# Simplifying the WCAG:

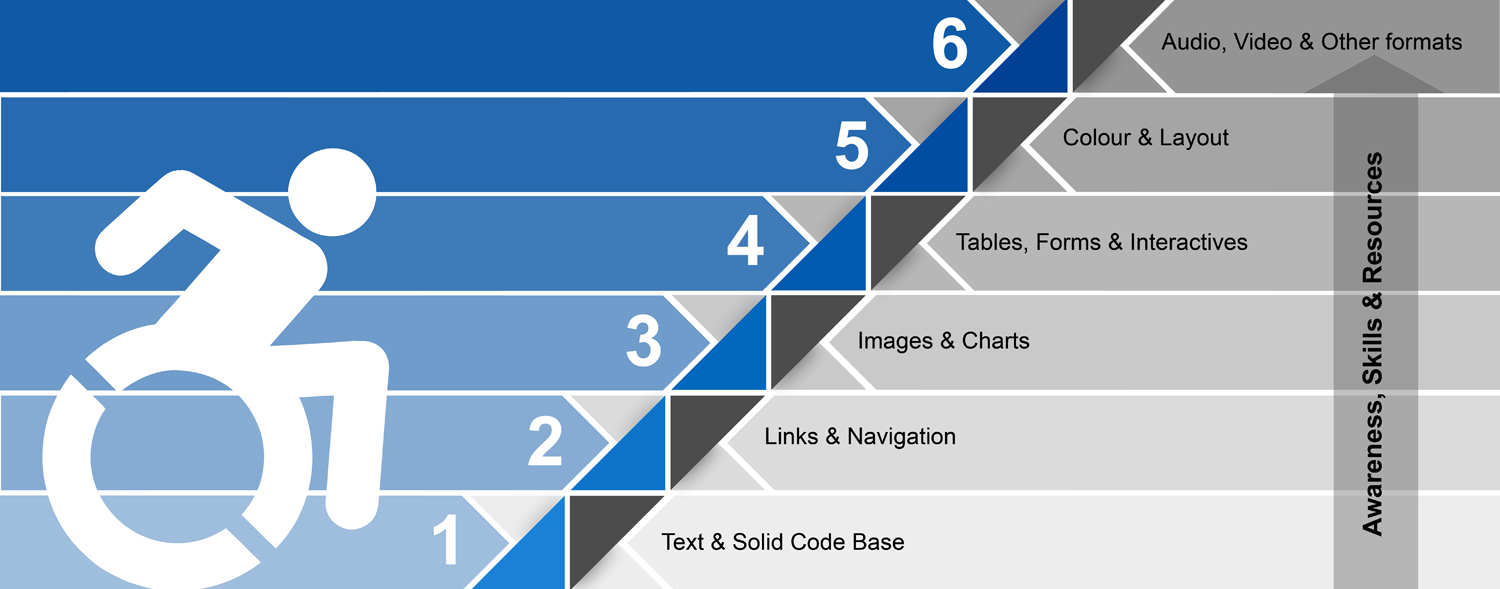
# An Action Based Guide to Coding in the Virtual Ramp

## Introduction

Accessibility legislation (Americans with Disability Act (ADA) 1968) for the physical environment was introduced in the late 1960's, on the heels of the civil rights movement. Accessible standards for the built environment came into being, as a result. Eventually, architectural features such as ramps in public buildings and sidewalk curb cuts on streets became more common. These new architectural features, originally built for people with disabilities, actually benefit parents with strollers, delivery people, skate boarders, rollerbladers, cyclists and people in wheel chairs. Even pedestrians often head for the curb cut instead of stepping directly off the curb to cross a street. It has turned out turned out that architectural design made with inclusion in mind is universally beneficial. This is known as the “curb-cut effect” (Steve Jacobs 1999).

Thirty years later, in the digital realm, accessibility legislation (Section 508 of the revised Rehabilitation Act, 1998) was enacted in order to ensure that users of assistive technologies (AT) would be able to access digital content and be active participants on the world wide web. The Web Content Accessibility Guidelines (WCAG) (published 1999; updated 2008) accompanied other web standards to ensure an accessible web from the beginning. As in the physical environment, there are universal benefits when designers and developers build websites with accessibility in mind. Websites that are built to standard and that are compliant with the WCAG not only work better with AT like screen readers and zoom text software, but are also generally easier to use and understand. They work better on mobile devices, perform better, are easier to maintain and are generally of a higher quality. As a result, websites built and designed to standard often have a higher return on investment. This has become known as the “electronic curb-cut effect” (Steve Jacobs 1999).

In spite of the realization that accessible design is universally beneficial, there has not been wide implementation of the Web Content Accessibility Guidelines (WCAG 2.0). Part of the reason for this is the unnecessary complexity in the way that the guidelines are organized and explained. There are 4 principles (the POUR principles), 12 guidelines, 65 success criteria and 3 grading levels. Even if you are a seasoned web developer this can sound a little daunting and intimidating. In this paper, I focus on a conceptual approach to the WCAG based on the decisions we make during the design process. I feel that this approach is well aligned with the WCAG's four main principles. These are the “POUR” principles: a website must be **P**erceivable, **O**perable, **U**nderstandable and **R**obust. I then visualize each potential barrier a website might contain as a step (Numbered 1 to 6 in Figure 1) that represents a particular content type or design choice. The corresponding techniques that enable web content to be accessible are then visualized as a virtual ramp. There a several techniques for each stair. As we make our way up the *staircase* the content types and design choices become more complex, thus the *virtual ramp* requires more awareness, skills and resources to make the *virtual stairs* accessible.

Figure 1 The 6 steps in the virtual stair case represent potential barriers to accessibility, while the inclined plane is the virtual ramp which represents strategies to enable accessibility. As one goes higher, more awareness, skills and resources are required. 

## 1. Text & Solid Code Base

Electronic text is accessible by nature: it can be interpreted and read by all devices. The beauty of electronic text is that it can be rendered visually, in sound, or, in tactile format such as braille (WCAG). There are, however, a few critical techniques that one must employ to ensure that all aspects of text-based web content is made accessible to all users and all devices.

### 1.1 Code Meaning into the Content

HTML's built-in semantic structure should be used to *code meaning directly into the content.* HTML is a structural markup language. It gives us the tools to define the meaningful (semantic) structure of the content. This structure is invisible to most users, but not to machines such as screen readers. When the most meaningful HTML element is used for a particular piece of content, AT can interpret and convey that meaning to the user. Using HTML correctly and in the most meaningful way also improves a designers ability to style that content more efficiently. Thus many benefits result - from better user experience to higher returns on investment – when designers and developers take the time to use semantic HTML markup.

# Code sample:

{{DEMO in ChromeVox: Headings, quote, list of items.}}

### 1.2 Language

One should use the *lang* attribute to define a page's natural language and to mark any changes in that language. Defining the document's language and marking changes is essential to ensure proper pronunciation by screen reader and text-to-speech software.

# Code samples:

<html lang=”en”>....

<p>Hello world can be expressed in french as <em lang=”fr”>Bonjour tout le monde.</em></p>

### 1.3 Reading Level

Web authors have a difficult job in composing content that can be understood by as many people as possible. A good rule of thumb is to keep it as simple as possible and appropriate to your audience. For example, a government web site has the full range of the population's reading level to consider when composing content, while a university professor writing content for a third year physics course has a much narrower target to meet.

### 1.4 Solid (Valid) Code Base

There are three required elements that are needed on every web page to be able to run the code through a validator (syntax checker). Doing so, ensures the code is error free and can be optimally accessed by assistive technology and display correctly in all browsers. These three essential bits of code are:

1. the DOCTYPE
2. the Character encoding (character set)
3. the HTML page TITLE element

Oddly enough, one or all of these three essential items are often missing from web page templates. Ensuring these three items are part of the all top level HTML page templates is the best starting point for robust code. The doctype and the character set are required in order to ensure that a browser (or AT) can properly interpret your code. A concise and well-written unique page title is the first element that a screen reader reads out, telling the user that they have made it to the intended destination. The page title is also really important for helping people find your web page. Search engine algorithms generally weight the content in the page title heavily. A good page title means that your web page content will be indexed correctly and therefore more easily found by the person looking for it.

### Summary for Text & Solid Structure

The perceivability, operability, understandability and robustness (POUR) of your website is profoundly affected by the content structure, the language setting, the reading level and the basic code base of your website. Basic awareness for content developers and very basic training for developers can ensure that the text on your site is accessible to everyone.

## 2. Links & Navigation

### 2.1 Meaningful links

Links are the stepping stones of the internet: they take users places. To ensure that users can easily determine that a link is really where they want to go, the text of the link needs to consist of unique, meaningful words that describe the destination and function of that link, and ensure that the destination and function are clear even when the link text is placed out of context. If the link wraps an image, the alternative text of that image must contain the destination and function of the link. It's also a good practice not to repeat the same link several times on the same page as it can be frustrating for users to click on a link to a place they have already visited. If duplicate links are required, it is important that the link text be consistent for each duplicate link, so all users can see that they have already been there.

# Code sample:

{{DEMO in ChromeVox: meaningful and not so meaningful links.}}

### 2.2 Changes in Context

A link is a change in context. Changes in context should be left in the control of the user. Specifically this means that one should not use the target attribute to force a new tab or window to open when a user clicks on a link. While this is unfortunately common practice, it is now deprecated code for the following reason: for some users (especially users of screen readers) this practice is extremely disorientating. The user immediately loses access to the page's browsing history which means that the browser's back button won't take them back to where they came from. At the very least a user should be warned of any changes in context in order to be ready for and be able to understand that change.

# Code sample:

Very common, but no longer valid code:

<a href=”[http://www.myfriendswebsite.com](http://www.mysite.com/)” target=”\_blank”>my friend's site</a>

### 2.3 Skip Navigation Links & Page Landmarks

To provide direct access to a page's main content, **visible** *skip navigation* links are a best practice. *Skip navigation* links are a fairly well-known accessibility technique that allow users of screen readers (or any keyboard user) to by-pass long lists of navigation (usually the first content on the page) to get directly to the page's main content. Designers often choose to hide these links from visual users. However, in reality, visual people who prefer to navigate using the keyboard also benefit from having access to these links in order to quickly get to the heart of the page's content.

HTML5 and WAI-ARIA give developers code features that define landmarks on a page. Landmarks on a web page work exactly the same as in the built environment – they are “helpful way finders” (Quesenbury & Horton). Users of screen readers can use the landmarks to navigate and scan through a page's content. Like *skip navigation* links, when page landmarks are employed a screen reader user can easily get to the section of content she finds most interesting.

# Code sample:

{{DEMO in ChromeVox or regular browser: a skip navigation.}}

### 2.4 Consistent Navigation

Well-planned,consistent navigationbenefits all users. Consistency improves usability and prevents users from losing their way.

### Summary for Links & Navigation:

Links and navigational elements are essential to a good user experience for everyone. Basic awareness training for content developers will make sure that meaningful links that are mindful of context changes get created over the lifespan of the website. A site's navigation and flow requires a hefty amount of design time for any web project. Adding accessibility goals to the design of navigation at the beginning of a project adds little to the overall budget item; however, attempting to add accessible navigation features at the end of a project can be extremely costly. Links and navigation are central to all websites and must be accessible to be perceivable, operable and understandable to all users.

## 3. Images & Charts

### 3.1 Text Alternatives

Providing meaningful text alternatives for all content images is a requirement of the WCAG to make image content perceivable to those who cannot see. The goalof text alternativesis to maintain the meaning of the document whether or not the user can see the image. There are several ways to provide text alternatives for images.

1. **In the code:** The HTML *alt* attribute is a required attribute for all images. A succinct description fits well in the *alt* attribute. The content of the *alt* attribute is read aloud by screen readers and browsers display this content on the screen if for some reason the browser cannot load the actual image.
2. **In a visible element:** In some cases a description of the image is beneficial for all users. A web author may choose to place the description of an image nearby in a nicely styled *image caption*. HTML5 has a new tag, *figcaption*, specifically for this purpose.
3. Another option or case could be that the image is fully **described in the body** of the nearby content, so no more supplementary description is needed to maintain the full meaning of the page.

In cases 2 or 3 above, and when an image is purely decorative, the *alt* attribute can be left intentionally blank to prevent a screen reader from reading out unnecessary duplicate content. In all cases, writing good alternative text and descriptions for images is not an easy task. Good image descriptions depend on context (WebAIM.org) and once the appropriate text is written by a web author, the designer needs to make sure it looks good and the developer has to make sure the description ends up in the right place in the code. So much of accessible web development requires a team effort. Like safety in the airline industry (Kolly & Groff), accessibility in web design needs to be everybody's responsibility. In the airline industry, safety was enhanced dramatically over a period of X years by making safety everyone's responsibility. Similarly, accessibility in web design must be made everyone's responsibility.

### 3.2 Complex Images

For charts or complex images you willlikely need to link to supplementary content on a separate page. To do this ensure your link to the description is accessible (i.e. is a meaningful link) and that an accessible link back from the description page is also provided. Transcripts to video can be handled in a similar way. Linking to long descriptions can also be achieved programmatically using the longdesc attribute in HTML versions prior to HTML5.

# Code sample:

{{Link to DEMO}}

### Summary for Images

Images are not textual content. To ensure the meaning of the images are perceivable to all users, we must provide an appropriate description of the image. This is an easy accessibility requirement to meet, but it is not always easy to get it right. Everyone on the team must understands what the accessibility requirements and goals are for the website, so that image descriptions do not fall through the cracks. Scrambling to write alternative text for images at the end of development right before you launch will cost more than working image descriptions into the natural design and development process.

## 4. Tables, Forms & Inter-actives

As we make our way up the steps of potential barriers, our web content types become more complex, and require more skill and resources to make them accessible. Just by their function and purpose, tables and forms are complex structures. To reduce complexity, these elements should be kept as simple as possible. For tables, this means avoiding complex data relationships – don't span rows and columns unless absolutely necessary. For forms, simplicity means only asking for what you need and nothing more.

### 4.1 Accessible Tables

A rich set of **semantic table mark-up** is available (*caption*, *thead, tfoot, tbody, th, td*); however, these elements are rarely used to their fullest potential. The *caption* element is of particular importance as it offers a meaningful summary description of the data presented in the table. Perhaps a summary is just what the user needs. If so, that user can skip the laborious task of reading the data data cell by cell. A *caption* is like a heading except it offers more as it is connected programmatically to the *table* element; a clear relationship exists in the code. When a user decides she really needs the data in the table, the quality of her reading experience depends on the other meaningful table elements. The *thead, tfoot* and *tbody* elementscreate meaningful sections in the table and the *th* elements act as labels for columns and rows and finally the *td* elements hold the actual data. Directional relationships can be further refined and defined by using the *scope* attribute on the *th* elements. And remember, web design has moved beyond deprecated layout techniques circa. 2005. We use **tables for tabular data only** – never for content layout!

# Sample code:

{{Link to DEMO}}

### 4.2 Accessible Forms & Inter-actives

Forms and interactive exercises (such as those used in educational resources on the web) are dynamic. A user enters data and makes choices of one kind or another. When ready, the user submits the information. Many things can go wrong and cause frustration. Before even getting to the code, we need to be clear about data requirements, and remember to only ask for what one needs. We also need to be clear about form flow and how best to communicate errors. Once these 3 items are clearly defined we can employ the rich set of **semantic form mark-up** to organize the controls and to code meaning right into the forms.

The most basic form relationship, the *label-control* relationship which programatically connects the form control with its label, is often left undefined, or worse, left out all together. This relationship is a must for an accessible form. It is defined both programmatically in the code and by visual proximity in the design.Clear organization will help define the flow and allow a user to tab through the form in a logical order from top to bottom. For a user of voice activation software, the visible submit button text must match the submit button text in the code or the user won't be able to submit the form data after taking the time to fill in the form.

The goal of a form is successful completion. It is paramount to clearly define errors and guide the user on how and where to fix them. The user must be given the power the power to choose more time to prevent any unwanted time-outs. The WAI-ARIA standard should be employed to define roles and behaviours that HTML cannot define on its own. With the rich semantic form code in place, the designer can use another web standard, CSS, to make the form look awesome.

# Sample form code:

<label for=”user-name”>Name</label><input id=”user-name”>

<label for=”password”>Password</label><input id=”password”>

<button for=”login”>login</button>

# Sample login form:

{{Link to a DEMO}}

# Important call out box

HTML5 has made many improvements for form controls. (Pilgrim, M.)

### Summary for Tables & Forms

As website builders or owners, we cannot predict how a user will access our site. She may use voice activation software, a screen reader or zoom text. He may have a bad shoulder or a busy hand and just prefer to use the keyboard instead of a mouse to tab through the form. Understanding the complexities of how to build accessible tables and forms can mean the difference between a user getting essential information or not getting that information or with forms, it can be the difference between success and failure, both situations can effect your bottom line.

# Important call out box

Data tables and web forms have both been the result of landmark legal cases in web accessibility (Maguire v. SOCOG 1999 & Jodhan v. Canada 2011) that were filed and won under human rights legislation.

## 5. Colour & Layout

Up until now, we have been focusing on HTML code and how to make our content meaningful by using the most meaningful HTML element for the job. Now, we will turn our attention to how design choices can make the difference between accessible pages or pages with barriers.

### 5.1 Colour

Due to many types of visual impairments, including colour blindness, as well as just differences in colour perception, users see colour differently. Because of this, web designers have to think very carefully and critically about colour. Naturally, this thought process, this design thinking, has to happen at the beginning of the design cycle and cannot be tagged on at the end. Colour choices are likely one of the most critical decisions a designer will make as these choices are often tied to many other aspects of a brand or product. Thus, designers must c**hoose colours wisely** and work the following golden rules about colour into their design process.

1. **Don't rely on colour alone** for meaning.

This means an element's colour cannot be the only way of determining the element's function or purpose. A very straightforward example is the styling of links. Links are functional elements that have to look different from the surrounding text and they are most often styled to have a different colour. To ensure that function is clear to a user who can't see that colour difference some other formatting like underlining, font-style and/or the addition of a small icon are all potential solutions for creating links that can be seen by everyone.

# Important call out box

Judy Brewer [R] has done a lot of work to help designers avoid choosing confusing colours. Please see the {{***Brewer Palette}}.***

1. Test for **sufficient contrast** ratios.

The WCAG criteria for sufficient contrast ratios between an element's foreground and background have been in place since the standard was created. These requirements are not difficult to meet and do not hinder visual aesthetics of a website when ratios are part of the colour decision making process. Making the decision to do accessible design means being aware of the required contrast ratios and testing your choices for every state of of every element (body text, link states, callout boxes, buttons, etc.)

# Important call out box

There are numerous contrast checking tools*.* I like the contrast checkers by {{Leah Verou}} , the {{Paciello Group}} and {{WebAIM}}.

# Important call out box

Links states, focus in particular, should be explicitly designed and defined. These states are: link, visited, hover, focus, active. A visual focus style is essential for keyboard users to see where they are on the page.

### 5.2 Layout

Using **CSS & HTML together properly** is the best method for making your content accessible and your entire website easier to maintain due to the robustness of the code base. The HTML standard was specifically designed for creating meaningful (semantic) document structure, so we use HTML to define the structural meaning and function a site's sections and content types. The CSS standard (Cascading Style Sheets) is employed to efficiently layout the sections and style the content. The two standards are meant to work together. Developers need to make choices to keep presentational elements out of the HTML and in the CSS. Awareness training and team style guides go a long way in creating and maintaining good coding practices. Training for web content authors is an important part of maintaining accessible content over the lifespan of the site. To keep content as accessible as possible, presentational elements need to stay out of website pages. This is a big challenge in web applications that allow web authors to create and edit content using minimalist HTML editors. Training to make content authors aware of the HTML features of these editors will help ensure that content remains accessible after the launch.

A skilled CSS/HTML developer can reduce the lines of code in a project, leading to a website of much higher quality. Quality code is less code; less code loads faster and is easier to maintain. An important concept to take into account is that the correct use of the HTML standard is most important in making electronic text accessible to assistive technology and the correct use of the CSS standard has high impact for visual accessibility. Both, together, form the foundation of accessible web design.

### Summary for Colour & Layout

Using open standards (HTML, CSS and ARIA) together with web industry best practices gives you the biggest return on your design investment. In such a dynamic industry, continual investment in innovation and training potentially have huge pay offs.

## 6. Audio/Video & Other Formats

### 6.1 Audio & Video

Like images, audio and video (multimedia content) are not text-based. To make multimedia perceivable, electronic text-based alternatives must be provided. These alternatives can take several forms such as captions, audio/video description and transcripts.

1. **Captions** for audio and video are the natural choice and can be open or closed. Closed captioning is usually the preferred implementation method and the user is provided an accessible control to turn captions on and off. High quality captions include the description of meaningful sound effects and follow recognized standards for readability.
2. **Audio description also called video description** is the primary technique for making visual media accessible to the blind and visually impaired. A second sound track that takes advantage of natural pauses in the dialogue to describe the actions is added. Audio description can also be open or closed.
3. **Transcripts** provide the dialogue in an asynchronous, but never the less a perceivable format. In some cases, a transcript makes better sense than captions. For example, for videos that consist of just a talking head, or perhaps an interview, a transcript that can be read along side of or instead of the media may meet a projects accessibility goals without the added expense of captions.

More importantly, however, access to the controls that play and stop the multimedia is often the first barrier. User control is paramount in all aspects of design. This principle is especially important with multimedia content. Multimedia content is multi-sensory and can be disruptive, so our first design principle is we need to take care to leave control with the user. This means, as designers, we do not have any media play automatically. The WCAG is more lenient this issue (specifically see Success Criterion 1.4.2), however, the WCAG's own help guidelines actively “discourage the practice of starting sounds automatically”, as it can interfere with a screen reader user's ability to find the control that stops the sound from playing. Thus, in order to make the media operable, the controls that start and stop the media must be keyboard accessible.

### 6.2 Other Formats

HTML content is the native content type for web pages. Thus, it is the natural choice for putting content on the web. Other types of document types, however, do exist, are important and are found on the web. The portable document format (PDF is the most common non-HTML document type found on web pages. PDFs, like HTML content need to be accessible. Luckily, the concepts for making PDFs accessible are very similar to those for making HTML content accessible. PDF documents need meaningful document structure and descriptions for images. PDF form controls must have appropriate descriptions.

The tools, however, to make PDFs accessible are indeed quite different and often require skills that a web designer/developer does not have. So, the first design question to answer when determining if PDF (or another non-HTML document type) is appropriate for the situation: is there a reason not to use HTML? If there is, indeed a reason to use PDF, then the steps to make that document accessible need to be learned and taken. The PDF standard has a content tag nomenclature akin to HTML, however, these “tags” are not as apparent nor as easy to adjust and change as HTML tags. To be accessible PDFs must be tagged.

### Summary for Audio/Video & Other Formats

Audio, video and other formats have been placed at the top of the staircase mostly because making these types of media accessible often require high costs. Special in-house human resources have to be allocated or a decision to out source this type of content has to be approved and made. Speech recognition software and auto captioning tools are getting better and better; however, formatting for grammar and readability are still a long way off from automatic. As the users of the web want more video driven content, vigilance is required to ensure this content is made accessible.

## Conclusion

When web design teams ramp up a website for users of assistive technology and other edge cases, the payoff happens at many levels. The website will likely be easier to find, be of a higher quality, be easier to maintain, last longer and work better for everyone - all of which contribute to a better return on the investment. Just like accessible ramps at building entrances, the virtual ramp built into websites benefits users of all abilities and disabilities.

## Things teams can do

1. Develop a team style guide both for code development and for visual demonstration.
2. Build a set of templates that create an accessible and robust starting point and provide documentation on how to use the templates.
3. Build an awareness around colour and colour contrasts for designers and web authors.
4. Involve people with disabilities in user testing.
5. Make sure everyone is on board. Accessibility is everyone's responsibility.
6. Advocate for, create and implement policy that supports accessibility.

## References (currently incomplete and not properly formatted)

Clark, Joe. (2003). *Building accessible websites.* Indianapolis: New Riders.

Horton, S. & Quesenbery, W. (2014) *Web For Everyone*. Rosenfeld Media. <http://rosenfeldmedia.com/books/a-web-for-everyone/>

Jacobs, S. (1999). *Section 255 of the Telecommunications Act of 1996: Fueling the Creation of New Electronic Curbcuts.* The Center for an Accessible Society website. Retrieved from: [http://www.accessiblesociety.org/topics/technology/eleccurbcut.htm#012](http://www.accessiblesociety.org/topics/technology/eleccurbcut.htm" \l "012)

Kolly, J.M. & Groff, L.S. (????). *The Benefits of a Safety Studies Program to Proactively Promote Aviation Safety.* National Transportation Safety Board. (need a more explicit study on safety).

Pilgrim, M. (????) A form of madness. In *Dive into HTML5.* <http://diveintohtml5.info/forms.html>

Accessible Icon Project: <http://www.accessibleicon.org/>

Brewer Palette: <http://colorbrewer2.org/>

HTML5: <http://www.w3.org/TR/html5/>

W3C: <http://www.w3.org/>

WAI: http://www.w3.org/WAI/

WCAG 2.0: <http://www.w3.org/WAI/WCAG20/glance/Overview.html>

WCAG 1.0: <http://www.w3.org/TR/WAI-WEBCONTENT/>

WCAG Samurai: <http://www.wcagsamurai.org/>

WAI-ARIA: <http://www.w3.org/WAI/intro/aria.php>

Wave Accessibility Evaluation Tool: <http://wave.webaim.org/>

WebAIM: <http://webaim.org/>

Wikipedia: <http://wikipedia.org/>

<http://www.w3schools.com/html/html_form_attributes.asp>

Paciello Group. Contrast Analyser Tool: <http://www.paciellogroup.com/resources/contrastanalyser/>

Verou, L. Contrast Ratio Tool: <http://leaverou.github.io/contrast-ratio/>

WebAIM Contrast Checker Tool: <http://webaim.org/resources/contrastchecker/>