

Web Accessibility Tips

Considering Various Disabilities

Difficulty: Beginner

Category:

- Principles

Web accessibility affects the following disability categories

- [Visual disabilities](#) - blindness, low vision, and color-blindness
- [Auditory disabilities](#) - deafness and hard-of-hearing
- [Motor disabilities](#) - difficulty using a mouse or keyboard
- [Cognitive or learning disabilities](#)
- [Photo-sensitive epilepsy](#)

While care should be taken when grouping anyone into a category, when approaching web accessibility issues, it's often useful to consider the distinct needs of users with each of these types of disabilities.

Defining Web Accessibility

Difficulty: Beginner

Category:

- Principles

Web accessibility is about reaching the broadest range of users regardless of age or disability. A web site can always be made more accessible, and thus, will always be inaccessible to someone. Guidelines, policies, and laws provide measures of accessibility that can be useful in establishing goals and in evaluating accessibility. Viewing web accessibility as a continuum on which improvement can always be made will help ensure that accessibility is continually improving and that it is about people, not merely compliance with law or guidelines.

Assistive Technologies

Difficulty: Beginner

Category:

- Principles

Assistive technologies, or AT, are technologies that help one overcome some physical issue or limitation so they can function in ways they otherwise would be unable to. AT includes both software and hardware. It can range from screen readers which convey information audibly to those who have [visual](#) or [cognitive disabilities](#), to captions for the [Deaf or hard-of-hearing](#), to refreshable Braille devices which present content to those who are deaf-blind, to screen enlargers for those with [low vision](#), to physical devices to assist those with varying levels and types of [motor function](#). For a web site to be accessible, it must be compatible with varying assistive technologies.

Screen Readers

Difficulty: Beginner

Category:

- Principles
- Structure
- Presentation

A [screen reader](#) is a type of assistive technology that reads the text content of a web page audibly to the user. It is primarily used by individuals who are [blind](#) or have [low vision](#), but can also be useful to individuals with certain [cognitive or learning disabilities](#).

Screen readers read the text content of a web page linearly based on the underlying source code order - left to right, top to bottom. Screen reader users typically do not use a mouse, but instead use a wide array of keyboard shortcuts to navigate and interact with a web page. A screen reader can also output content to a refreshable Braille device to form Braille characters,

thus allowing users who are deaf-blind to access web content. Design webpages with these principles in mind - techniques required include providing skip navigation links so that the screen reader users can bypass navigational menus and jump directly to the content, associating form labels with their appropriate form elements so the purpose of each form control is identified, identifying data table headers to aid in table orientation and navigation, etc. Much of web accessibility is about ensuring compatibility with screen reader technologies.

WCAG 2.0

Difficulty: Beginner

Category:

- Principles
- Validation

The [Web Content Accessibility Guidelines \(WCAG\) 2.0](#) is a W3C (World Wide Web Consortium) specification for measuring web accessibility. WCAG 2.0 is based upon four core principles - perceivable, operable, understandable, and robust; or as an acronym - POUR. [Ensuring POUR content](#) across disabilities will ensure highly accessible web content. These guidelines are most useful as tools for implementing and evaluating accessibility, but true accessibility is determined by the end user experience, not by compliance to a set of rules. Level AA conformance with WCAG 2.0 is a common standard measure of good accessibility.

Captions and Transcripts

Difficulty: Beginner

Category:

- Principles
- Content
- Presentation

In order for multimedia content to be perceivable to the Deaf and hard-of-hearing, [captions and transcripts](#) must be provided. Captions provide a synchronized text presentation of the audio content. Beyond providing basic accessibility to users with auditory disabilities, captions are also useful to other users in noisy or very quiet locations, increase accessibility to those with cognitive or learning disabilities, and provide enhanced comprehension of complex content. They are also helpful when the audio is unclear or in a language that is not well known.

Transcripts provide a static text version of the spoken dialog. Transcripts allow the information to be searchable (by both users and search engines), remove the time-based nature of the content (they can read the transcript at their own pace), and allow the content to be accessible to screen readers and refreshable Braille devices.

Captions and transcripts should be provided for video content and transcripts alone should be provided for audio content (e.g., podcasts).

Descriptive Link Text

Difficulty: Beginner

Category:

- Content
- Presentation

Screen reader users often navigate a page by reading only the links. The link text (typically blue, underlined text) is read. The function of the link should be available from the link text itself. Ambiguous links, such as "click here", "more", etc. do not provide useful information to these users. The user must navigate or read the link's surrounding context to determine what the link does or where it goes. These ambiguous links should be avoided and can almost always be made clearer.

Example:

Change "To perform a search, [click here](#)" to "[Perform a search](#)".

Color Dependency

Difficulty: Beginner

Category:

- Presentation

Individuals with [color-blindness or color deficiency](#) may have difficulty in differentiating between certain color combinations. This affects approximately 7% of males and .1% of females. Difficulty in differentiating red and green is most common - these are most often the colors used to convey meaning or differences. While using appropriate colors, providing good color contrast, and not using color as the only means of conveying content is important for those with color-blindness or color deficiency, these techniques may also improve accessibility to those who are blind (screen readers do not typically identify color) or have low vision (these users may override page colors to make the page more readable). An easy test for color dependency is to print the page on a black and white printer and see if any information is missing when the color is removed from the page.

Conveying Information Using Color Alone

Difficulty: Beginner

Category:

- Presentation

Avoid using color or other stylistic differences as the only means of conveying information or meaning. Blind users do not care about the color, font size, or styling of text, so long as the visual aspects of that text are not used to convey content. Screen readers generally do not present stylistic information to the end user. Users with certain types of color deficiencies (color blindness) may not be able to differentiate certain color combinations. Users with low vision may override page colors. Each of these may have difficulty differentiating content based on color alone. The following relies on color to convey information:

The green mushrooms listed here are OK to eat. The red mushrooms will kill you.

- Amanita
- Chanterelle
- Porcini
- Shitake
- Tylopilus

This can be made accessible by simply creating two lists - one for mushrooms that OK to eat and one for mushrooms that will kill you.

Form Labels

Difficulty: Beginner

Category:

- Structure
- Presentation
- Validation

Most types of form elements (text boxes, text areas, checkboxes, radio buttons, etc.) have a text label adjacent to them that identifies the function of that particular form element (e.g., "First name" adjacent to a text box). Sighted users make a visual association between a label and its corresponding form element based on proximity and positioning. A user who is blind cannot make this visual association, so the label must be programmatically associated with its form element using the HTML <label> tag. When properly associated, a screen reader will read the text label when it encounters the form element.

```
<label for="fname">First name:</label> <input type="text" id="fname">
```

For more information, see [Creating Accessible Forms](#)

Page Title

Difficulty: Beginner

Category:

- Content
- Presentation

Providing a descriptive, succinct page title is important for accessibility. The page title (usually appears at the very top of the browser window) is the first thing seen by sighted users, is the first thing read by screen readers, and is used by people and search engines to determine the content of the page. Because the page title is read on each page, it should be short (generally no more than a few words). The site name and other information may be included in the page title if it is succinct and informative (e.g., "Acme, Inc.: Order Form")

Keyboard Accessibility

Difficulty: Beginner

Category:

- Structure
- Presentation

Ensuring that your web page is fully accessible using only the [keyboard](#) is a vital component of web accessibility. All interactive elements, including links, buttons, form fields, media controls, etc. should be accessible using only the keyboard. Use the Tab key to navigate through page elements. Shift + Tab will navigate backward. The Enter key will typically activate links, buttons, etc. The navigation order of interactive elements should be logical and intuitive. It should generally follow the visual presentation order (left to right, top to bottom). Particular care should be taken to ensure that forms can be completed and submitted using only the keyboard. If you must press the Tab key a significant number of times to interact with a page, consider adding a ["Skip to main content" or "skip navigation" link](#).

Seizure Inducing Media

Difficulty: Beginner

Category:

- Presentation

Flashing, strobing content can cause [photo-epileptic seizures](#). In order to cause a seizure, strobing content must flash more than 3 times per second, be sufficiently large, and have significant contrast in the flashes. The color red is also more likely to cause a seizure. Bright flashing content, particularly in videos, must be avoided.

Animation

Difficulty: Beginner

Category:

- Presentation

Animation or other moving elements on a web page can be distracting to all users. Even simple animation can render a web page completely inaccessible to users with certain cognitive disabilities or high levels of distractibility. Animation should generally be avoided. Brief animations (ones that completely stop in less than 3 seconds), however, can be used to draw attention to important page elements, such as error messages or alerts.

A common use of animation to increase accessibility and usability is the implementation of what is called the Yellow Fade Technique. This technique presents a bright, quickly fading background to important page elements or messages.

Line Length

Difficulty: Beginner

Category:

- Presentation

Pages should be designed so the line length, the number of characters that appear per line, should be neither too short nor too long. Anything more than around 80 characters can introduce reading difficulties in scanning from the end of a line of text to the beginning of the next line. Short line lengths also introduce difficulty and vertical scrolling. The width of the body area of a web page should be constructed to ensure that lines of text are constrained to both minimum and maximum lengths across both small (such as mobile devices) and large screen resolutions.

Multiple Ways of Accessing Content

Difficulty: Beginner

Category:

- Content
- Structure
- Presentation

Some users prefer using a search form to quickly find information. Others may follow navigation or other links. Others may prefer a site map or index. Providing [multiple ways](#) of getting to content will improve the usability and accessibility of your site.

Clear and Simple Writing

Difficulty: Beginner

Category:

- Content
- Presentation

[Clear and simple writing](#) is one of the most important, yet often neglected, aspects of web accessibility. Technical accessibility provides access to content, but that content must be written and presented so it is understandable by the audience. This includes considering the use of headings and content sections, avoiding slang and jargon, being mindful of the required

reading level, defining acronyms and abbreviations, avoiding misspellings, and presenting the text in a highly readable font face and presentation.

Error Prevention, Validation, and Recovery

Difficulty: Beginner

Category:

- Content
- Presentation

[Form error prevention, validation, and recovery techniques](#) can have a significant impact on site usability and accessibility. Forms should be easy to understand, complete, and submit. If there are problems with a form submission (e.g., the user did not complete a required form field), you should:

1. Alert the user to the presence of the error in an apparent and accessible manner
 2. Allow the user to easily access the form elements that need to be modified
 3. Allow resubmission and revalidation of the form
-

Skip Links

Difficulty: Beginner

Category:

- Structure
- Presentation

["Skip to main content" or "skip navigation" links](#) provide a mechanism for keyboard users to jump over repetitive navigation directly to the main content of a page. These links must be one of the first links on a page and must be visually apparent in order to provide quick access to the main content for keyboard users, including screen reader users. They can be intrusive to visual design. However, because "skip" links are really only useful to keyboard users, they can be

hidden from view until they receive keyboard focus, at which point they are presented visually and prevalently within the page - thus maintaining high utility to both sighted and blind keyboard users, but having no impact on other users - they probably won't even know that the link exists.

[Read more about hiding "skip" links visually.](#)

Alternative Versions

Difficulty: Beginner

Category:

- Principles

Building one version of a site that is highly accessible is the best approach to accessibility. Providing alternative versions or directing users to accessible or mobile versions of a site can be interpreted a segregation and separation, particularly if the content and functionality is not similar or equivalent to the main site. [Statistics](#) show that screen reader users are not likely to use this content anyway. A good site has full accessibility built in naturally, thus removing the need for alternative versions.

Cognitive Disabilities

Difficulty: Beginner

Category:

- Principles
- Content

The number of people with [cognitive disabilities](#) is greater than the number people with all other disabilities combined. Cognitive disabilities are complex and vary greatly. [Recommendations for those with cognitive disabilities](#) share much overlap with general usability recommendations, however, the impact of poor usability and understandability can

have a much more significant impact on those with cognitive disabilities. Care should be taken to ensure optimal functionality and understandability of web content by those with cognitive or learning disabilities.

Robustness

Difficulty: Beginner

Category:

- Validation

[Robustness](#) addresses the technology implications of accessibility. It requires following web standards (e.g., [valid HTML](#) and [CSS](#)), creating content/functionality that works across various user agents (web browsers and assistive technologies), ensuring compatibility with various hardware (such as mobile devices), and considering present, future, and older technologies. While it does not mean that a web site must be compatible with *all* present, future, or former technologies, it does suggest that consideration be given to ensure accessibility to the broadest range of technologies feasible.

Text Size and Zoom

Difficulty: Beginner

Category:

- Presentation

Content must remain readable and understandable when [text size](#) or page zoom is set to at least 200% or twice the default size. While it is optimal to decrease the likelihood of horizontal scrollbars or changes in visual design or document layout at these settings, readability and understandability are most critical to accessibility. This can be tested by selecting Control + (or Command + on a Mac) in your web browser or by increasing the text size under the View menu.

Contrast

Difficulty: Beginner

Category:

- Presentation

Determining whether there is enough color contrast between text and background can be difficult. [WCAG 2.0](#) has contrast guidelines at both level AA and level AAA and a complex formula for determining a contrast ratio. Large text (over 14 point bold) requires slightly less contrast than 'normal' text. There are many tools, such as [WebAIM's color contrast checker](#) that can generate this contrast ratio and give immediate feedback whether the contrast is sufficient. Simply enter a foreground and background color to see the results. Users of the Firefox web browser can use the [Colorzilla](#) add-on too easily select color values from within the web page.

Link Contrast

Difficulty: Beginner

Category:

- Presentation

Links should be clearly distinguishable from surrounding text. The default underline for links clearly distinguishes them visually. Underline should generally not be used for text that is not linked. If underlining is disabled in CSS for links, [WCAG 2.0](#) requires sufficient [contrast](#) between the link text and the surrounding non-link text. Because the linked text and the non-linked text must also have sufficient contrast from the background, the window of colors available for non-underlined links is very narrow. WCAG 2.0 also requires that non-underlined links provide a non-color designator (typically addition of the underline) on both mouse hover and keyboard focus links are clearly differentiated as users interact with them. The easiest way to avoid these additional requirements is to underline your links by default.

Evaluation Tools

Difficulty: Beginner

Category:

- Validation

Web accessibility evaluation has many aspects. No automated [evaluation tool](#) can tell you if your site is accessible. A good evaluation tool will guide a person through evaluation of true accessibility. While guidelines and checklists are useful, a site can be compliant to an accessibility guideline, standard, or checklist, yet still be totally inaccessible to some users. Screen reader or other user testing is a wonderful way to ensure accessibility for that user, but may not provide sufficient insight into true accessibility. Only a combination of effective automated testing, manual checks, and user testing will provide a good assurance of the accessibility of a site.

See the [WAVE Web Accessibility Evaluation Tool](#).

Images

Difficulty: Beginner

Category:

- Content
- Presentation

[Images](#) can enhance accessibility. The expression "a picture is worth a thousand words" very much applies to accessibility as well. Complex concepts and content can be presented through images, illustrations, charts, graphs, etc. Icons can present complex functionality or information in a very small amount of space. As long as images are given [appropriate alternative text](#) and are presented in accessible ways, they can greatly improve the functionality, understandability, and accessibility of a web page.

Alternative Text

Difficulty: Beginner

Category:

- Content
- Presentation

[Alternative text](#) conveys the content and functionality (if present) of an image. It is very rarely a description of what the image looks like. When authoring alternative text, consider what the core information presented by the image is. Asking the question, "If you could not use the image, what text would you use to replace it?" is often a good test to determine appropriate alternative text.

Succinct Alternative Text

Difficulty: Beginner

Category:

- Content
- Presentation

[Alternative text](#) should be succinct. There is not a defined maximum length for alt text, but a rough guideline of around 100 characters is generally a good threshold. Alternative text longer than around 100 characters typically indicates that the content should be provided some other way (such as in text in the context of the image, on a separate description page, etc.)

Screen Reader Testing

Difficulty: Advanced

Category:

- Validation

Evaluating the accessibility of a web site in a screen reader can provide an interesting and accurate representation of true accessibility. [Screen readers](#) are very complex software and can be difficult to learn and use. Learning enough to simply listen to a web page and navigate through a web page can be sufficient to identify accessibility issues. Tutorials are available for using [JAWS](#) and [NVDA](#) for web accessibility evaluation. To ensure that your experience is accurate, turn off your monitor and don't use a mouse.

Tabindex and Reading/Navigation Order

Difficulty: Advanced

Category:

- Content
- Structure
- Presentation

The underlying source code determines the default reading and navigation order of content in a web page. [Tabindex](#) can be used to specify a specific navigation order, but this does not affect the reading order through the page. Tabindex can almost always be avoided if the source code order is logical and intuitive. The reading and navigation order should generally closely follow the visual order of the page. For complex layouts, disabling styles or using the structure/order or text-only views on [WAVE](#) can allow you to easily determine default reading/navigation order.

Guidelines for Alternative Text

Difficulty: Advanced

Category:

- Content
- Presentation

Alternative text for images can be presented either in the alt attribute of the image (``) or in text near an image (e.g., an image text caption). When a caption or surrounding text provides the full content of the image, the image can generally be given a null or empty alt attribute (`alt=""`) to avoid redundancy. Screen readers identify the presence of images, so it is not important to indicate in alt text that it is an image (i.e., avoid "image of..." or "graphic of..."), however it may be useful to inform the user if an image is a photograph, illustration, chart, etc. if this information is considered central to the content of the image.

Linked Images

Difficulty: Advanced

Category:

- Content
- Presentation

When an image is the only thing contained within a link, that image must be given appropriate alternative text. Because the image provides a function, a screen reader encountering the image must convey some information to the user. If alternative text has not been provided for the image, a screen reader will typically read the image file name or the address of the page being linked to. Proper alternative text describing the link location is much more helpful and accurate. A screen reader will identify the presence of the link, so it is not necessary to put "link to" or similar in the alternative text.

Example:

```
<a href="http://www.apple.com/iphone"></a>
```

For more information, see <http://webaim.org/techniques/alttext/>.

Decorative Images

Difficulty: Advanced

Category:

- Content
- Presentation

In some cases images are used to convey a mood, feeling, or non-literal content - something to make the page more friendly, personable, or visually pleasing. It is often difficult to convey meaningful alternative text for such images in a way that is useful for someone that cannot see the image. Of course if the image conveys important content or has a function, it must have [alternative text](#). But for many of these more decorative images, it is often best to give them empty or null alternative text (`alt=""`) rather than try to convey the mood, feeling, or visual subtleties in alternative text.

Complex Images

Difficulty: Advanced

Category:

- Content
- Presentation

Important and relevant content from [complex images](#) must be provided in an accessible way. The alt attribute is generally reserved for a succinct alternative (less than around 100 characters is a general guideline). For images that require a longer description, this long description text may be provided directly on the main page in the context of the complex image (e.g., in a data table or descriptive caption immediately after the image). Alternatively, an image can be given a longdesc attribute. The longdesc value must contain the URL of a page that contains the long description (``). Due to very limited support in web browsers, longdesc is not suitable as the only method of access to the longer description page. Longdesc should be used in conjunction with a standard link to the long description page.

Frame Titles

Difficulty: Advanced

Category:

- Structure
- Validation

Frames (a way of combining multiple web pages into one visual presentation) are not used as frequently today, but when they are used, they must be identified with descriptive text. All frames must be given a descriptive title. The title should succinctly present the content and/or functionality of that frame.

```
<frameset>  
<frame title="navigation" src="nav.htm">  
<frame title="main content" src="content.htm">  
</frameset>
```

Headings

Difficulty: Advanced

Category:

- Structure
- Presentation

A logical [heading structure](#) greatly enhances the accessibility of a web page. They are used for page navigation and orientation. Each page should generally have one first-level heading (<h1>). While it is usually most logical to have the <h1> present the site name for the home page, for content pages the <h1> should typically contain the document title or main heading for that page. Heading levels should not be skipped, though it's acceptable to have lower level headings (such an <h2>'s) precede the <h1>, if the page structure requires it.

Fieldsets and Legends

Difficulty: Advanced

Category:

- Structure
- Presentation

Groups of checkboxes and radio buttons often have a higher-level description that indicates the purpose of the grouping. In such cases, a fieldset and legend should be used to visually identify the grouping and to ensure that the higher-level legend is read by assistive technologies.

```
<fieldset><legend>Choose your shipping method</legend>  
Radio buttons for shipping options  
</fieldset>
```

Fieldsets can be used for groupings of other types of form elements (e.g., "Shipping Address" vs. "Billing Address"), but should be used with care because the legend is identified for each form element in the fieldset.

For more information, see [Creating Accessible Form Controls](#)

Link Focus Indicators

Difficulty: Advanced

Category:

- Presentation

A link that has keyboard focus (accessed using the Tab key) is typically identified by a dotted line that surrounds the link. This dotted line is called the focus indicator or link outline. This focus indicator can be removed with `outline: 0` in CSS. Doing so will make it very difficult for sighted keyboard users to determine which element currently has keyboard focus as they navigate through the page. The default focus indicator can be enhanced by providing additional visual styles when the link receives focus. Typically, any style changes when the user

hovers their mouse over a link should be duplicated in the keyboard focus styles. All links should be clearly differentiated from surrounding text for visual users and keyboard-only users.

In your CSS, you can generally change

```
a:hover { ... }
```

to

```
a:hover, a:focus { ... }
```

to have all hover styles also appear on keyboard focus.

Title Attributes

Difficulty: Advanced

Category:

- Content
- Presentation

As defined in the HTML specification, the title attribute is for "advisory information". As such, it should not be relied on to provide accessibility. The title attribute is generally ignored by screen readers, except when used on form elements that do not have [labels, frames and iframes](#), and acronym and abbreviation (though many screen readers and screen reader settings ignore title for even these elements). The title attribute value can be used to present additional, supplementary information, but you cannot be sure it will be read. It is not visible to sighted keyboard-only users or to users who do not hover their mouse over the element (e.g., on touchscreen devices). Additionally, because title generates a browser tooltip when the mouse is placed over an element, it should be used with care and should not present redundant information that is already apparent visually in the page.

Data Tables

Difficulty: Advanced

Category:

- Structure
- Presentation

Because [data tables](#) are presented and navigated differently by screen readers than [layout tables](#), it is vital that data tables be marked up to support greater accessibility. Row and/or column headers should be identified as such (using the <th> tag) and given an appropriate scope designating that header as a column header or a row header (<th scope="col"> or <th scope="row">). For most tables, this markup will ensure that each data cell is programmatically associated with its appropriate row and/or column headers.

Table Captions

Difficulty: Advanced

Category:

- Structure
- Presentation

When a visual caption is provided that describes the content of a [data table](#), it should be marked up as a <caption> element. The <caption> element should be the first tag within the table. This associates the caption to the data table and allows it to be identified by assistive technologies.

Example:

```
<table>  
<caption>Fall Class Schedule</caption>  
<tr>  
...
```

Hidden Screen Reader Content

Difficulty: Advanced

Category:

- Content
- Presentation

By positioning content off the left side of the visible screen using CSS, it removes it from the visual presentation of the page, but allows it to still be read by a screen reader. Off-screen text should be used with care and generally only in cases where information or structure is apparent visually, but may not be readily accessible to a screen reader accessing it linearly. Content hidden with `display:none` or `visibility:hidden` in CSS is not read by a screen reader.

For more information, see [Invisible Content Just for Screen Reader Users](#)

User Style Sheets, Specified Colors, and Page Flexibility

Difficulty: Advanced

Category:

- Presentation

The end user always has ultimate control over what the page looks like. They can apply user style sheets to change font sizes, colors, layout, or any other property of the page. As much as possible, web pages should be designed to accommodate user customization. This means that the page has specified foreground and background colors for all elements. Each element need to be defined specifically due to the cascading nature of CSS, but at a minimum, the body element should have a foreground (color) and background (background-color) color specified in CSS. These colors must have sufficient contrast. This helps ensure that if the user overrides page colors, that each page element can be adequately overridden to remain readable. The page should also be designed so that page elements to not overlap or become invisible when text is increased or when the browser size is very small or very large. Bulletproof and flexible web design is an important aspect of usability and web accessibility.

JavaScript Events

Difficulty: Advanced

Category:

- Presentation

To ensure that [JavaScript](#) events can be triggered with both the mouse and the keyboard either a device independent event handler OR a mouse dependent event handler (such as `onmouseover`) and a keyboard dependent event handler (such as `onkeypress`) must be used. Keep in mind that some event handlers change their device dependence based on where they are used. For example, the `onclick` event handler is device independent (works with both the mouse and the keyboard), but only on elements that can receive keyboard focus (links and form elements that can be tabbed to).

Noscript and Progressive Enhancement

Difficulty: Advanced

Category:

- Presentation

The `noscript` element provides a fall-back for when JavaScript is not enabled or supported. Because most screen readers and other assistive technologies support JavaScript, and because users of these technologies are generally no more likely to have JavaScript disabled, the use of `noscript` is not really an accessibility concern. Developers must simply consider that if they require JavaScript, that some users may not get that content or functionality. It is important, however, that the functionality fail gracefully (e.g., it is clear or the user is alerted that a web application will not work without JavaScript).

Progressive enhancement is a better approach than `noscript`. `noscript` provides a fall-back for when JavaScript is disabled, whereas progressive enhancement is an approach where the web page is built to be fully functional without JavaScript, and is then enhanced with JavaScript. Using progressive enhancement ensures that the page works regardless of whether JavaScript is enabled.

ARIA

Difficulty: Advanced

Category:

- Structure

While support for [ARIA](#) (Accessible Rich Internet Applications) is not perfect, implementing it can never make a web application less accessible. While native accessibility should be implemented to the extent possible in HTML, ARIA allows enhanced semantics and accessibility that is not possible with HTML alone. Much of ARIA will be part of native HTML5, but because ARIA accessibility support is much more extensive than HTML5 accessibility support and because ARIA can be applied to HTML and XHTML pages, it makes the most sense to implement ARIA and take advantage of the enhanced accessibility it provides now.

ARIA Components

Difficulty: Advanced

Category:

- Structure

The core components of [ARIA](#) are roles, states, and properties. The dozens of attributes and role values possible in ARIA provide significantly enhanced semantics and accessibility detail over HTML alone. Basic ARIA attributes such as landmark roles, various roles to define the function of non-standard interactive elements (e.g., sliders, links used as buttons, etc.), `aria-required` for required form elements, `aria-labelledby` to enhance labeling for complex form elements, and so forth can be easily implemented now without any negative impact on functionality, presentation, or accessibility.

aria-labelledby

Difficulty: Advanced

Category:

- Structure

Traditional form labeling with the `<label>` has distinct limitations; you can only have one label for each form element. You cannot have one label for multiple form elements (consider "Telephone Number" as a label for 3 text boxes to comprise a complete telephone number). You also cannot have multiple labels for one form element (consider a text box that has a label preceding it and the word "required" immediately after it, both of which provide important information). The `aria-labelledby` attribute (part of the [ARIA](#) specification) overcomes these limitations by allowing any element to be labeled by any other page element.

Landmark roles

Difficulty: Advanced

Category:

- Structure

Adding landmark roles to your page is an easy way to enhance accessibility. Landmark roles are part of [ARIA](#), a new accessibility specification. Landmark roles define significant web page areas and provide the user quick access to them. The available landmark roles are application, banner, complementary, contentinfo, form, main, navigation, and search. You can simply add `role="search"` to your search form (`<form role="search">`) to identify it specifically as the search form on the page. `<div role="main">` designates the element that contains the main content. `<ul role="navigation">` could be used to specify the navigation items for a page. Landmark roles can easily provide significant functionality and increased accessibility to keyboard and screen reader users.

TabIndex

Difficulty: Advanced

Category:

- Structure

[Tabindex](#) values of 1 or greater assign a specific tab order to elements. Because the default tab order of the page is typically logical and intuitive, positive values are rarely necessary. A tabindex value of 0 allows elements other than links or form elements to be focusable in the default tab order of the document. Tabindex of -1 allow other elements to receive focus, but only when focus is set through JavaScript or via a link. Setting and modifying tabindex values and controlling keyboard focus with JavaScript can greatly increase the accessibility of dynamic and interactive web applications and widgets.

Semantic Structure

Difficulty: Advanced

Category:

- Structure
- Presentation

[Semantic structure](#) in HTML is very useful for accessibility. Headings, lists, tables, and other elements can be identified by screen readers and can be used for navigation. For example, a screen reader user can navigate by headings (and even by specific heading level), lists, buttons, links, etc. As such, it's important that HTML elements be used appropriately. Headings should only be used for headings, not simply to style text as big and bold. True lists should be used rather than simply adding bullet characters or numbers. Lists, on the other hand, should not be used to provide a visual indentation for content that is not truly a list.

Audio Descriptions

Difficulty: Advanced

Category:

- Content

Audio descriptions can enhance accessibility of video content for users who are blind or have other visual disabilities. Audio descriptions provide an auditory presentation of content that is

presented visually, but not presented audibly in video. They typically take the form of a narrator that describes the important information or content elements that sighted users can see. Audio descriptions are difficult and costly to produce. You typically must present two versions of the video. When producing video content, if you simply ensure that all visual content is presented audibly, then there will be no need for audio descriptions.

Scaling Images

Difficulty: Advanced

Category:

- Presentation

Users with low vision often use a screen enlarger to increase the size of elements in a web page. Images can become pixelated or blocky when they are enlarged. Text within images can become particularly difficult to read. When you can get the same visual presentation using real text, you should use it instead of text within images. When images that contain text are used, ensure that the text remains readable when the image is at least doubled in size. Using sans-serif fonts and adequate contrast makes enlarged text in images more readable.

Adobe PDF

Difficulty: Advanced

Category:

- Content
- Structure
- Presentation

[Adobe PDF](#) files can be made accessible. PDF files store accessibility information for screen readers in tags. Tags are text elements that are hidden within the PDF file, but that are presented to the screen reader. You must ensure that PDF files are authored in a way that

supports accessibility within these tags, and that the tags contain the full content of the PDF document. PDF accessibility is best implemented and modified using Adobe Acrobat Pro.