HTML Accessibility Review ✔

Introduction to Accessibility

- **Web Content Accessibility Guidelines**: The Web Content Accessibility Guidelines (W content more accessible to people with disabilities. The four principles of WCAG are Understandable, and Robust.

Assistive Technology for Accessibility

- **Screen readers**: Software programs that read the content of a computer screen out l visually impaired to access the web.
- **Large text or braille keyboards**: Used by people with visual impairments to access th
- **Screen magnifiers**: Software programs that enlarge the content of a computer scree access the web.
- **Alternative pointing devices**: Used by people with mobility impairments to access th trackballs, and touchpads.
- **Voice recognition**: Used by people with mobility impairments to access the web. It al

Accessibility Auditing Tools

- **Common Accessibility Tools**: Google Lighthouse, Wave, IBM Equal Accessibility Che accessibility tools used to audit the accessibility of a website.

Accessibility Best Practices

- **Proper heading level structure**: You should use proper heading levels to create a log using assistive technologies understand the content of your website.
- **Accessibility and Tables**: When using tables, you should use the `th` element to defir cells. This helps people using assistive technologies understand the structure of the t the caption (or title) of a table, so users, especially those who use assistive technolog content. You should place the `caption` element immediately after the opening tag o and other assistive technologies can provide more context by announcing the captio
- **Importance for inputs to have an associated label**: You should use the `label` eleme people using assistive technologies understand the purpose of the input.

assistive technologies understand the content of the image.

- **Importance of good link text**: You should use descriptive link text to help users unde using assistive technologies understand the purpose of the link.
- **Best practices for making audio and video content accessible**: You should provide c to make it accessible to people with hearing impairments. You should also provide au accessible to people with visual impairments.
- **`tabindex` attribute**: Used to make elements focusable and define the relative orde keyboard. It is important to never use the `tabindex` attribute with a value greater t -1. For more information, review the prior lecture video on keyboard accessibility.

```
<p tabindex="-1">Sorry, there was an error with your submission.</p>
```

- **`accesskey` attribute**: Used to define a keyboard shortcut for an element. This can h website more easily.

```
<button accesskey="s">Save</button>
<button accesskey="c">Cancel</button>
<a href="index.html" accesskey="h">Home</a>
```

WAI-ARIA, Roles, and Attributes

- **WAI-ARIA**: It stands for Web Accessibility Initiative - Accessible Rich Internet Appli HTML elements to improve accessibility. It provides additional information to assisti the content.
- **ARIA roles**: A set of predefined roles that can be added to HTML elements to define technologies understand the content of the website. Examples include `role="tab"`

There are six main categories of ARIA roles:

- **Document structure roles**: These roles define the overall structure of the web page. understand the relationships between different sections and help users navigate the
- **Widget roles**: These roles define the purpose and functionality of interactive elemer
- **Landmark roles**: These roles classify and label the primary sections of a web page. Sc navigation to important sections of a page.
- **Live region roles**: These roles define elements with content that will change dynamic technologies can announce changes to users with visual disabilities.
- **Window roles**: These roles define sub-windows, like pop up modal dialogues. These
- **Abstract roles**: These roles help organize the document. They're only meant to be us you should know that they exist but you shouldn't use them on your websites or web

which helps people using assistive technology (such as screen readers) understand th
when the visual label for an element is an image or symbol rather than text. `aria-la`
attribute while `aria-labelledby` allows you to reference existing text on the page

```
<button aria-label="Search">
    <i class="fas fa-search"></i>
</button>
```

```
<input type="text" aria-labelledby="search-btn">
<button type="button" id="search-btn">Search</button>
```

- **`aria-hidden` attribute**: Used to hide an element from assistive technologies such a
  hide decorative images that do not provide any meaningful content.

```
<button>
    <i class="fa-solid fa-gear" aria-hidden="true"></i>
    <span class="label">Settings</span>
</button>
```

- **`aria-expanded` attribute**: Used to convey the state of a toggle (or disclosure) featu

```
<button aria-expanded="true">Menu</button>
```

- **`aria-live` attribute**: Used to indicate that an element's content is important enou
  should be announced immediately to screen reader users. This can include status me
  returned from a search, or an error message displayed after an unsuccessful form sul

```
<div aria-live="assertive">
    <p>Your session will expire in 30 seconds.</p>
</div>
```

```
<div aria-live="polite">
    <p>File successfully uploaded</p>
</div>
```

These attributes can be used to indicate the state of an element and help people usi
the website.

- **`aria-haspopup` attribute**: This state is used to indicate that an interactive element
can only use the `aria-haspopup` attribute when the pop-up has one of the followin
`dialog`. The value of `aria-haspopup` must be either one of these roles or `true`,

```
<button id="menubutton" aria-haspopup="menu" aria-controls="filemenu"
<ul id="filemenu" role="menu" aria-labelledby="menubutton" hidden>
  <li role="menuitem" tabindex="-1">Open</li>
  <li role="menuitem" tabindex="-1">New</li>
  <li role="menuitem" tabindex="-1">Save</li>
  <li role="menuitem" tabindex="-1">Delete</li>
</ul>
```

- **`aria-checked` attribute**: This attribute is used to indicate whether an element is in
creating custom checkboxes, radio buttons, switches, and listboxes.

```
<div role="checkbox" aria-checked="true" tabindex="0">Checkbox</div>
```

- **`aria-disabled` attribute**: This state is used to indicate that an element is disabled
screen readers.

```
<div role="button" tabindex="-1" aria-disabled="true">Edit</div>
```

- **`aria-selected` attribute**: This state is used to indicate that an element is selected
tabbed interface, a listbox, or a grid.

```
<div role="tablist">
    <button role="tab" aria-selected="true">Tab 1</button>
    <button role="tab" aria-selected="false">Tab 2</button>
    <button role="tab" aria-selected="false">Tab 3</button>
</div>
```

- **`aria-controls` attribute**: Used to associate an element with another element that
technologies understand the relationship between the elements.

```
    <button role="tab" id="tab1" aria-controls="section1" aria-selec
        Tab 1
    </button>
    <button role="tab" id="tab2" aria-controls="section2" aria-selec
        Tab 2
    </button>
    <button role="tab" id="tab3" aria-controls="section3" aria-selec
        Tab 3
    </button>
</div>
```

- **`aria-describedby` attribute**: Used to provide additional information about an ele
  contains the information. This gives people using screen readers immediate access t
  the element. Common usage would include associating formatting instructions to a t
  invalid form submission.

```
<form>
    <label for="password">Password:</label>
    <input type="password" id="password" aria-describedby="password-
    <p id="password-help">Your password must be at least 8 character
</form>
```

Assignment

Review the HTML Accessibility topics and concepts.

Please complete the assignment

Submit

Ask for Help