

Keep these standard guidelines in mind as you work on the webpage.

1.3.1 Info and Relationships

Landmark roles in webpage design provide programmatic access, aiding navigation for users of assistive technology. By identifying sections, like common navigation menus, they facilitate easy skipping and navigation, enhancing user experience for those with disabilities.

Example 1: Simple landmarks

```
<div role="banner"> site logo and name, etc. here </div>
<div role="search"> search functionality here </div>
<div role="navigation"> a list of navigation links here </div>
<div role="form"> a sign-up form here </div>
<div role="main"> the page's main content here </div>
<div role="region"> a sponsor's promotion here </div>
<div role="complementary"> sidebar content here </div>
<div role="contentinfo"> site contact details, copyright information, etc. here </div>
```

Example 2: Multiple landmarks of the same type and aria-labelledby

```
<div aria-labelledby="site-nav-heading" role="navigation">
<h2 id="site-nav-heading">Site</h2>
<ul>
<li><a href="#">nav link 1</a></li>
<li><a href="#">nav link 2</a></li>
<li><a href="#">nav link 3</a></li>
</ul>
</div>
<div aria-labelledby="related-nav-heading" role="navigation">
<h2 id="related-nav-heading">Related Topics</h2>
<ul>
<li><a href="#">topic link 1</a></li>
<li><a href="#">topic link 2</a></li>
<li><a href="#">topic link 3</a></li>
</ul>
</div>
```

Example 3: Search form

```
<form role="search">
<label for="product-search" id="search-label">Search</label>
<input id="product-search" placeholder="title, author, or ISBN" type="text">
<button type="submit">Find Books</button>
</form>
```

Best Practices for Implementing Links and Buttons in Web Design

Avoid emulating links and buttons. Use the `a` and `button` tags appropriately. Avoid using `a` tags for buttons. **Avoid using div, span, etc. tags for links or buttons.**

```
<!-- Correct usage of a link -->
<a href="https://example.com">Visit Example</a>
```

```
<!-- Correct usage of a button -->
<button type="button">Click Me</button>
```

1.3.5 Identify Input Purpose

If a form field asks for information about the user and if there is an appropriate HTML `autocomplete` attribute, include that `autocomplete` attribute.

```
<form autocomplete="on">
<!-- Input fields with autocomplete -->
<label for="country">Country:</label>
<input type="text" id="country" name="country" autocomplete="country"><br>

<label for="city">City:</label>
<input type="text" id="city" name="city" autocomplete="city"><br><br>

<!-- Submit button -->
<input type="submit" value="Submit">
</form>
```

1.4.1 Use of Color

Links should always be easily identifiable through non-color means, including both default and hover states. The easiest and most conventional way to signify links is underlining.

```
<body>
<h1>Styled Links Example</h1>
<p>
This is an example paragraph with a <a href="#">styled link</a>. Hover over the link to see the color change and underline removal.
</p>
</body>
```

```
<style>
/* Styling links */
/* Default state */
a {
    text-decoration: underline; /* Add underlines to links */
    color: blue; /* Set link color */
}

/* Hover state */
a:hover {
    text-decoration: none; /* Remove underline on hover */
    color: red; /* Change link color on hover */
}
</style>
```

2..4.4 Link Purpose (In Context)

If the visible text alone is not sufficient to convey meaning, use advanced techniques to provide additional meaning, such as ARIA attributes, screen reader only text, or the `title` attribute.

1. ARIA attributes:

```
<button aria-label="Close">X</button>
```

2. Title attributes:

```
<a href="#" title="Click here to learn more">Learn more</a>
```

3.1.2 Language of Parts

If a portion of the page is in a different language, use the `lang` attribute on that part.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Example Page</title>
</head>
<body>
<h1>Welcome to My Website</h1>
<p>This is a paragraph in English.</p>

<div lang="fr">
<h2>Bienvenue sur mon site web</h2>
<p>Ceci est un paragraphe en français.</p>
</div>
</body>
</html>
```

3.3.1 Error Identification

Identify errors using `aria-invalid`.

```
<input type="text" id="username" aria-invalid="true" aria-describedby="username-error">
<div id="username-error">Please enter a valid username.</div>

or , add the validation .
```


3.3.2 Labels or Instructions

When providing inline help text, use `aria-describedby`(or `placeholder`) to associate the help text with the input.

```
<label for="email">Email:</label>
<input type="email" id="email" aria-describedby="email-help">
<span id="email-help">Please enter your email e.g; example@example.com</span>
```

Example 2

```
<input type="email" id="email" placeholder="Enter email e.g; example@example.com">
```

4.1.2 Name, Role, Value

Use ARIA to enhance accessibility only when HTML is not sufficient. Use caution when providing ARIA roles, states, and properties.

```
<label for="name" >Name:</label>
<input id="name"
name="name"
type="text"
role="textbox"
aria-label="Name"
aria-required="true"
value="">
```