

Accessibility Resources

H. Conrad Cunningham

13 April 2019

Accessibility Resources

I initially prepared this page as a handout for the Spring 2019 offering of CSci 487, Senior Project, in Computer and Information Science at the University of Mississippi.

What is Accessibility

To comply with legal expectations such as the Americans with Disabilities Act of 1990, the Rehabilitation Act of 1973, and other laws, court rulings, and international conventions, we must make our electronic documents, websites, software applications (mobile, desktop, web, etc.), video, audio, and hardware accessible to persons with a wide range of abilities and disabilities. This, of course, applies to materials for our courses, whether online or in the classroom.

As examples, consider users who have:

- visual impairments (e.g., blindness, low vision, color blindness)
- hearing impairments (e.g., deafness, hard of hearing)
- fine motor impairments (e.g., inability to use some input devices such as mice; necessity to use assistive technologies such as speech input, head pointers, or eye trackers)
- learning challenges (e.g., dyslexia, auditory processing disorder)
- limited English fluency or literacy
- mobile devices rather than full-size monitors
- low bandwidth or unreliable internet connections
- limited experience using computing devices

It is usually better to treat accessibility as a *design concern* from the beginning rather than an *add-on feature* at the end of development.

Online Resources

- [University of Washington, Getting Started with Accessibility](#)

- There are subpages on creating accessible documents, websites, and videos.
- [UW IT Accessibility Checklist and Tutorial](#)
- [Universal Design of Instruction \(UDI\): Definition, Principles, Guidelines, and Examples](#), by Sheryl Burgstahler, University of Washington, DO-IT Center.
- [Web Accessibility In Mind \(WebAIM\)](#)
 - [WebAIM’s Resources page](#)
 - [Introduction to Web Accessibility](#)
 - [WebAIM’s WCAG 2 Checklist](#)
 - [WebAIM Wave tool](#)
- [World-Wide Web Consortium \(W3C\), Web Content Accessibility Guidelines \(WCAG\)](#)

Although this explicitly addresses Web documents, most other electronic media use this description as the “definitive” set of industry guidelines.
- [Creating accessible Office documents \(Microsoft\)](#)
- [Pandoc and Markdown](#)

Cunningham note: My personal workflow for course materials is to use Pandoc-flavored Markdown to “encode” documents in plain text and the Pandoc to convert the document to other formats such as HTML, LaTeX/PDF, and Word.
- [Accessible R Markdown Documents](#)
- [Accessible Graphics Resources](#)

Web Content Accessibility Guidelines

Note: I adapted this section from a draft paper by Hao Zhou and H. Conrad Cunningham.

To clarify what is meant by accessibility in Web pages, the W3C standardized the Web Content Accessibility Guidelines (WCAG). WCAG 2.1 is organized according to thirteen technology-independent *guidelines* grouped according to the following four *principles*. Accessible documents must be [WCAG 2.1]:

1. perceivable
2. operable
3. understandable
4. robust

The guidelines are not testable, but WCAG also associates one or more testable *success criteria* with each each of thirteen guidelines. For a criterion to be testable, it should be

possible for either a program or a human reader to determine unambiguously whether the criterion is satisfied for a specific Web page.

Using the WCAG and its success criteria, WebAIM provides the web accessibility evaluation tool WAVE to help authors and publishers develop compliant Web pages.

Perceivable

The user must be able to perceive both the content and the user interface controls. To be *perceivable*, an accessible document must [WCAG 2.1]:

- “provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language”
- “provide alternatives for time-based media”
- consist of “content that can be presented in different ways (for example simpler layout) without losing information or structure”
- “make it easier for users to see and hear content including separating foreground from background”

As an example of a testable success criterion for the “provide text alternatives for any non-text content” guideline, WCAG requires any essential “non-text content that is presented to the user” to have “a text alternative that serves the equivalent purpose”. For instance, an image (i.e. HTML `` element) conveying essential information must include an alternative (i.e. `alt`) text attribute.

Operable

The user must be able to operate and navigate through the user interface. To be *operable*, the user interface of an accessible document must [WCAG 2.1]:

- “make all functionality available from a keyboard”
- “provide users enough time to read and use content”
- not present “content in a way that is known to cause seizures or physical reactions”
- “help users navigate, find content, and determine where they are”
- enable “users to operate functionality through various inputs beyond [the] keyboard”

As an example of a testable success criterion for the “help users navigate” guideline, WCAG requires the content to be organized using section headings arranged into a logical hierarchy (such as `<h1>`, `<h2>`, ... without skipping levels).

Understandable

The user must be able to understand both the content and the user interface operation. To be *understandable*, the presentation of an accessible document must [WCAG 2.1]:

- “make text readable and understandable”
- “make Web pages appear and operate in predictable ways”
- “help users avoid and correct mistakes”

As an example of a testable success criterion for the “make text readable” guideline, WCAG requires that it be possible for a program to determine the default human language for each Web page. For instance, a Web page should declare the default language using the `lang` attribute of the page’s `<html>` element.

Robust

A wide variety of “user agents” (both human users and assistive technologies) must be able to interpret the content reliably. To be *robust*, an accessible document must [WCAG 2.1]:

- “maximize compatibility with current and future user agents, including assistive technologies”

As an example of a testable success criterion for this guideline, WCAG requires that a page’s markup meet the specification for the markup language. For example, all HTML elements must have start and end tags, be validly nested, and not contain any duplicate attributes. All identifiers should be unique except in circumstances where duplication is allowed.