Disclaimer: This resource provides general guidance and is not legal advice. Consult qualified counsel for jurisdiction-specific requirements.

Screen Reader Optimization

Ensuring content is accessible to assistive technology users

Prepared by: AccessGuard

Version: 1.0

Date: October 18, 2025

Standards: WCAG 2.2 AA, Section 508, ARIA 1.2

Screen Reader Optimization

Table of Contents

[Back to TOC]

Screen readers are assistive technologies that read aloud the content of a webpage for users who are blind or have low vision. Optimizing your website for screen readers ensures that all users can access and interact with your content.

Understanding Screen Readers

[Back to TOC]

Popular Screen Readers

- **NVDA** (Windows, free): https://www.nvaccess.org/
- **JAWS** (Windows, paid): https://www.freedomscientific.com/
- **VoiceOver** (macOS/iOS, built-in): Enable in System Preferences
- **TalkBack** (Android, built-in): Enable in Settings
- **Narrator** (Windows, built-in): Enable in Settings

How Screen Readers Work

1. **Parse HTML** - Screen readers interpret semantic HTML

2. **Build Virtual DOM** - Create an accessible representation
3. **Announce Content** - Read content aloud to users
4. **Provide Navigation** - Allow users to navigate by headings, links, landmarks, etc.

Semantic HTML

Back to TOC

Use Native HTML Elements

❌ **Bad: Generic div with click handler**

```
<div onclick="handleClick()">Click me</div>
```

✅ **Good: Semantic button**

```
<button onclick="handleClick()">Click me</button>
```

Heading Structure

```
<h1>Page Title</h1>
  <h2>Section 1</h2>
    <h3>Subsection 1.1</h3>
    <h3>Subsection 1.2</h3>
  <h2>Section 2</h2>
    <h3>Subsection 2.1</h3>
```

Screen reader navigation: Users can jump between headings using keyboard shortcuts (H key in NVDA/JAWS).

Landmarks

```
<body>
  <header role="banner">
    <nav role="navigation" aria-label="Main navigation">
      <!-- Main navigation -->
    </nav>
  </header>

  <main role="main">
    <article>
      <!-- Main content -->
    </article>
  </main>

  <aside role="complementary">
    <!-- Sidebar content -->
  </aside>

  <footer role="contentinfo">
    <!-- Footer content -->
  </footer>
</body>
```

Screen reader navigation: Users can jump between landmarks using keyboard shortcuts (D key in NVDA/JAWS).

ARIA Labels and Descriptions

Back to TOC

Accessible Names

aria-label - Provides accessible name

```
<button aria-label="Close dialog">×</button>
```

aria-labelledby - References element(s) that provide the name

```
<h2 id="dialog-title">Confirm action</h2>
<div role="dialog" aria-labelledby="dialog-title">
  <!-- Dialog content -->
</div>
```

aria-describedby - References element(s) that provide additional description

```
<input
  type="email"
  aria-describedby="email-help"
>
<span id="email-help">We'll never share your email</span>
```

Roles

```
<!-- Navigation -->
<nav role="navigation" aria-label="Main navigation">

<!-- Main content -->
<main role="main">

<!-- Search -->
<form role="search">

<!-- Banner -->
<header role="banner">

<!-- Contentinfo -->
<footer role="contentinfo">

<!-- Complementary -->
<aside role="complementary">
```

```
<!-- Form -->
<form role="form">

<!-- Article -->
<article role="article">

<!-- Region -->
<section role="region" aria-label="Product features">
```

States and Properties

```
<!-- Expanded/Collapsed -->
<button
  aria-expanded="false"
  aria-controls="menu"
>
  Menu
</button>
<ul id="menu" hidden>...</ul>

<!-- Selected -->
<button
  role="tab"
  aria-selected="true"
>
  Overview
</button>

<!-- Checked -->
<input
  type="checkbox"
  aria-checked="true"
>

<!-- Required -->
```

```
<input
  type="email"
  aria-required="true"
>

<!-- Invalid -->
<input
  type="email"
  aria-invalid="true"
  aria-describedby="email-error"
>

<!-- Live Regions -->
<div role="status" aria-live="polite">
  Form submitted successfully
</div>

<div role="alert" aria-live="assertive">
  Error: Invalid email address
</div>
```

Images and Media

Back to TOC

Alternative Text

Informative images - Describe the content

```
<img src="chart.png" alt="Sales increased 25% in Q4 2024">
```

Decorative images - Use empty alt

```
<img src="divider.png" alt="">
```

Complex images - Provide long description

```
<img
  src="flowchart.png"
  alt="Process flowchart"
  longdesc="flowchart-description.html"
>

<!-- Or use aria-describedby -->
<img
  src="flowchart.png"
  alt="Process flowchart"
  aria-describedby="flowchart-desc"
>
<div id="flowchart-desc">
  The process begins with user input, which is validated...
</div>
```

Icons

Icon buttons - Provide accessible name

```
<button aria-label="Close">
  <svg aria-hidden="true">...</svg>
</button>
```

Decorative icons - Hide from screen readers

```
<span class="icon" aria-hidden="true">✓</span>
```

```
<span>Completed</span>
```

Forms

Back to TOC

Labels

```
<!-- Explicit label -->
<label for="email">Email address</label>
<input type="email" id="email" name="email">

<!-- Implicit label -->
<label>
  Email address
  <input type="email" name="email">
</label>

<!-- ARIA label (use sparingly) -->
<input type="email" name="email" aria-label="Email address">
```

Error Messages

```
<label for="email">Email address</label>
<input
  type="email"
  id="email"
  name="email"
  aria-invalid="true"
  aria-describedby="email-error"
>
<span id="email-error" role="alert">
```

```
  Please enter a valid email address
</span>
```

Required Fields

```
<label for="email">
  Email address
  <span aria-label="required">*</span>
</label>
<input
  type="email"
  id="email"
  name="email"
  required
  aria-required="true"
>
```

Dynamic Content

Back to TOC

Live Regions

Polite - Announces when user is idle

```
<div role="status" aria-live="polite" aria-atomic="true">
  Form submitted successfully
</div>
```

Assertive - Announces immediately

```
<div role="alert" aria-live="assertive" aria-atomic="true">
  Error: Invalid email address
```

```
</div>
```

Off - Does not announce

```
<div role="status" aria-live="off">
  <!-- Content updates silently -->
</div>
```

Loading States

```
<button aria-busy="true" aria-label="Submitting form...">
  Submit
</button>

<!-- Or use aria-live -->
<div role="status" aria-live="polite">
  <span class="sr-only">Loading</span>
  <div class="spinner" aria-hidden="true"></div>
</div>
```

Tables

Back to TOC

Basic Table

```
<table>
  <caption>Monthly Sales Report</caption>
  <thead>
    <tr>
      <th scope="col">Month</th>
      <th scope="col">Sales</th>
```

```
      <th scope="col">Growth</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">January</th>
      <td>$10,000</td>
      <td>+5%</td>
    </tr>
    <tr>
      <th scope="row">February</th>
      <td>$12,000</td>
      <td>+20%</td>
    </tr>
  </tbody>
</table>
```

Complex Table

```
<table>
  <caption>Employee Directory</caption>
  <thead>
    <tr>
      <th scope="col" rowspan="2">Name</th>
      <th scope="colgroup" colspan="2">Contact</th>
      <th scope="col" rowspan="2">Department</th>
    </tr>
    <tr>
      <th scope="col">Email</th>
      <th scope="col">Phone</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">John Doe</th>
      <td>john@example.com</td>
```

```
      <td>555-0100</td>
      <td>Engineering</td>
    </tr>
  </tbody>
</table>
```

Skip Links

Back to TOC

```
<a href="#main-content" class="skip-link">
  Skip to main content
</a>

<header>
  <nav>...</nav>
</header>

<main id="main-content">
  <!-- Main content -->
</main>

<style>
.skip-link {
  position: absolute;
  top: -40px;
  left: 0;
  background: #000;
  color: #fff;
  padding: 8px;
  text-decoration: none;
  z-index: 100;
}

.skip-link:focus {
  top: 0;
```

```
}
</style>
```

Screen Reader Only Content

Back to TOC

```
.sr-only {
  position: absolute;
  width: 1px;
  height: 1px;
  padding: 0;
  margin: -1px;
  overflow: hidden;
  clip: rect(0, 0, 0, 0);
  white-space: nowrap;
  border-width: 0;
}
```

```
<button>
  <svg aria-hidden="true">...</svg>
  <span class="sr-only">Close dialog</span>
</button>
```

Testing with Screen Readers

Back to TOC

NVDA (Windows, Free)

Keyboard shortcuts:

- **H** - Jump to next heading

- **Shift+H** - Jump to previous heading
- **L** - Jump to next list
- **F** - Jump to next form field
- **B** - Jump to next button
- **D** - Jump to next landmark
- **Insert+F7** - Open elements list

JAWS (Windows, Paid)

Keyboard shortcuts:

- **H** - Jump to next heading
- **Shift+H** - Jump to previous heading
- **Insert+F7** - Open elements list
- **Insert+Ctrl+Home** - Jump to top of page

VoiceOver (macOS/iOS)

Keyboard shortcuts:

- **VO+Right/Left** - Navigate by item
- **VO+U** - Open rotor
- **VO+H** - Jump to next heading
- **VO+Shift+H** - Jump to previous heading
- **VO+L** - Jump to next list

Testing Checklist

Back to TOC

Content Structure

- [ ] Heading hierarchy is logical
- [ ] Landmarks are properly labeled
- [ ] Lists use semantic HTML
- [ ] Tables have headers and captions

Interactive Elements

- [ ] Buttons have accessible names

- [ ] Links have descriptive text
- [ ] Form inputs have labels
- [ ] Custom components have ARIA attributes

Dynamic Content

- [ ] Status messages are announced
- [ ] Error messages are announced
- [ ] Loading states are communicated
- [ ] Modal dialogs are announced

Images and Media

- [ ] Images have appropriate alt text
- [ ] Decorative images use empty alt
- [ ] Complex images have long descriptions
- [ ] Icons have accessible names

Testing

- [ ] Tested with NVDA or JAWS
- [ ] Tested with VoiceOver
- [ ] Keyboard navigation works
- [ ] All content is announced correctly

Common Mistakes

Back to TOC

❌ Don't

- Use generic divs for interactive elements
- Hide content with `display: none` or `visibility: hidden`
- Use color alone to convey information
- Create inaccessible custom components
- Forget to test with actual screen readers

✅ Do

- Use semantic HTML elements
- Provide alternative text for images
- Use ARIA attributes appropriately
- Test with screen readers regularly
- Provide skip links and landmarks

Resources

Back to TOC

- **WCAG 2.2 Guidelines**: https://www.w3.org/WAI/WCAG22/quickref/
- **ARIA Authoring Practices**: https://www.w3.org/WAI/ARIA/apg/
- **WebAIM Screen Reader Testing**: https://webaim.org/articles/screenreader_testing/
- **NVDA Documentation**: https://www.nvaccess.org/about-nvda/
- **JAWS Documentation**: https://www.freedomscientific.com/support/jaws-documentation/