|  |
| --- |
|  |
| Accessibility Review |
| Open Source Math Homework Systems |

Prepared by Josie Gray, Advisor, Inclusive Design and OER Collections

BCcampus

March 21, 2020

A picture containing drawing

Description automatically generatedThis report is released under a [Creative Commons Attribution 4.0 International licence](https://creativecommons.org/licenses/by/4.0/).

# 

# Table of Contents

[Overview 3](#_Toc45186417)

[Tools 3](#_Toc45186418)

[Accessibility Standards 3](#_Toc45186419)

[Assessment Process 4](#_Toc45186420)

[Report 5](#_Toc45186421)

[Introduction 5](#_Toc45186422)

[Math Content 5](#_Toc45186423)

[Commitment to Accessibility 6](#_Toc45186424)

[WCAG Conformance 7](#_Toc45186425)

[General Usability 8](#_Toc45186426)

[Conclusion 8](#_Toc45186427)

[Appendix A: Identified Accessibility Issues 9](#_Toc45186428)

[Numbas 9](#_Toc45186429)

[IMathAS 10](#_Toc45186431)

[WeBWorK 11](#_Toc45186436)

[Appendix B: WCAG Criteria Mentioned in Report 12](#_Toc45186437)

[References 14](#_Toc45186438)

# 

# Overview

This report presents findings of an accessibility review of the student interfaces of three open source math homework platforms: IMathAS, WeBWorK, and Numbas. This review was conducted as part of the BCcampus Open Homework Systems Project. Note that the accessibility of instructor interfaces and their ability to create equations were not assessed.

Each platform was assessed through the following three lenses:

1. The conformance for all student-facing interfaces to WCAG 2.0 Level AA.
2. The active and visible commitment of developers to maintaining and improving the accessibility of the tool for all students.
3. The usability of the tool from a student’s perspective, including the ability to access the platform on multiple devices and availability of support resources for students.

## Tools

Each platform was reviewed on a computer and on a mobile device. I used the following tools to complete the review:

Computer: Microsoft Surface Laptop with Windows 10

* Browser: Firefox
* Screen reader: NVDA
* Accessibility checkers: WAVE and AChecker

Mobile: Google Pixel 3 smartphone

* Browser: Google Chrome

## Accessibility Standards

Each platform was evaluated based on the standards set in the **Web Content Accessibility Guidelines (WCAG) 2.0, Level AA**. WCAG is an internationally recognized standard that provides the basis for most international accessibility rules and legislation, including the United States’ *Americans with Disabilities Act* (ADA) and Section 508 of the *Rehabilitation Act*, which dictates accessibility standards for federal agencies (WebFX, n.d.). In Canada, WCAG is also the standard referred to in the *Accessibility for Ontarians with Disabilities Act* (AODA) (Government of Ontario, 2019). Similar laws have also been passed in Manitoba in 2013 and Nova Scotia in 2017 (Essential Accessibility, 2020). While British Columbia does not currently have provincial accessibility legislation, it is in the works. The Ministry of Social Development and Poverty Reduction is leading the project and concluded its consultation phase in November 2019 (Government of British Columbia, 2019).

WCAG defines three levels of accessibility conformance: A, AA, and AAA. During this review, I used the Level AA of conformance, which is the recommended conformance level for accessible websites. There is one higher level of conformance (Level AAA), but

it is generally not recommended to adopt AAA as a policy because it is not possible to satisfy all Level AAA Success Criteria for some content (Digital Education Strategies, The Chang School, 2019).

**Note about WCAG 2.1:** The W3C released WCAG 2.1 in 2018. These updated guidelines include additional considerations that address mobile accessibility, people with low vision, and people with cognitive and learning disabilities (W3C, 2019). These new guidelines were not factored into this review.

## 

## Assessment Process

Each platform was assessed through the process described in this section.

First, **documentation for the platform was reviewed**. The focus was on documentation that specifically addressed the accessibility of the platform and any documentation created to support students in using the platform. The goal was to discover what work the creators have done to make the platform accessible, the current status of accessibility of the platform, their ongoing commitment to accessibility, and what kinds of supports exist for students using the platform.

From there, the actual **accessibility of the platform was tested** following guidelines outlined in [*Professional Web Accessibility Auditing Made Easy*](https://pressbooks.library.ryerson.ca/pwaa/), published by Ryerson University’s Chang School. This included a combination of automated and manual testing. Common pages and functionality that students could be expected to access and navigate were identified and taken through the following tests:

1. **Automated accessibility checker.** I used both AChecker and WAVE to check accessibility. These tools made it easier to evaluate the accessibility of colour, links, images, and the structure of the page.
2. **Tab key navigation.** I conducted a Tab-key navigation test. This involves navigating through the entire page using only the Tab key. This is meant to tell me if the page is set up in a way that makes sense (i.e., what is the tab order?), if all functionality can be accessed using the Tab key, and if I can visually see where the focus is as I navigate through the page.
3. **Screen reader test.** I used the NVDA screen reader to test each platform for screen reader accessibility. This is to ensure that someone using a screen reader can easily navigate and access all information on the page and that information and functionality is presented in a way that makes sense.
4. **Magnification and mobile view.** To test the responsiveness of each tool, I magnified it in my browser to 200 per cent to ensure that the content was still navigable. From there, I accessed each platform from my mobile phone, a Google Pixel, to see what the mobile experience was like.
5. And finally, while exploring each platform and testing for accessibility, the **general usability of the platform was assessed** to evaluate the student experience. This was a very subjective analysis and should not be considered representative. I am someone comfortable learning online and navigating complex digital platforms, and I have a personal computer and reliable access to high-speed Internet. Also, note that I have never taken a math class at a university level.

Report

Introduction

When viewed through an accessibility lens, WeBWorK is the strongest platform. WeBWorK developers have made visible and active commitments to accessibility, and support resources have been created for students using screen readers. WeBWorK’s efforts include presenting at accessibility conferences and publishing a Voluntary Product Accessibility Template (VPAT). WeBWorK is the only tool to claim full WCAG 2.0 AA conformance, an assessment that holds up in accessibility testing. When testing, you can see the work developers have done to make the tool robust, accessible, and easy to use for students.

In contrast, both Numbas and IMathAS have varying levels of accessibility. Numbas has provided updated accessibility statements for both the student and instructor interfaces, but a number of accessibility issues were identified during testing which may make the platform difficult to navigate for students using screen readers. As for IMathAS, there is no recent or robust commitment to accessibility. The only accessibility notice I could find likely dates to before 2008. This is reflected in the accessibility testing, which identified a number of navigation and usability issues that would make it more difficult for students using keyboards to navigate and those using screen readers to access content.

Math Content

All three platforms have comparable math-content accessibility. Each of the platforms uses LaTeX, a common markup language for creating math equations and scientific documents. The LaTeX is rendered with MathJax, an open source JavaScript display engine. MathJax allows math to be displayed properly in various browsers, and it can take LaTeX and convert it to MathML, which ensures math content can be read by screen readers that are set up to read math equations. In addition to being more accessible to screen readers, MathJax allows equations to be more responsive. Students can increase or decrease the size of math equations and can collapse and expand sections of an equation to work through it, piece by piece.

Ultimately, the accessibility of the math content in these three platforms is largely the responsibility of question creators. Equations need to be written so they prioritize semantic meaning over display. In addition, any images included in questions need to include alternative text to allow students who can’t see them to access the content. From what I can find, none of the tools provide resources for instructors on how to make sure they author accessible questions. However, in a presentation at a conference in 2014, WeBWorK described some work they had done to make collections of accessible equations (Portland Community College Disability Services, 2014).

Commitment to Accessibility

Looking at the information available for each tool that addresses accessibility, the three platforms are clearly on three different levels.

WeBWorK provides the most robust documentation related to accessibility and seems to be the most proactive in ensuring their platform is accessible to students with disabilities. For example, according to WeBWorK documentation, accessibility is a “high priority project,” and they have an open call for people who are looking to contribute to or test the accessibility of the tool (WeBWorK, 2019). WeBWorK has also actively engaged with the accessibility community. For example, their developers have given presentations on the platform at accessibility conferences, including the AHEAD Conference in 2014.

In January 2015, WeBWorK published a Voluntary Product Accessibility Template (VPAT), which presents a self-evaluation of the accessibility of the student interface of WeBWorK. VPATs are documents that describe how a tool or platform conforms to the United States’ Section 508 Standards for IT accessibility and are used by federal agencies to make procurement decisions (Section508.gov, 2018). The WeBWorK VPAT claims support for all applicable accessibility criteria, except for criteria that addresses support services and documentation for students with disabilities. For those considerations, the VPAT claimed WeBWorK “partially supports or supports with exceptions” because it is an open source tool run by volunteers and its capacity to provide direct support to students is limited. The VPAT can be viewed here: [WeBWorK 2015 VPAT [PDF]](http://spot.pcc.edu/~ajordan/WeBWorKVPAT2-10.pdf).

Most significantly, WeBWorK claims full WCAG 2.0 AA compliance and provides an [Accessibility Guide for WeBWorK](https://webwork.maa.org/wiki/Accessibility_Guide). This guide provides an overview of the accessibility of the platform and provides support for students using screen readers in navigating and interacting with the platform. This guide would be very helpful in orienting these students to the platform for the first time and speaks to the proactive work developers have done to ensure this tool is accessible to all students.

Following WeBWorK is Numbas. Numbas’ documentation provides a high-level overview of the accessibility of the platform and claims that they aim to meet WCAG 2.1 AA level standards. They provide two accessibility statements: one for the student interface (Exams) and one for the authoring interface (Editor).

Numbas has clearly prioritized the accessibility of the Exams interface. The [Numbas Exams Accessibility Statement](https://docs.numbas.org.uk/en/latest/accessibility/exam.html) (Newcastle University, 2019b) provides an overview of accessibility considerations that they prioritized in the design, including the ability to zoom, use of colour, keyboard navigation, screen reader compatibility, responsive layout, and ensuring that the student interface works on a range of browsers. According to their accessibility statement, Numbas has tested the student interface with NVDA and Orca screen readers, but that it should also work with JAWS, too.

The accessibility statement also provides a list of tips on how students can adapt and interact with the content in different ways, including how to change the size of text and images, navigate with a keyboard, print exams, and use a screen reader. While Numbas’ accessibility guide provides a broader range of tips than WeBWorK’s, they are all very basic. WeBWorK’s accessibility guide provides more robust support for students using screen readers.

Overall, the Numbas team seems to have a strong understanding of web accessibility principles. However, the robustness of their testing is unclear as they do not seem to have done any user testing with students with disabilities or conducted a comprehensive accessibility audit of the tool. In addition, while they claim the student interface is accessible, in the accessibility statement of the Editor interface, they identify a number of accessibility issues (Newcastle University, 2019a).

Of the three, IMathAS has the least information available about the accessibility of the platform. Any accessibility information for this tool was difficult to find. There is no mention of accessibility on IMathAS’s main page, nor was it listed within the various menus. Accessibility information was finally found after using the following search terms on Google: “accessibility site:http://www.imathas.com/,” which found the following page: [IMathAS Accessibility](http://www.imathas.com/accessibility.html). This page is very out of date, since it references WCAG 1.0, which was replaced by WCAG 2.0 in 2008. However, it does show that the developers had a clear understanding of web accessibility and had considered the accessibility of the tool at least at one point in time. Ultimately, the lack of up-to-date accessibility information does not signify an active commitment to accessibility of IMathAS on the part of the developers.

## 

## WCAG Conformance[[1]](#footnote-1)

Of the three platforms, WeBWorK showed the most attention to accessibility for people using screen readers. It is the only platform to provide a “Skip to Content” link, which makes the platform much easier to navigate with a keyboard (2.4.1). It also reads out “correct” or “incorrect” notices when students submit their work. In addition, the WeBWorK platform doesn’t include decorative images with alt text, which helps to simplify the platform for people using screen readers. In contrast, I experienced a number of problems when trying to navigate the interfaces and assignments in IMathAS and Numbas with a screen reader. In Appendix A, I have provided a description of the accessibility issues identified in Numbas and IMathAS.

General Usability

Based on my extremely subjective evaluation (with a note that I don’t have experience with usability assessment or university math), I found the usability of the math activities for IMathAS and WeBWorK fairly comparable.

Of the three, I found the Numbas platform the least usable from a student perspective. You navigate from one question to the next using the left-hand menu, which lists all questions available in a quiz. In contrast, WeBWorK and IMathAS provide next and previous arrows, with the ability to navigate to a certain question if desired. Of the three, WeBWorK is the only tool that allows students to check how their answer will display before submitting. This gives students the chance to catch and correct any coding errors they may have made.

The biggest difference between the three platforms was their comparative complexity from a student’s perspective. Numbas is a less complex system than IMathAS or WeBWorK because it can be accessed directly within an institution’s LMS. In contrast, both IMathAS and WeBWorK require students to create their own accounts and access an external tool. Of the three, IMathAS is the most complex platform, as it acts like a math-centric LMS and not just a homework system. It includes forums, gradebooks, a calendar, course maps, and messages. In addition, content is organized into folders or modules, so all content related to the class would be provided there.

All three platforms can also be accessed on a mobile device. However, the Numbas exam interface was quite tiny on a mobile device and required zooming to read text and interact with content.

Conclusion

Overall, WeBWorK provides the most accessible platform for students completing math homework. It adheres to Web Content Accessibility Guidelines and includes features that make it easier to navigate for students using keyboards and screen readers. In addition, WeBWorK has made visible commitments to improving and maintaining the accessibility of the tool, which suggests any future accessibility issues that may be identified will be prioritized and quickly addressed.

In contrast, there are a few problems that the Numbas and IMathAS platforms should address to improve their usability and increase accessibility for students using assistive technologies. Both tools would benefit from user testing with students with disabilities, who would be able to provide the most accurate feedback about how the platforms can be improved.

And finally, one of the biggest issues is the accessibility of the math content itself, which is in the control of the question authors. Resources should be created to support instructors in creating accessible math content. Resources could and should include an explanation of the difference between semantic and display LaTeX equations and guidelines for creating text alternatives for complex images. It would also be useful to look into how each of the platforms could support and prompt instructors to create accessible content.

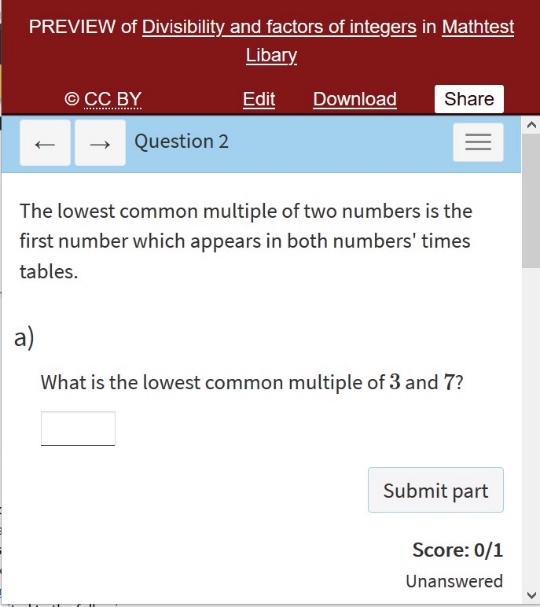
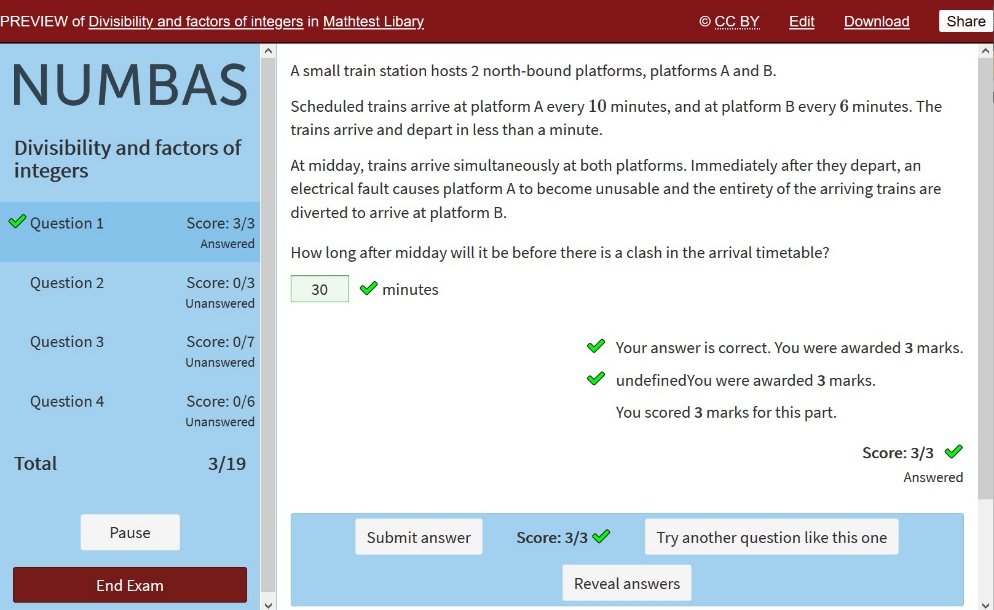
Appendix A: Identified Accessibility Issues

Numbas

For Numbas, I used the “Test Run” function available through the instructor’s editing interface to test various questions and exams in Numbas. I was unable to test the accessibility of Numbas when embedded in an LMS.

**Issue 1 (WCAG 2.1.1):** When previewing an exam in full screen, someone who is navigating with a keyboard is not able to access the left-hand menu. This is a significant barrier, as this menu presents a table of contents for the exam and allows the student to navigate between questions. Without access to this navigation menu, a student would not be able to advance to the next question. Workaround: When you reduce the screen size, next and previous arrow buttons appear, which you can then navigate with a keyboard.

**Issue 2 (WCAG 4.1.3):** When you click “Try another question like this one” or “Reveal answers,” a dialogue box pops up, but the text is not automatically announced. You just hear “Clickable Okay button.” In addition, [NVDA] + [b], which should read the dialogue box out, does not work. You have to use the arrow keys to get the text to be read out loud.

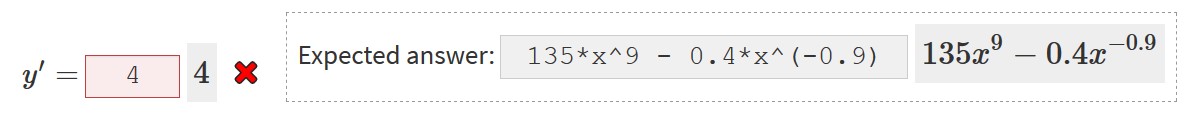


A screenshot of a cell phone

Description automatically generated

Potential issues when using a screen reader

* When you reveal an incorrect answer, the program immediately says, “Your answer is not correct.” However, you have to navigate back through the page to have the correct answer read out.
* When you reveal an answer, the screen reader reads through both what the student would type in and the MathJax equation without separating them.



IMathAS

Login page

* WCAG 1.4.3: Colour contrast of page titled “[About Us](https://mathtest.bccampus.ca/app/IMathAS/index.php)” is 1.87:1, which is not a high enough ratio for large text.
* WCAG 1.3.1: There are no page regions or heading structure, which may not be a problem because it is a short page.
* WCAG 1.3.5 and 1.3.1: On the “New Student Signup” page, a screen reader doesn’t read out “Course ID” or “Enrollment Key” when focus is on those form fields.

Forums

* WCAG 2.1.1: Students navigating the interface with a keyboard are only able to write forum posts in plain text because the rich text options are not accessible via a keyboard.

Math activities

* After reading the question, the screen reader will read out tips that aren’t visible on the screen.
* A screen reader reads “pts” as “P-T-S” rather than “points,” which may be confusing.
* When you submit an answer, a screen reader will sometimes read through the entire question again before it will tell you if you were correct or incorrect. It seems like it might be out of order. However, this isn’t a consistent problem.
* WCAG 3.2.2: Links to videos or examples open in new windows with no notice provided visually or for people using screen readers.
* WCAG 2.4.1: No ability to skip to the question when arriving on a new page. A student using a keyboard has to tab through the whole menu for each new question.

Math in Society (Lippman) course

* There are duplicate links to weekly content on the main course page (folder icon and text are both linked separately). And the link on the folder just reads “Folder graphic link,” which isn’t useful.
* When in a course, there are two menus (a top menu and a left-hand menu) that contain the same links.
* WCAG 1.3.1: Posts do not use headings.
* WCAG 2.4.1: No ability to skip to the main content when arriving on a new page. A student has to tab through the whole menu.
* WCAG 3.2.2: Links that take students to external resources open in new tabs. There is a visual marker that this will happen, but that behaviour is not communicated to students using screen readers.

The documentation for both WeBWorK and Numbas is posted on forums or wikis. In contrast, information associated with IMathAS is provided on a website. While a close assessment of the accessibility of documentation was not a part of this review, a few significant accessibility errors were identified in the [student-facing IMathAS documentation for entering answers](https://mathtest.bccampus.ca/app/IMathAS/info/enteringanswers.php) that should be noted. These include but are not limited to the following:

* WCAG 1.3.1: There are no page regions.
* WCAG 1.1.1: Fractions and equations are presented as images without alternative text.
* WCAG 1.4.3: The colour contrast of the page title is not a high enough ratio.
* WCAG 1.3.1, 1.3.2: The table is not accessible.

WeBWorK

I was unable to find any obvious accessibility issues in the student-facing WeBWorK interface. This platform has clearly been designed with screen-reader users in mind.

Appendix B: WCAG Criteria Mentioned in Report

1.1.1 Non-text Content — Level A

All non-text content that is presented to the user has a text alternative that serves the equivalent purpose. This means that images that convey information that isn’t already provided in the text require text descriptions.

1.3.1 Info and Relationships — Level A

Information, structure, and relationships conveyed through presentation can be programmatically determined or are available in text. For example, the HTML identifies the header, navigation menu, main content, and footer. This criterion also applies to the use of semantic headings and data tables.

1.3.2 Meaningful Sequence — Level A

When the sequence in which content is presented affects its meaning, a correct reading sequence can be programmatically determined. This means that things like headings and headers are used to ensure information on a page or within a table is presented in the correct order.

1.3.5 Identify Input Purpose — Level AA

The purpose of each input field collecting information about the user can be programmatically determined when: A) The input field serves a purpose identified in [Input Purposes for User Interface Components](https://www.w3.org/TR/WCAG21/#input-purposes); and B) The content is implemented using technologies with support for identifying the expected meaning for form input data.

1.4.3 Contrast (Minimum) — Level AA

The visual presentation of text and images of text has a contrast ratio of at least 4.5:1. Large text must have a contrast ratio of at least 3:1.

1.4.4 Resize Text — Level AA

Except for captions and images of text, text can be resized without assistive technology up to 200 percent without loss of content or functionality.

1.4.10 Reflow — Level AA

Content can be presented without loss of information or functionality, and without requiring scrolling in two dimensions.

2.1.1 Keyboard — Level A

All functionality of the content is operable through a keyboard interface.

2.4.1 Bypass Blocks — Level A

A mechanism is available to bypass blocks of content that are repeated on multiple web pages. This is helpful if there is a large menu that repeats on every page that someone using a keyboard may want to be able to skip.

2.4.7 Focus Visible — Level AA

Any keyboard-operable user interface has a mode of operation where the keyboard focus indicator is visible. This means that, as a user tabs through a page, they should be able to see where the focus is.

3.1.2 Language of Parts — Level AA

The default human language of each web page can be programmatically determined. By identifying the language of the page, screen readers are able to read content in the appropriate language. It also makes it easier to automatically translate content.

3.2.2 On Input — Level A

Changing the setting of any user interface component does not automatically cause a change of context unless the user has been advised of the behaviour before using the component.

4.1.3 Status Messages — Level A

In content implemented using markup languages, status messages can be programmatically determined through role or properties such that they can be presented to the user by assistive technologies without receiving focus.

References

Digital Education Strategies, The Chang School. (2019). [*Professional Web Accessibility Auditing Made Easy*](https://pressbooks.library.ryerson.ca/pwaa/)*.* CC BY-SA. https://pressbooks.library.ryerson.ca/pwaa/

Government of British Columbia. (2019, September 16). [Accessibility through legislation](https://engage.gov.bc.ca/govtogetherbc/consultation/accessibility-through-legislation/). *govTogetherBC*. https://engage.gov.bc.ca/govtogetherbc/consultation/accessibility-through-legislation/

Government of Ontario. (2019, October 22). [How to make websites accessible](https://www.ontario.ca/page/how-make-websites-accessible). *Ontario*. https://www.ontario.ca/page/how-make-websites-accessible

Essential Accessibility. (2020, April 28). [An overview of Canada’s accessibility laws: A look at the old and the new](https://www.essentialaccessibility.com/blog/canadian-accessibility-laws/). https://www.essentialaccessibility.com/blog/canadian-accessibility-laws/

Newcastle University. (2019a, October). [Accessibility statement for the Numbas editor](https://numbas-editor.readthedocs.io/en/latest/accessibility/editor.html). *Numbas Editor Documentation*. https://numbas-editor.readthedocs.io/en/latest/accessibility/editor.html

Newcastle University. (2019b, October). [Accessibility statement for Numbas exams](https://numbas-editor.readthedocs.io/en/latest/accessibility/exam.html). *Numbas Editor Documentation.*  https://numbas-editor.readthedocs.io/en/latest/accessibility/exam.html

Portland Community College Disability Services. (2014, July 14). [AHEADWeBWorK [Video]](https://youtu.be/pfY5eUYyY24). YouTube. https://youtu.be/pfY5eUYyY24

Section508.gov. (2018, April). [Voluntary Product Accessibility Template (VPAT)](https://www.section508.gov/sell/vpat). https://www.section508.gov/sell/vpat

W3C. (2019, April 8). [What’s new in WCAG 2.1](https://www.w3.org/WAI/standards-guidelines/wcag/new-in-21/). https://www.w3.org/WAI/standards-guidelines/wcag/new-in-21/

WebFX. (n.d.). [ADA vs. 508 Compliance vs. WCAG](https://www.webfx.com/web-design/ada-vs-508-compliance.html). https://www.webfx.com/web-design/ada-vs-508-compliance.html

WeBWorK. (2019, December 13). [Accessibility Guide](https://webwork.maa.org/wiki/Accessibility_Guide). https://webwork.maa.org/wiki/Accessibility\_Guide

1. Numbers in parentheses in this section (i.e., 1.1.1) refer to the relevant WCAG criteria. Full descriptions of the WCAG criteria mentioned in this review are provided in Appendix B. [↑](#footnote-ref-1)