

2020

Accessibility for Everyone

Daniel Newell
Grand Valley State University

Follow this and additional works at: <https://scholarworks.gvsu.edu/cistechlib>

ScholarWorks Citation

Newell, Daniel, "Accessibility for Everyone" (2020). *Technical Library*. 363.
<https://scholarworks.gvsu.edu/cistechlib/363>

This Project is brought to you for free and open access by the School of Computing and Information Systems at ScholarWorks@GVSU. It has been accepted for inclusion in Technical Library by an authorized administrator of ScholarWorks@GVSU. For more information, please contact scholarworks@gvsu.edu.

Accessibility for Everyone: A Framework for Teaching Accessibility to Web Developers

Daniel Newell

A Project Submitted to

GRAND VALLEY STATE UNIVERSITY

In

Partial Fulfillment of the Requirements

For the Degree of

Master of Science in Applied Computer Science

School of Computing and Information Systems

December 2020



The signatures of the individuals below indicate that they have read and approved the project of Daniel Newell in partial fulfillment of the requirements for the degree of Master of Science in Applied Computer Science.

Robert Adams, Project Advisor 12/17/2020

Robert Adams, Graduate Program Director 12/17/2020

Paul Leidig, Unit head 12/17/2020

Abstract

Computer accessibility is defined in this context as the accessibility of a computer system to all people, regardless of disability type or severity of impairment. As developers we are taught how to build a table for any styling, but not taught how to make sure that table could be used by an individual with a visual impairment. Naively, we are sometimes under the impression that we don't need to or even can't solve the needs of all individuals when it came to common software, and boy are we wrong.

It is our responsibility and privilege as developers to create software that works for all individuals and in all situations to the best of our ability. This application's primary focus is to further the education for designers and developers alike in the accessibility needs of our users. This application goes through a brief history of accessibility, includes many resources for implementors, and includes a series of lessons intended to increase the knowledge of the accessibility needs of users and give tips and tricks of how we can solve them. It by no means is a comprehensive guide of all that pertains to accessibility but an excellent starting point for individuals who are looking to see how they can better support all of their users through technology.

Introduction

The web is an everyday tool that is more and more becoming an essential part of modern life. It is our responsibility as developers to ensure we are making the web an inclusive place for all of its users regardless of how those users interact with it. Accessibility allows us the opportunity to bring in new customers, and to comply with legal and international standards for web experiences. The World Wide Web Consortium (W3C) has defined a specific set of requirements that need to be followed when creating web applications.

Contrast

Contrast is defined as the state of being strikingly different from something else in juxtaposition or close association. Which means that in order to have good contrast we need to have different colors placed close to each other when designing and developing our applications. This is important for a wide variety of impairments such as color blindness, visual deficiencies and even screens with poor color resolution.

Color Usage

Color usage is extremely important when it comes to accessibility. It can be used to differentiate boundaries as well as convey messages. But as with every accessible feature the goal is to ensure that all users can access whatever information is being conveyed by the color choices.

When we think about color, we often think about supporting people who experience “Color Blindness” but there are many other types of individuals that need to be considered, like the following:

- Users with partial sight often experience limited color vision
- Some older users may not be able to see color well
- People using text-only, limited color, or monochrome displays may be unable to access color-dependent information
- Users who have problems distinguishing between colors can look or listen for text cues
- People using Braille displays or other tactile interfaces can detect text cues by touch

Interactive Elements

The most important feature when it comes to interactive elements is to make them easily identifiable. Developers need to make sure that whatever disability the user is working with, they understand that a button is a button, and a link is a link, and a form element is a form element. There are an infinite number of ways to accomplish this and there are new designs coming out daily that make it easier and easier for a user to identify these elements.

Navigation

Navigation is one of the most imperative elements when it comes to creating an accessible site. Navigation elements are the main capability that allow users to traverse and navigate a web experience. Even users without accessibility needs have experienced how difficult it is to navigate a web experience if they can't understand how to move around. It can be off putting and frustrating and can cause users to abandon the experience all together.

Navigation is also something that can change drastically between different screen sizes. Often time how you navigate a desktop experience can be very different then how you navigate a tablet experience and how you navigate a tablet experience can often be very different then how you navigate a mobile experience.

User Feedback

There are many ways in which we can provide user feedback, such as confirming form submission, alerting the user when something goes wrong, or notifying the user of changes on the page. Applications should clearly identify important feedback that requires user action and should be present it in a prominent style. It is important that we are offering multiple feedback cues to our users as well as offering those feedback cues in a consistent manner. A quick example of this is would be having a common error messaging system.

Spacing

When it comes to spacing, we want to make sure that we have sufficient spacing around elements that need to be identified. One easy example is the text of a button. A button presumably has an outline, and we want to make sure that the button outline does not interfere with the user's ability to read the text of that button. As a developer we accomplish this using padding in the element to ensure a sufficient amount of space between the text and the border. We also want to make that the button is spaced properly from its neighboring elements which gives the user not only a good view, but also a large enough of a clickable area to make the button easier to use.

ARIA (Accessible Rich Internet Applications)

The acronym ARIA stands for Accessible Rich Internet Applications, and it is set of HTML attributes that developers can use to give their users a more accessible experience. It supplements HTML so that interactions and widgets commonly used in applications can be passed to assistive technologies when there is not otherwise a mechanism. For example, ARIA enables accessible navigation landmarks in HTML4, JavaScript widgets, form hints and error messages, live content updates, and more.

Accessibility for Everyone

Web accessibility as a topic and a set of standards has been around for quite some time now, but many developers are still unaware of the standards in general and are vastly unaware of how to make their application comply. This capstone project consists of a web

application that addresses the education of those developers and attempts to inform them of key topics around accessibility by providing history, resources and online learning modules to educate developers. This web application is an Angular 10 application built using Angular Material Design standards. It is hosted on an Amazon Web Services (AWS) S3 bucket. The application is data driven using various JSON objects to populate the content and overall experience of the site, allowing it to easily grow and incorporate new lessons, and resources, creating a fuller richer user experience.

History

These screenshots include view of the history page within the application. As you can see it covers topics including what is accessibility, who sets accessibility standards and what are some core accessibility principles.

HISTORY

What is Accessibility?

In modern life the web has become much more than a convenient tool or a collection of nice to have features, it has become an everyday essential part of life. Maybe your work email comes to your phone, or you stream the latest news, pay your bills online, or use apple maps to get from you home to the grocery store, you are in fact using the web. This is why it has become imperative to make sure that the web is usable by all individuals, even those suffering from various disabilities. It is our responsibility as designers and developers to ensure that the web is inclusive to everyone.

**The following content is taken from [Web Accessibility Initiative](#)*

Web accessibility means that websites, tools, and technologies are designed and developed so that people with disabilities can use them. More specifically, people can:

- perceive, understand, navigate, and interact with the Web
- contribute to the Web

Web accessibility encompasses all disabilities that affect access to the Web, including:

- auditory
- cognitive
- neurological
- physical
- speech
- visual

Web accessibility also benefits people without disabilities, for example:

- people using mobile phones, smart watches, smart TVs, and other devices with small screens, different input modes, etc.
- older people with changing abilities due to aging
- people with "temporary disabilities" such as a broken arm or lost glasses
- people with "situational limitations" such as in bright sunlight or in an environment where they cannot listen to audio
- people using a slow internet connection, or who have limited or expensive bandwidth

HISTORY

Who sets Accessibility Standards?

There are a couple sets of standards when it comes to web accessible guidelines, but the most notable guidelines are the [WCAG 2.0 Standards](#) Which were release in 2008. They are internationally recognized as a gold standard for making web application accessible to the widest range of users and disabilities.

These standards are created and maintained by [World Wide Web Consortium \(W3C\)](#) which is an international organization designed to help design web application standards.

Accessibility Principles

Perceivable Information and User Interface

- Provide alternative text for non text based assets
- Provide captions for audio based content
- Provide easily distinguishable content
- Provide content that can always be seen and heard

Operable User and Navigation

- Provide functionality that is available from keyboard
- Provide sufficient time for all users to interact
- Provide easy and understandable ways to navigate
- Provide abilities for users to input and interact without a keyboard

Understandable Information and User Interface

- Provide text that is readable and understandable
- Provide content that operates predictably
- Provide proper feedback to users to correct errors

Robust Content and Reliable Interpretation

HISTORY

Accessibility Principles

Perceivable Information and User Interface

- Provide alternative text for non text bases assets
- Provide captions for audio based content
- Provide easily distinguishable content
- Provide content that can always be seen and heard

Operable User and Navigation

- Provide functionality that is available from keyboard
- Provide sufficient time for all users to interact
- Provide easy and understandable ways to navigate
- Provide abilities for users to input and interact without a keyboard

Understandable Information and User Interface

- Provide text that is readable and understandable
- Provide content that operates predictably
- Provide proper feedback to users to correct errors

Robust Content and Reliable Interpretation

- Provide assets that are compatible with current and future technologies

How can I learn more?

[Redacted]

HISTORY

Perceivable Information and User Interface

- Provide captions for audio based content
- Provide easily distinguishable content
- Provide content that can always be seen and heard

Operable User and Navigation

- Provide functionality that is available from keyboard
- Provide sufficient time for all users to interact
- Provide easy and understandable ways to navigate
- Provide abilities for users to input and interact without a keyboard

Understandable Information and User Interface

- Provide text that is readable and understandable
- Provide content that operates predictably
- Provide proper feedback to users to correct errors

Robust Content and Reliable Interpretation

- Provide assets that are compatible with current and future technologies

How can I learn more?

[Go To Resources](#)

[Go To Lessons](#)

Resources

This screenshot is a view of the resources page of the application. This page includes various helpful resources and tools to give the user further detailed knowledge around different accessibility topics.

RESOURCES

- W3C and WAI**
Strategies, standards, resources to make the Web accessible to people with disabilities.
- WAI Design**
Design tips for creating accessible web applications.
- WAI Dev**
Development tips for creating accessible web applications.
- Contrast Checker**
Quickly check the contrast between two different colors.
- WCAG Standards**
Technical standards for web accessible applications.
- Accessibility Fundamentals**
Introduction and fundamentals for web accessibility.
- Google Standards**
Google accessibility standards and resources for designing great user experiences.

Aria Guides
Accessible rich internet applications

Lesson Example

These screenshots show the 7 lessons the application provides. Below you can see an example of a single lesson. All lessons follow the same general format as it is the content of the lesson that changes. The first two screenshots show example of the reading section of

the lesson. This is where the user can educate themselves on that specific topic. The next screenshot is the quiz portion of the lesson, where the user will be asked a series of multiple-choice questions on the given topic. Lastly a certificate is given showing the user they have completed the lesson.

Accessibility 1

1 Reading / Example 2 Quiz 3 Certificate

ARIA

Background

The acronym ARIA stands for accessible rich internet applications, and it is a set of HTML attributes that developers can use to give their users a more accessible experience. It supplements HTML so that interactions and widgets commonly used in applications can be passed to assistive technologies when there is not otherwise a mechanism. For example, ARIA enables accessible navigation landmarks in HTML4, Javascript widgets, form hints and error messages, live content updates, and more.

To highlight an example of what aria elements can do for you application we have chosen to show the HTML markup for a progress bar. As you can see the element we have chosen is just a div element. By itself it is not able to tell a screen reader what the purpose of the element is and it most certainly can not tell any specific details about the element. So we have added our attribute on the element. The first element we have added is the role attribute. This attribute is used to define the overall purpose of the element, the next element is `aria-valuenow` and is used to show the current progress of the loading bar. The third and fourth key attributes are `aria-valuemin` and `aria-valuemax` and they are used to show the min and max values of the progress bar respectively. All of these elements are key examples of how we as developers can add key information to elements to help with accessibility needs.

```
Incorrect  
<div id="percent-loaded"></div>
```

```
Correct  
<div id="percent-loaded" role="progressbar" aria-valuenow="75" aria-valuemin="0" aria-valuemax="100"></div>
```

For a more complete and live example please click on the "Live Example" button in the bottom left of your screen.

Standards

There are five key rules to keep in mind when it come to aria attributes.

Live Example Next

GVSU Capstone CIS 693

Accessibility 1

1 Reading / Example 2 Quiz 3 Certificate

Corrective rule states that if you give an HTML element a specific role it should act like that role. For example, if using `role="button"` the element must be able to receive focus and a user must be able to activate the action associated with the element using both the enter (on WIN OS) or return (MAC OS) and the space key.

Do not use `role="presentation"` or `aria-hidden="true"` on a visible focusable element

A general rule of thumb for showing and hiding elements is `aria-labels` that describe that are not needed. When we dynamically show and hide elements the elements are added automatically. So explicitly defining them in the markup is not necessary.

All interactive elements must have an accessible name

There are certain elements we need to link through HTML attributes. The most common example of this is input forms. When we have an input element we often accompany that input box with a label. Now there are several ways that we can do that but as long as they are created as separate elements we need to link them. Below you can see two examples that are both syntactically correct however as you can see in the second example the two HTML elements are linked.

```
Incorrect  
<label>user name</label> <input type="text">
```

```
Correct  
<label for="uname">user name</label> <input type="text" id="uname">
```

Resources

There are many valuable resources that can help you with ARIA attributes, below are just a few.

- [W3 Documentation](#)
- [Mozilla Documentation](#)
- [Google Documentation](#)

Live Example Next

GVSU Capstone CIS 693

1 Reading / Example 2 Quiz 3 Certificate

ARIA

What does the acronym ARIA stand for

- Accessible Rich Internet Applications
- Accessible Reliant Information Applications
- Automated Reliance of Internet Accessibility
- Automated Rich Internet for Accessibility

Which is not a key rule to remember with ARIA attributes

- Use native elements where you can use them
- All interactive elements must have an accessible name
- HTML elements must have a limit of 3 ARIA attributes
- Do not change native semantics, unless you really have to

Live Example Next Previous

1 Reading / Example 2 Quiz 3 Certificate

Congratulations

You have completed the following course:

ARIA Attributes

Live Example Next Previous

Project Management

For the project management I operated in a Kanban approach. I had a list of tasks that were prioritized in the order in which I need to accomplish them. I estimated each task and added a brief acceptance criterion so I could understand when that task was “completed”. I had weekly reviews with my advisor where we discussed the previous weeks progress as well as what was to be accomplished in the next week.

Organization

There are two major pieces in my project. The first was the main infrastructure of the application. I chose to create an Angular 10 application that would be deployed onto an AWS S3 bucket to be hosted. This was the bulk of the work to make sure these capabilities were functional. The three main capabilities I had to build were a history component, a resources component and a lesson component. I wanted these components to be data driven so they could grow overtime. The second piece of work was the content. I had to build the content of each component to complete the experience. The history was a brief history of accessibility, the resource component was a collection of useful accessibility resources and the lesson component was populated by JSON data that the application contained, one JSON file per lesson.

Reflection

This was a project that I was passionate about, however was a little naive about the scope of what I originally wanted to accomplish. Accessibility being such a large topic it was difficult to narrow down the key areas the project could focus on. As with everything, there was a deadline to meet and I quickly learned how I needed to scale back the scope in order to meet it. Looking back there was a little more time spent on the workings of the application, opposed to the content, than needed.

Conclusions

This project taught me a great deal about not only about accessibility but also about UI development in general. One of those most interesting takeaways from learning about accessibility is that often times, the improvements that we make for accessibility purposes are not just beneficial to users with different disabilities but tend to create richer and more helpful user experience for all users. This project accomplished a great deal including creating a working data driven application that includes lessons, history and various other resources. It includes seven lessons of various accessibility topics. At this time there are a couple minor issues including a small bug for the navigation menu items. They are not updated until the application is refreshed, even though they should be updated in real time. In the future I will continue to expand the lesson plan including more detailed lessons for users to continue their accessibility education. As this is an ever-growing topic this application needs to continue to grow and refresh the lesson plan with the needs of accessibility users.