

# AudioEye's Accessibility Essentials

An Introductory List of AudioEye's Accessibility Best Practices

## **Purpose and Audience**

The information provided in this document is based on international guidelines provided by the World Wide Web Consortium (W3C), the Web Content Accessibility Guidelines Version 2.0 (WCAG 2.0). This document is not a comprehensive guide to understanding the Web Content Accessibility Guidelines 2.0, instead it attempts to provide industry best practices for many common issues and accessibility pitfalls. The information provided hopes to guide developers and design team members to create the most accessible and usable content for all users.

## **About WCAG 2.0**

The Web Content Accessibility Guidelines covers a wide range of recommendations and best practices that developers should follow in order to make web content more accessible. The guidelines are designed to ensure that content is accessible to a wide range of individuals with disabilities. The guidelines are a technology-agnostic list of testable statements that, when followed, ensure a high degree of access to the largest possible audience of consumers of the content.

For those developers interested in gaining a more comprehensive understanding of the Web Content Accessibility Guidelines published by the World Wide Web Consortium (W3C), the full text of the WCAG 2.0 is published here: <https://www.w3.org/TR/WCAG20/>.

[More information about W3C's Web Accessibility Initiative](#)

## Accessibility Essentials Topics

### Page Titles

#### BEST PRACTICES:

- A page should have a `<title>` that briefly and adequately describes the page's content.
- Each page should have a unique `<title>` to distinguish it from other pages.

### Decorative Images

#### BEST PRACTICES:

- Image information is provided by adjacent text
- Image conveys no meaningful information
- If at least one of the above are true, apply an empty alt attribute to the image.  
`<image alt="" ...`

### Non-decorative Images

#### BEST PRACTICES:

- All images should contain an alternative text (alt) attribute.
- If the image is linked, the alternative text should inform the user of the link destination. If there is room, a brief image description can follow the link destination text, but is not necessary.
- An image's alt attribute should contain a clear textual representation of the non-text content.
- If there is text present in the image, that text should appear in the alt attribute.
- An image's alt attribute should be descriptive, but brief. Alternative text should be less than 125 characters (excluding spaces). If it requires more than 125 characters to describe an image, then the long description (longdesc) attribute should be used in conjunction with the alt attribute.
  - The longdesc attribute should contain a link to another page or a section in the same page that describes the image/graphic in greater detail.

```
<image longdesc="anotherpage.html"
```

OR

```
<image longdesc="thispage.html#idOfDesc"
```

- Not all assistive technologies support long description links, to ensure the broadest reach add a compliance hidden anchor, that points to the long description element or page.

```
<a href="anotherpage.html"> Click to access  
images description</a>
```

OR

```
<a href="thispage.html#idOfDesc"> Click to access  
images description</a>
```

## Headings

### BEST PRACTICES:

- Headers should be used in their standard hierarchal flow. h1 tags should be used first followed by h2 for sub-sections with h3 for further subdivisions, and so on.
- Headers should not skip levels.
- Headers should not be used to style elements, but rather to convey the information present on the page.
- All pages should include one h1.

## Color Contrast

### BEST PRACTICES:

- Contrast ratio of at least 4.5:1 for normal text and images of text (14 point regular or smaller)
- Contrast ratio of at least 3:1 for large text and images of large text (18 point or 14 point bold or larger)
- Non-text, visual elements require a contrast ratio of at least 3:1 against adjacent color(s):
  - User Interface Components (i.e. controls like buttons)
  - Graphical Objects (meaningful graphics that help the user to understand the content, e.g. standalone icons or lines in a graph)

## Keyboard

### BEST PRACTICES:

- All functionality should be operable through a keyboard.
- All focusable components should receive focus in an order that preserves meaning and operability.
- Visible focus indicator should be easy to perceive (see “focus visible” best practices).

## Link Text

### BEST PRACTICES:

- Avoid ambiguous link text within the anchor like “Click here” or “Read more”. These kinds of links can be confusing when the context of the link is unknown.
- Link Text should inform the user of the destination.
- Avoid linking full URLs, i.e. <https://www.example.com/page/>.

## Link Appearance

### BEST PRACTICES:

- If the underline is removed for styling purposes, there are 2 styling requirements for links within body text:
  - The link text must have a 3:1 contrast ratio from the surrounding non-link text.
  - The link must present a "non-color designator" (typically the introduction of the underline) on both mouse hover and keyboard focus.
  - These requirements only allow for very limited colors options for links, so it is suggested that links in body text are underlined
- Use hover and focus effects like underlining, color change or background color to help user know that the link is an interactive element.  
Define `a:hover`, `a:focus` in CSS to have the same visual presentation so that keyboard users have the same experience as mouse users.

## Focus Visible

### BEST PRACTICES:

- Add a border around the interactive element and its label, typically using the CSS border or outline properties.
- Visual indicator must adhere to color contrast standards. Consider highly contrasting colors, a thick line, and other visual indicators such as glow.

- ❑ For links, a visual cue or indicator can be an underline (highly recommended), bold, italic or increase in font size or it can be the addition of an icon or graphic.
- ❑ Use the `a:focus` style for `a:hover` and, depending on the situation, even `a:active`.

## Adjacent Links

### BEST PRACTICES:

- ❑ Join adjacent links that navigate to the same location.
- ❑ Select one element to be the link or wrap all elements within one link.
- ❑ The image's alternative text should be null.

```
<a href="products.html">  
      
    Products page  
</a>
```

## Form Input field labels

### BEST PRACTICES:

- ❑ All form elements need a label.
- ❑ The `<label-for>` attribute should be applied to all labels, linking them to the id of the form input. The placeholder attribute for an input is not an accessible alternative to a label.
- ❑ All form elements and their associated labels should have a unique ID. While web standards dictate that all ID's be unique, it is extremely important that this is the case to ensure accessible forms.
- ❑ Form fields should be coded as a `<form>` element.

## Form Instructions

### BEST PRACTICES:

- ❑ If input is required, the label should state required, or if asterisk `*` is present, the top of the form needs to have the statement "Fields marked with asterisk `*` are required"

## Forms: Visual presentation

### BEST PRACTICES:

- Error states should be highly visible. Do not use color alone to convey meaning. Use color in addition to another visual indicator like icons, bold font weight, heavy border or outlining.
- Provide helpful text describing the problem and what needs to be fixed.
- Focus should be driven to the upper most text field error.

## Moving, Flashing, or Blinking Content

### BEST PRACTICES:

- If content moves, blinks, or scrolls for more than five seconds, it needs a pause or stop button.
- If there is automatically updating information, it needs a pause or stop button.
- Do not include content that flashes/blinks more than three times in one second.
- Endless single pages should be implemented with pagination links or “Load more” buttons.

## Tables

### BEST PRACTICES:

- Use table headers to categorize information. If the information in each of the table cells is not capable of being understood without a header, be sure to use `<th>` tags to define headers. This will associate a header description with each cell.
- Use the “scope” attribute to associate table headers with a row or column:  
`<th scope="col">` or `<th scope="col">`