PART A

1. A DOCTYPE specifies the type and version of html or xhtml used in page. It ensures that browsers understand the language being used, whether it is HTML5, HTML4, or another version. The purpose <! DOCTYPE html > acts like a web language tag, ensuring that your page speaks the same language as the browsers, avoiding confusion and ensuring a smooth viewing experience
2. a) The <header> element is a structural element that outlines the heading of a segment of a page.it falls within the body element

b) The <nav> element identifies the section for content such as global navigation, a table of content, previous/next or other noteworthy groups.

c) The <section> element is used to identify a thematic grouping of content, which generally, but not always, includes a heading.

d) The <article> element identifies a section of independent, self-contained content that may be independently distributed or reused.

c) The <aside> element holds content, such as sidebars, inserts, or brief explanations that is tangentially related to the content surrounding it.

d) The <footer> element identifies the closing or end of a page, article, section, or other segment of a page. Generally, the footer element is found at the bottom of its parent.

|  |  |
| --- | --- |
| <section> element | <article> element |
| Defines distinct sections within a document. These sections can represents chapters ,headers ,footers or any other logical divisions | Specifies independent , self-contained content |
| It helps organize the content by grouping related information together | An article should make sense of its own even if removed from the rest of the site |
| Think of it as a container for a specific topic or theme within a larger | It represents a complete ,standalone piece of content that can be distributed independently |
|  |  |

1. To create a hyperlink we us the <a href= “ “ target=” -blank></a>

For instance to open a new page in your site.

<a href = “ #index” target = “\_\_ blank “ >Home</a>

1. Accessibility

* Semantic HTML improves accessibility for all users, including those with disabilities

2. SEO (search engine optimization)

* Search engine favor sematic HTML because it provides clear context and meaning.
* This can positively impact a website’s ranking in search results.

3. Readability and maintainability.

* Semantic HTML makes code more readable and understandable for developers

4. Interoperability

* Semantics follows standardize naming conventions.

**ESSAY**

**TOPIC 1.**

**Alt attribute in < img > tags**

* The alt attribute in <img> tags is crucial for both accessibility and SEO.

**Accessibility**

* The alt attribute provides alternative text that describes the image’s content.
* Screen readers use this text to conveys the images meaning to users with visual impairments.
* If an image fails to load , the alt text displays in its place.
* Example
* <img src = “pancakesstack.jpg “ alt = “syrup being poured over a stack of pancakes”>

SEO

* Search engines index the alt text to understand the images context
* Properly descriptive alt text can improve your sites ranking in image search results.tags
* Howevr ,avoid oversuing keywords; focus on accurate descriptions
* Example
* < img src = “ logo.png” alt = “ Company logo” >

Header tags ( <h1> , <h2> , etc ) for SEO:

* Header tags provides structures and context for your content
* They help both users and search engine understand our page.
* Hierarchy : < h1> (main topic) , h2 (subheadings), < h3 > to <h6> (additional subheadings)

**ARIA (Accessible Rich internet Applications)**

ARIA stands for **Accessible Rich Internet Applications**. It's a set of attributes that you can add to HTML elements to make web content and applications more accessible for people with disabilities, especially those who rely on assistive technologies (AT) like screen readers.

Regular HTML offers a good foundation for accessibility, but it can sometimes fall short for complex elements or dynamic content. ARIA bridges this gap by providing a way to communicate additional information to ATs about the purpose, state, and properties of these elements.

Here is an example of how ARIA attributes can improve accessibility:

Imagine a search bar on a webpage. By default, a screen reader might just announce it as an <input> element, which isn't very informative. But by adding the aria-label attribute with the text "Search for products" to the input element, you can clearly tell the screen reader what the element's function is. This makes it much easier for users with visual impairments to navigate and interact with your webpage.

There are many different ARIA attributes available, each serving a specific purpose. Some common ones include:

* aria-label: Adds a textual label to an element
* aria-labelledby: References another element that labels the current one
* aria-describedby: Describes the element with another element's text content
* aria-hidden: Indicates if an element is hidden or not visible
* aria-disabled: Specifies if an element is disabled

**Topic 2.**

**1.** The key difference between the <img> and <figure> elements in HTML lies in their purpose:

* <img> (image): This tag is specifically designed to **insert an image** into your web page. It's a simple and straightforward way to display visual content.
* <figure> (figure): This element acts as a **container** for self-contained content that complements the main flow of your text. It's often used for figures, diagrams, charts, or illustrations, but it can also hold images. The <figure> element provides a semantic meaning to the content it surrounds.

Here's how you can use <figure> with <figcaption> to add captions to your images:

1. **Wrap the image with the** <figure> **tag:** This creates a container for the image and any associated caption.
2. **Add the caption using the** <figcaption> **tag:** Place this tag **inside** the <figure> element. The text within <figcaption> will appear as a caption below the image.

**Example:**

HTML

<figure>

<img src="image.jpg" alt="Description of the image">

<figcaption>A caption describing the image content.</figcaption>

</figure>

This code will display an image and its corresponding caption. Screen readers and other assistive technologies will recognize the <figure> and <figcaption> elements, making the content more accessible for users with disabilities.

**Additional benefits of using** <figure>**:**

* **Semantic meaning:** It conveys the relationship between the image and the content, improving SEO (Search Engine Optimization) and accessibility.
* **Styling flexibility:** You can style the <figure> and <figcaption> elements independently for a more visually appealing layout.

While the <img> tag gets the job done for displaying images, the <figure> element offers a semantic and accessible way to incorporate images with captions into your webpages.

2. While the <iframe> tag can be used to embed videos in a web page, it's generally not the recommended approach for video playback. Here's why:

* **Accessibility:** <iframe> doesn't provide as much control over accessibility features like captions and keyboard navigation compared to the dedicated <video> element.
* **Limited control:** You have less control over the video playback behavior with <iframe>.

For better video integration, use the element:

1. **Basic video embed:**

HTML

<video src="your\_video.mp4" controls></video>

This code snippet inserts a video element with playback controls. Replace "your\_video.mp4" with the actual path to your video file.

1. **Specifying video format:**

The <video> element supports multiple video formats. You can use the type attribute to specify the format:

HTML

<video src="your\_video.mp4" type="video/mp4" controls></video>

Here, we explicitly define the video type as video/mp4. It's a good practice to include this for wider browser compatibility.

**Fallback content for unsupported browsers:**

Browsers that don't support the <video> element or the specific video format you're using will not be able to display the video. To provide an alternative for these users, you can use the <source> element within the <video> tag:

**<video controls>**

**<source src =**“lecture.mp4”**type="video/mp4"> <source src="**lecture.mp4**.webm" type="video/webm"> Your browser does not support the video tag.**

**</video>**

**3. Using the** data-\* **attribute:**

1. **Structure:** The attribute name always starts with data- followed by a hyphen (-) and then your custom attribute name.
2. **Data type:** The data value can be a string, number, or even JSON (JavaScript Object Notation) for more complex data structures.

**Example:**

HTML

<img src="product.jpg" alt="Product image" data-product-id="123" data-price="9.99">

In this example, we've added two custom data attributes to an image element:

* data-product-id: Stores the product ID as "123"
* data-price: Stores the product price as "9.99"

**Benefits of using** data-\* **attributes:**

* **Flexibility:** You can store any kind of information specific to your needs.
* **Separation of concerns:** Keeps presentational markup clean by separating data from HTML structure.