Applied Accessibility

Created By	Nate L
Last Edited	@Jan 2, 2019 2:17 PM

- The alt attribute on the img tag describes the content of the image and provides a text-alternative.
    - This helps in case the image fails to load or can't be seen by a user
    - It's also used by search engines to understand what an image contains to include it in search results
    - Those with visual impairments rely on screen readers to access the alt attribute to read the contents to deliver key information
    - Good alt text is short but descriptive and briefly conveys the meaning of an image
    - HTML5 specifications consider this mandatory
    - Images grouped with captions or are for decoration only don't need alt text
    - If the image is already explained or adds no meaning to the page, it can have an empty alt attribute
    - Background images are usually considered as decorative, however they are usually applied with CSS rules
- Screen readers can be set to read only the headings on a page to get a summary of the content, heading tags `h1` through `h6` elements help provide structure and label said content
    - It is important to provide semantic meaning to your markup, headings should relate to each other. You shouldn't skip headings, like putting h5 directly under h2.
        - If you want to change the font size of an element be sure to use CSS instead
        - You should only have one h1 element on a given page
        - Headings are used in part by search engines to understand the topic of the page
- HTML 5 introduced more semantic elements such as `main`, `header`, `footer`, `nav`, `article`, and `section`, etc.
    - The main `<main></main>` element represents the dominant content of the body of a document, it should consist of content that is directly related to or expands upon the central topic of a document or the central functionality of an application
        - It goes *between* the header and footer, and does not include them
    - The header `<header></header>` element represents the introductory content, typically a group of introductory or navigational aids. It can contain heading text and other elements

    such as a logo, search form, author name, etc.

  - The footer `<footer></footer>` element contains information about the author of the section, copyright data or links to related documents
  - The nav `<nav></nav>` element represents a section of a page whose purpose is to provide navigation links, either within the document or externally. Common use cases of `nav` sections include menus, tables of contents, and indexes.
    - Usage:
      - `nav` is intended only for major block of navigation links, not all links have to be contained in a `nav` element. The footer element has a list of links that don't need to be in a `nav` element
      - A document may have several `nav` elements, ex: one for internal navigation and another for intra-page navigation. The `aria-labelledby` attribute can be used to promote accessibility
        - `aria-labelledby` contains the element IDs of labels in objects such as input elements, widgets, and groups; the attribute establishes relationships between objects and their labels; assistive technologies use this attribute to catalog objects in a document so that users can navigate between them, without it, the assistive technology cannot catalog the object
      - User agents, such as screen readers targeting disabled users, can use this element to determine whether to omit the initial rendering of navigation-only content.
  - The article `<article></article>` element represents a self-contained composition in a document, page, application, or site, which is intended to be independently distributable or reusable.
    - Each `<article>` should be identified typically by including a heading `<h1>-<h6>` as a child of the article element
    - When an `<article>` is nested, the inner element represents an article related to the outer element
    - Author information in an <article> can be provided through the `<address></address>` element
    - Publication date and time of an `<article>` can be described using the datetime attribute of a `<time>` element
    - Use cases:
      - Forum post, a magazine or newspaper article or a blog entry
- The section `<section></section>` element represents a standalone section which doesn't have a more specific semantic element to represent it; typically sections have a heading
  - Usage:

        - Each `<section>` should be identified, typically by including a heading `<h1>-<h6>` as a child of the element
        - Do not use `<section>` as a generic container, that's what a `<div>` is for, especially when the sectioning is only for styling purposes.
        - Rule of thumb: a section should logically appear in the outline of a document
        - You can have sections within articles
- The audio `<audio></audio>` element is used to embed audio content and provide controls to access said audio (the "control" attribute does not need a value, but needs to be included)
    - Include a `<source>` element which identifies the audio source and its type
    - Have a transcript or related text nearby for those who cannot hear the audio
- The figure `<figure></figure>` element wraps figures and their captions
    - The caption itself should be wrapped by a `<figcaption></figcaption>` element
    - The image still needs alt text, but it should naturally be similar to the caption text
- Input fields on a form usually each have text associated with them, and this text should be wrapped in the label `<label></label>` element (Note: the text being referred to here isn't the placeholder text)
    - Each input choice should have its own label element and input element pair
    - This applies to all types of forms, including radio, checkbox, and text field forms
    - The label element should have a "for" attribute equal to the id of the input element
        - This is how screen readers are able to associate an input's descriptive text with the input element itself
- The fieldset `<fieldset></fieldset>` element is used to group several controls as well as `<label>` elements within a web form
    - The legend `<legend></legend>` represents a caption for the content of its parent `<fieldset>`, it provides a description for the grouping which is read by screen readers for each choice in the `<fieldset>` element
        - Fieldset and legend aren't necessary when the choices are self explanatory
            - Using a label with the for attribute for each input (radio button, etc) is sufficient
    - example of usage:

```
 <form>
  <fieldset>
    <legend> Choose an item </legend>
    ... inputs/labels...
  </fieldset>
</form>
```

- Setting input type to "date" will produce an interactive calendar
  - Having some text to inform the user of the expected date format is a good idea, since older browsers do not support the calendar feature
- Wrapping inline dates with the time `<time></time>` element and setting the datetime attribute to a standardized date format (YYYY-MM-DD) helps screen readers identify dates

```
<p>Let's schedule for <time datetime="2013-02-13">next Wednesday</time>, since we're both free then.</p>
```

- CSS can make elements only visible to screen readers
  - Example usage: screen readers need a table version of a graphic, but you want to hide the table from general audiences
  - The following rule will accomplish this

  ```
  .sr-only {
      position: absolute;
      left: -10000px;
      width: 1px;
      height: 1px;
      top: auto;
      overflow: hidden;
    }
  ```

- Considering contrast ratio is important. The recommended minimum contrast ratio is 4.5:1
  - This becomes more complex when you consider colors
  - There are tools to help with this: https://webaim.org/resources/contrastchecker/
- Avoid using similar colors on top of each other for colorblind users
- Hyperlink text should describe whatever it's linking to

- The accesskey attribute lets users jump to certain elements using a keyboard shortcut
- The tabindex attribute lets users tab to an element. Its value determines the tab order.
  - Input elements are automatically tabbable