the content somehow sensible. In this area, most interesting examples is Lynx and WebbIE. Others are more or less similar than the Lynx - only thing to consider is, that W3M and Links works responsively inside of CLI, Lynx does not.

Chapter 3

Screen Readers

Screen readers are software applications, primarily used by visually impaired people. Screen readers convert web content (text, buttons, images, and other elements) into speech or braille output. Screen readers attempt to convey what visually non-impaired people see on a display via non-visual means like text-to-speech, sound icons, or a braille device.

In May - June 2021, WebAIM surveyed the preferences of screen reader users. They received 1568 valid responses. Figure 3.1 shows the primary screen readers preferences. The majority of users use JAWS and NVDA screen readers as their primary screen readers. Figure 3.2 shows historical trends for primary screen reader usage. After a decade of decreases in primary usage, JAWS is once again the most used screen reader, with NVDA and VoiceOver both decreasing in primary usage over the last two years.

In this paper, we focus on JAWS, NVDA, VoiceOver, Narrator, and TalkBack screen readers. They are further described in the sections below.

3.1 JAWS

JAWS is the most popular screen reader for Windows computers, however, it has a learning curve. JAWS has a lot of shortcuts and hotkeys that a user has to get used to in order to operate JAWS efficiently. JAWS has the most configurable options among the reviewed screen readers. JAWS is only available as paid software, either as an enterprise or single-use package (90 EUR a year or approximately 900 EUR as a one-time purchase). JAWS demands a lot of the computer's RAM and can occasionally slow the computer down. JAWS supports 10 languages.

3.2 NVDA

NVDA is the second most popular screen reader for Windows computers. NVDA is available as free and open-source software. NVDA is a good free alternative to JAWS, however, it is not as configurable as JAWS. NVDA has a lot of shortcuts and hotkeys that a user has to get used to in order to operate NDVA efficiently. NVDA supports 63 languages.

3.3 VoiceOver

VoiceOver is free and comes included with all Apple products. VoiceOver requires no installation or setup. VoiceOver is user-friendly and configurable. The learning curve takes some effort since VoiceOver is not operated the same way as typical mobile phone users are used to. Voice is operated by multiple special gestures (for example double finger drag, triple finger drag). VoiceOver supports 64 languages.

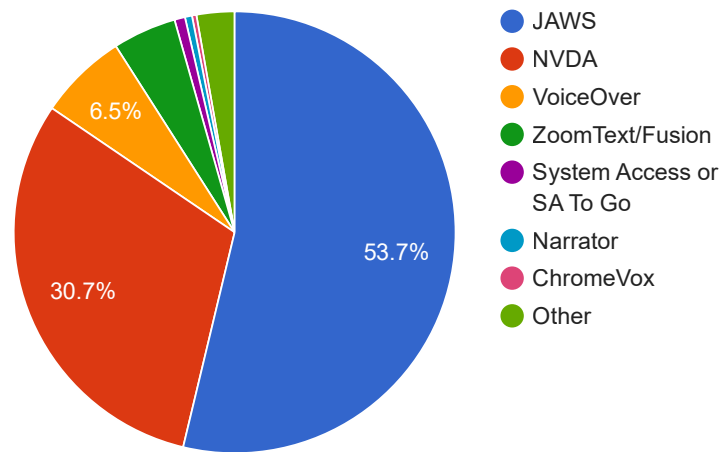


Figure 3.1: Primary screen reader. [Image extracted from WebAIM. ©2022 WebAIM.]

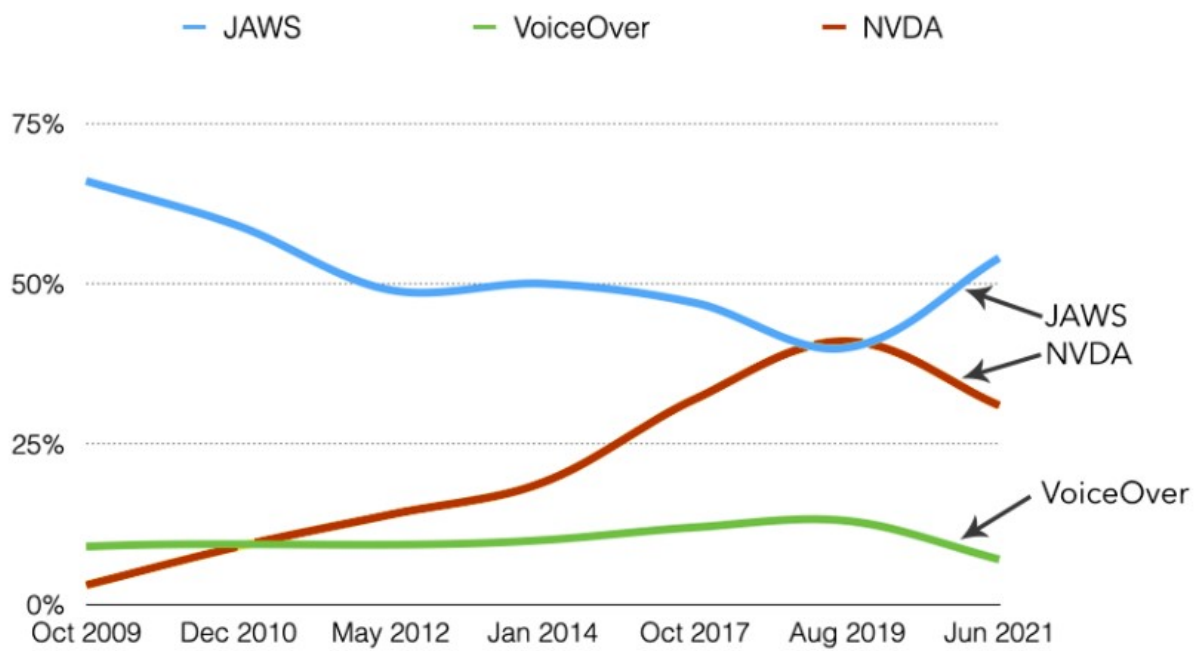


Figure 3.2: Historical trends for the primary screen reader. [Image extracted from WebAIM. ©2022 WebAIM.]

| Screen reader | Last update | System | Licence | No. of lang. |
|---------------|--------------|------------|----------------------|--------------|
| JAWS | 25. 10. 2022 | Windows | Commercial | 10 |
| NVDA | 1. 3. 2022 | Windows | Free and open-source | 63 |
| VoiceOver | 24. 10. 2022 | iOS, macOS | Free | 64 |
| Narrator | 2020 | Windows | Free | 49 |
| TalkBack | 27. 10. 2022 | Android | Free and open-source | 63 |

Table 3.1: Screen readers information

3.4 Narrator

Narrator is free and comes included with the Windows operating system. Narrator requires no installation or setup. Narrator has some shortcuts and hotkeys, but not as many as JAWS and NVDA. Narrator offers limited functionality with web browsers and web apps, especially with navigation and deeper levels of the Windows operating system. Narrator supports 49 languages.

3.5 TalkBack

TalkBack is free and open-source. TalkBack comes included with the Android operating system. TalkBack requires no installation or setup. TalkBack is user-friendly, however, the learning curve takes some effort, since TalkBack is not operated the same way as typical mobile phone users are used to. TalkBack is operated by multiple special gestures (for example swipe left then right). TalkBack supports 63 languages.

3.6 Showcase videos

We recorded showcase videos for each screen reader. We used the screen readers on <https://www.gov.uk/>, which is a good example of web accessibility site, and on <https://www.mirror.co.uk/> site, which is a bad example of web accessibility site. Showcase videos for JAWS ([Ožbej Golob 2022a], [Ožbej Golob 2022b]), NVDA ([Ožbej Golob 2022e], [Ožbej Golob 2022f]), VoiceOver ([Ožbej Golob 2022i], [Ožbej Golob 2022j]), Narrator ([Ožbej Golob 2022c], [Ožbej Golob 2022d]), and TalkBack ([Ožbej Golob 2022g], [Ožbej Golob 2022h]) are available online.

3.7 Screen Readers Conclusion

A comparison of information can be seen in Table 3.1. All screen readers in question, except Narrator, are maintained regularly. JAWS, NVDA, and Narrator are available on Windows, VoiceOver is available on iOS and macOS, and TalkBack is available on Android. JAWS supports 10 languages while other screen readers support between 49 and 64 languages. JAWS is the only screen reader available as paid software while other screen readers are available as free software (NVDA and TalkBack are also open-source). Windows users have 3 options: JAWS, NVDA, and Narrator. The most popular paid option for Windows is JAWS, and the most popular free option for Windows is NVDA. iOS and macOS users can use VoiceOver, and Android users can use TalkBack.

Chapter 4

Browser Extensions for Accessibility Auditing

Using browser extensions is one way to audit the accessibility of websites. This paper evaluates five different extensions for accessibility auditing: axe DevTools, Accessibility Insights for Web, Google Lighthouse, Siteimprove and WAVE. Three of the tools we have selected for this paper use the *axe-core* library, which is an open-source accessibility engine for automated web UI testing. As a result, many of the tools will give similar results and the main difference between the extensions is how the list of accessibility issues is presented and what additional features are offered. The library allows accessibility auditing to Web Content Accessibility Guidelines (WCAG) 2.0 and 2.1 on the levels A and AA.

Most of the browser extensions evaluated in this paper are completely free. Only one of the tools evaluated in this paper offers a paid version of the extension with additional features.

4.1 axe DevTools

Axe DevTools is a tool for auditing accessibility developed by Deque Systems Inc. The browser extension is based on the *axe-core* underlying library for auditing the accessibility, but also offers some additional stricter audit rules compared to the base *axe-core* library, allowing validation according to WCAG 2.1 AAA. The extension is available for Chrome and Edge. The Deque website contains a link to the axe DevTools extension for Firefox, but at the time of writing this paper it does not seem to work.

There are three different plans available, Free, Pro and Enterprise. The free version has a limited set of features and mostly consists of only the automated testing of accessibility. The Pro version costs \$40 per month and offers additional features, such as guided manual tests, partial accessibility testing and exporting of accessibility issues. Lastly, the Enterprise version of axe DevTools offers even more features, for example CI/CD integration and custom rules.

4.2 Accessibility Insights for Web

Accessibility Insights for Web is a browser extension for accessibility auditing using Chrome or Edge. The extension offers two different audit methods, automated checking, known as FastPass, and manual assessment. The FastPass automated check provides accessibility auditing using the *axe-core* library. The accessibility issues are then visible in a list and can be exported as an HTML report or directly to GitHub Issues or Azure Boards for easy integration in the development process. Another feature available in the extension is the visualization of accessibility issues directly on the web page. This allows scrolling the web page to see where accessibility issues occur. The manual assessment in the browser extension consists of extensive step-by-step instructions for auditing the accessibility manually. Of the browser extensions evaluated in this paper, Accessibility Insights for Web is the only free tool to offer support for manual testing.

| Extension | Browser | Licence | Downloads ¹ |
|-----------------------------------|------------------------------|-------------------|------------------------|
| axe DevTools | Chrome, Edge | Free, commercial | 210000 |
| Accessibility Insights for Web | Chrome, Edge | Free, open-source | 120000 |
| Google Lighthouse | Chrome (built-in) | Free, open-source | 900000 ² |
| Siteimprove Accessibility Checker | Chrome, Firefox, Edge, Opera | Free | 30000 |
| WAVE Evaluation Tool | Chrome, Firefox, Edge | Free | 520000 |

Table 4.1: Browser extensions information

¹Data collected from Chrome Web Store, Opera Addons, Firefox Add-ons and Microsoft Edge Add-ons.

²Google Lighthouse is built-in into Google Chrome, so the actual number of users may be significantly higher.

4.3 Google Lighthouse

Google Lighthouse is the most popular tool for auditing accessibility, which is included in Google Chrome. Lighthouse, like the previous browser extensions, is also based on the *axe-core* library. For auditing accessibility, Lighthouse offers a relatively basic set of features, and relies heavily on integration with Chrome DevTools and external references for accessibility issue descriptions. Accessibility issues are presented as a list of what accessibility checks have passed and which ones have failed. The user can then see descriptions about the accessibility issues and see where in the HTML code they occur. Reports of the accessibility audit can be saved as JSON.

4.4 Siteimprove Accessibility Checker

Siteimprove Accessibility Checker is the least used accessibility audit browser extension out of the extensions in this survey. The extension presents the accessibility issues, where you can expand each issue to read the description of it and highlight the element on the web page. Some features that the extension offers, that the other extensions do not offer is simulation of color blindness and excellent filtering of accessibility issues. The list of issues can be filtered by difficulty, role, WCAG level and HTML element type. However, one feature that is not offered is the possibility of exporting the audit report.

4.5 WAVE Evaluation Tool

The WAVE evaluation tool is another extension for auditing accessibility, supporting Chrome, Firefox and Edge. The extension functions somewhat differently than the other extensions, as the tool is embedded directly on the web page. It then inserts icons for elements that fail or succeed in the accessibility checks. The user can then scroll the web page and click the icons to learn more about the accessibility issue.

4.6 Showcase videos

For each of the browser extension, we have recorded videos to showcase the extension and their features. Using each extension, we performed an accessibility audit of <http://www.gov.uk/> <http://www.mirror.co.uk/>. The showcase videos for axe DevTools [Robin Karlsson 2022a], Accessibility Insights for Web [Robin Karlsson 2022b], Google Lighthouse [Robin Karlsson 2022c], Siteimprove Accessibility Checker [Robin Karlsson 2022d], WAVE Evaluation tool are available online [Robin Karlsson 2022e].

4.7 Browser Extensions Conclusion

TODO: add conclusion

Chapter 5

Web Tools For Accessibility Audits

Web tools for accessibility audits provide core auditing functionality to anyone interested. The amount of tools in this category is large, and most of these tools come to the same conclusions for an audited web page regardless of the level of detail. This is attributed to several factors, one being that many of these web tools are using the same core library to assess the accessibility of a given website. In the case of this survey, the used library is called axe-core, which is provided by the Deque company.

In this survey, the tools that are inspected closely are but a fraction of the tools that are currently available on the web. The chosen tools were selected in a way, which enables a closer look at tools that run with the axe core library and without. Additionally, the selection also tries to show the different qualities of each choice and what can be expected when someone tries to use these tools online.

In this survey, our selection of web tools for accessibility audits consists of the following: Accessi, WAVE, and Page Speed Insights which is the web tool equivalent to the Lighthouse browser extension. All of the mentioned web tools will be discussed in further detail in the following sections. For each of these three tools, there were also videos created, which demonstrates the features and the functionality of each of the tools. The videos also give a quick overview of the metrics that were used to determine the accessibility of the tested websites.

5.1 Accessi

Accessi [Accessi 2022] is one of the websites that were used to assess the accessibility of the two test pages. Accessi is a website that offers a free accessibility test to its users. It works according to the Web Content Accessibility Guidelines (WCAG) standard. This can be further refined while using the tool by a choice of the 2.0 version of this standard or its 2018 renewed 2.1 version. The website will analyze the target test website and give a basic rating between 0-100% telling the user the state of compliance of the test site. Through an automated test, further metrics are detected, one example being a statistical analysis of the found issues of the web page. Accessi ranks these issues in three categories: High impact, medium impact, and low impact. The high-impact issues are the most obvious and intrusive ones, descending in severity, followed by medium and low. All the issues that are detected by the automated test run will additionally be listed, described, and enhanced with examples. The feature that makes Accessi stand out from its competitors is the ability to export all of the aforementioned findings in PDF or CSV format, giving the user a form of a To-Do list or a fixed statement that they could work on if they would be looking towards improving the accessibility of their web site. An example of the graphical interface of Accessi can be seen in Figure 5.1 below.

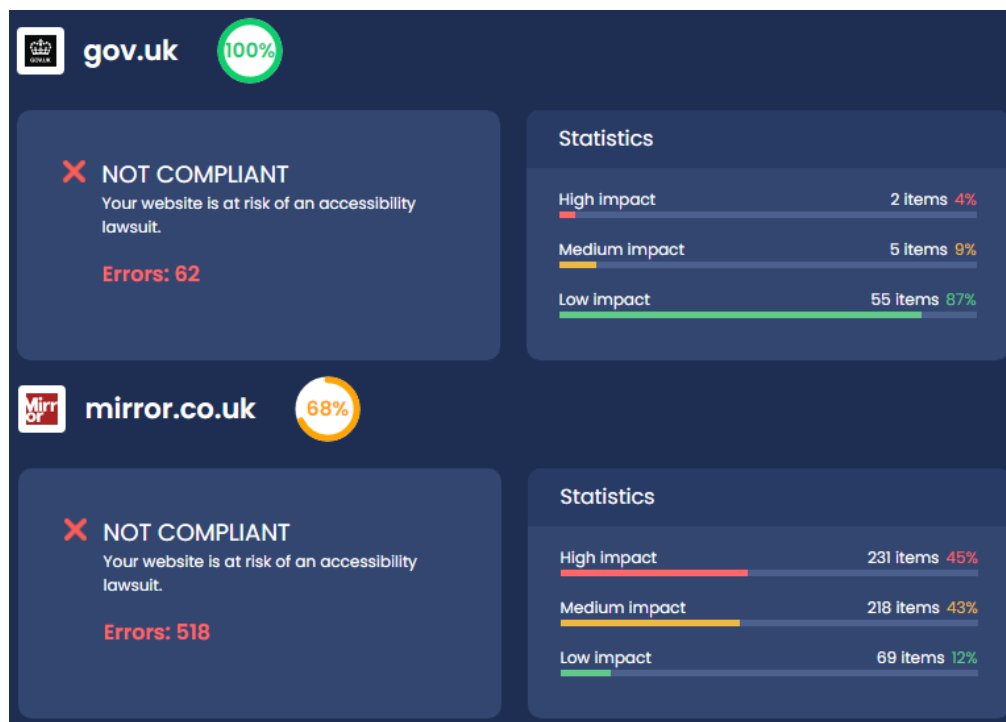


Figure 5.1: Accessi web tool header showing an overview for each test site.

5.2 WAVE

WAVE [Web AIM 2022] is a tool provided by the company WebAim. As the tools described in this survey its purpose is to assess the accessibility of a website. WAVE offers a different take on the user interaction with a website which is subject to an accessibility test. The user is offered an interactive web tool, which embeds itself in the test website which is being surveyed. This embedded interface gives the user the ability to interactively inspect all the errors and warnings that are produced for a given website. The user is given the ability to click on icons which are attached to elements of the test site, clearly outlining which element is considered non-compliant with accessibility guidelines. This interactivity is a special feature of the WAVE tool as through the course of research in our survey it is still the only tool providing these options. The interactive nature of the tool really provides the user with an indepth look into each error and where on the website it is located. In addition to this feature, WAVE also gives a summary over all the different kinds of errors or issues find on a given website. This detailed summary is linked to the icons that are embedded in the website that si being surveyed and will outline and focus the element from the list on the website if clicked on.

As can be seen in the Figure 5.2 below this web tool does not necessarily work with every website, as due to some scripting errors the tool might fail to integrate itself properly into some websites.

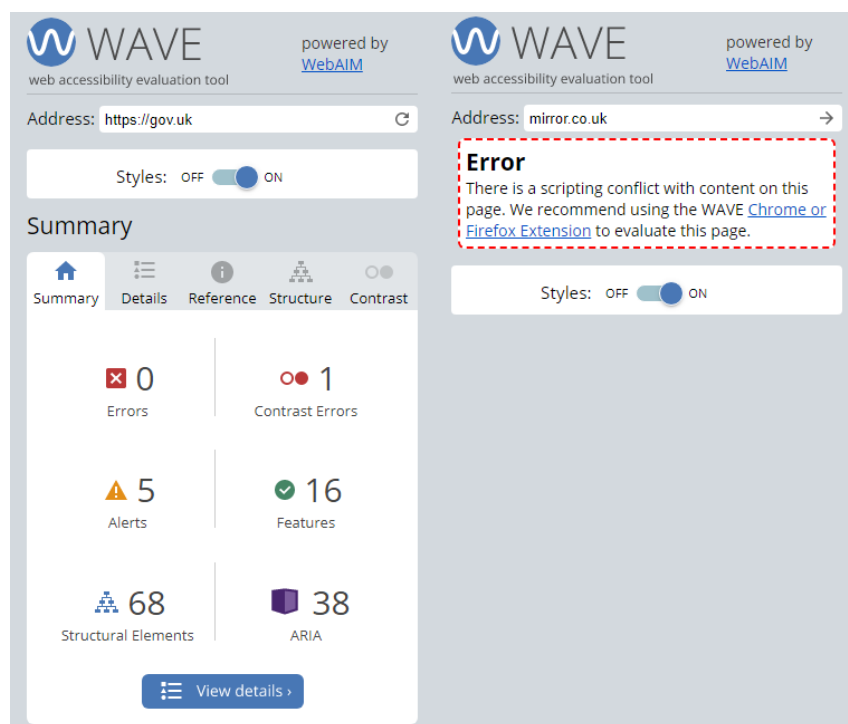


Figure 5.2: WAVE web tool header showing an overview for each test site.

5.3 Page Speed Insights aka. Lighthouse

Page Speed Insights [Google 2022] is a web tool based on the popular Lighthouse browser extension. Using the axe-core library as mentioned in Chapter 4.1 this tool provides the same functionality as its browser extension counterpart. The tool can evaluate not only in consideration to accessibility but other metrics relevant to website quality. The web tool offers insight into basic performance, best practices, and Search Engine Optimization (SEO), in addition to assessing the accessibility of the tested website. After the accessibility audit is done the user may interact with the tool for further information on the found issues. These issues are represented in a list format, explaining what the current error is, as well as suggestions on how to fix said errors and examples of what they might look like. This list view can be seen in the given example in Figure 5.3 below. Each of the elements in the list can be clicked on to view a more detailed version of the aforementioned errors and issues.

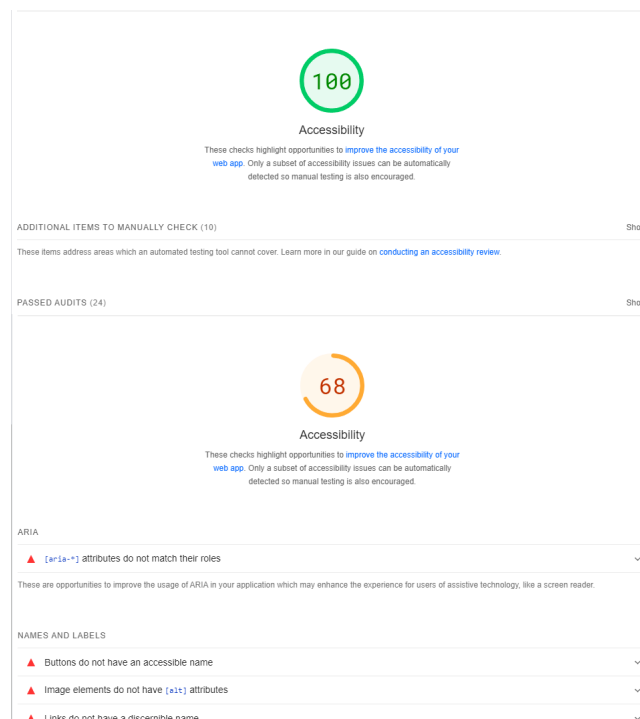


Figure 5.3: Page Speed Insights (Lighthouse) web tool showing an overview for each test site.

5.4 Showcase videos

In this survey, two websites were targeted for evaluation by the aforementioned tools. For a good example concerning web accessibility, the <https://www.gov.uk/> website was chosen whereas for the negative example <https://www.mirror.co.uk/> was the choice. The videos that were produced each showcase the interaction between the user, the tools, and the website that is being evaluated respectively. A video of this format was produced for Accessi [Alexander Thien 2022a], WAVE [Alexander Thien 2022c] and Page Speed Insights (Lighthouse) [Alexander Thien 2022b]. All of these videos showcase the features of the tool that is being covered as well as highlight a few extras the tool of each video might have over other tools that might have a different focus. The outcome of all of the videos is yet the same, as they all evaluate the positive example as being very well rated and rating the negative example as being non-compliant with the usual standards of accessibility for websites.

5.5 Web Tools For Accessibility Audits Conclusion

Concluding this chapter of the survey, it is fair to say that the selection of web tools for accessibility audits is plentiful. There are many tools which base themselves on the axe-core library and many others which use their own libraries to assess the accessibility of a website. As there is a standard in place which sets the boundaries for accessibility, usually all the tools come to the same general conclusion for a given website, evaluating a good website as compliant to accessibility and bad ones as being non-compliant. Through the big selection of available tools, it is advisable to choose the one which suits the needs of the auditor the most. An example would be when maintaining a website on their own, the auditor might choose to use either WAVE or Accessi, depending on if they prefer a list type of view or the more interactive way. When looking for a more general overview including other metrics like performance, it is probably safe to assume that Page Speed Insights will be more useful, as it always includes a performance report as well as many other metrics additionally to the accessibility assessment. The overall recommendation is to first find a tool that is applicable to the situation for each individual auditor. From this decision on the tools, commonly provide the same answer to a given test, where they focus on different details depending on which tool was chosen.

Chapter 6

Concluding Remarks

At the end of your survey, give a clear recommendation as to which approach or tool to use in which situation.

Bibliography

- Accessi [2022]. *Accessi*. 30 Nov 2022. <https://www.accessi.org/> (cited on page 16).
- ACM [2021]. *ACM Digital Library*. Full access to PDFs from IP addresses within TU Graz. 10 Nov 2021. <https://dl.acm.org/> (cited on page 1).
- Alexander Thien [2022a]. *Using Accessi for web accessibility*. 30 Nov 2022. <https://youtu.be/FqZQIkFlgYM> (cited on page 18).
- Alexander Thien [2022b]. *Using Page Speed Insights (Lighthouse) for web accessibility*. 30 Nov 2022. <https://youtu.be/oTBNtaBVXM0> (cited on page 18).
- Alexander Thien [2022c]. *Using WAVE for web accessibility*. 30 Nov 2022. <https://youtu.be/-URDq3PK8zs> (cited on page 18).
- Google [2022]. *Page Speed Insights*. 30 Nov 2022. <https://pagespeed.web.dev/> (cited on page 18).
- IEEE [2021]. *IEEE Xplore Digital Library*. Full access to PDFs from IP addresses within TU Graz. 10 Nov 2021. <https://ieeexplore.ieee.org/> (cited on page 1).
- King, Alasdair [2004a]. *Blind People and World Wide Web*. <https://www.webbie.org.uk/webbie.htm> (cited on page 3).
- King, Alasdair [2004b]. *Webbie*. <https://www.webbie.org.uk/index.htm> (cited on page 4).
- Lynx [no date]. *Lynx*. https://lynx.invisible-island.net/release/lynx_help/Lynx_users_guide.html (cited on page 4).
- Memon, Zeeman [2022]. *Best Linux Text-Based Browsers*. https://linuxhint.com/best_linux_text_based_browsers/ (cited on page 3).
- Ožbej Golob [2022a]. *Using JAWS on gov.uk site*. 30 Nov 2022. https://youtu.be/aX_vK0JpYZE (cited on page 9).
- Ožbej Golob [2022b]. *Using JAWS on mirror.co.uk site*. 30 Nov 2022. https://youtu.be/4zxA2TmmN_A (cited on page 9).
- Ožbej Golob [2022c]. *Using Narrator on gov.uk site*. 30 Nov 2022. https://youtu.be/ZVT012_vw1s (cited on page 9).
- Ožbej Golob [2022d]. *Using Narrator on mirror.co.uk site*. 30 Nov 2022. <https://youtu.be/LktZKcB7xNo> (cited on page 9).
- Ožbej Golob [2022e]. *Using NVDA on gov.uk site*. 30 Nov 2022. <https://youtu.be/oTBNtaBVXM0> (cited on page 9).
- Ožbej Golob [2022f]. *Using NVDA on mirror.co.uk site*. 30 Nov 2022. <https://youtu.be/SY2njevCTuI> (cited on page 9).
- Ožbej Golob [2022g]. *Using TalkBack on gov.uk site*. 30 Nov 2022. https://youtu.be/Hx_pI2bZ930 (cited on page 9).

- Ožbej Golob [2022h]. *Using TalkBack on mirror.co.uk site*. 30 Nov 2022. <https://youtu.be/EyYMB9Kn8Xw> (cited on page 9).
- Ožbej Golob [2022i]. *Using VoiceOver on gov.uk site*. 30 Nov 2022. <https://youtu.be/UBPnCozggK4> (cited on page 9).
- Ožbej Golob [2022j]. *Using VoiceOver on mirror.co.uk site*. 30 Nov 2022. https://youtu.be/_4sDqR2oT0c (cited on page 9).
- Riikola, Olli-Pekka [no date]. *Showcase video - the Lynx browser*. <https://youtu.be/fSYBxFXR4Q0> (cited on page 6).
- Robin Karlsson [2022a]. *Using axe web browser extension for accessibility audit*. 03 Dec 2022. <https://youtu.be/hMzs7v-xsDA> (cited on page 12).
- Robin Karlsson [2022b]. *Using Insights web browser extension for accessibility audit*. 03 Dec 2022. <https://youtu.be/4kIIB09bnhA> (cited on page 12).
- Robin Karlsson [2022c]. *Using Lighthouse web browser extension for accessibility audit*. 03 Dec 2022. https://youtu.be/ZuJ0I8LN_6s (cited on page 12).
- Robin Karlsson [2022d]. *Using SiteImprove web browser extension for accessibility audit*. 03 Dec 2022. <https://youtu.be/tGB5CJ3qU4U> (cited on page 12).
- Robin Karlsson [2022e]. *Using WAVE web browser extension for accessibility audit*. 03 Dec 2022. <https://youtu.be/vhMk3B3gPFE> (cited on page 12).
- W3M [no date]. *Lynx*. <https://w3m.sourceforge.net/> (cited on page 4).
- W3M [2002]. *Lynx*. <http://links.twibright.com/about.php> (cited on page 4).
- Web AIM [2022]. *WAVE Web Accessibility Evaluation Tool*. 30 Nov 2022. <https://wave.webaim.org/> (cited on page 17).