# SUMMARY FROM UNIT 1-4

# HTML

**HTML syntax**

Under this topic I have learnt that HTML is a language that is used to structure webs.

It also has two tags which are the opening tag <p> and the closing tag </P>.

These topics explore HTML elements and their usage, aiming to provide meaning to content and aid computer understanding. It also demonstrates for us how elements can be nested within one another, such as paragraphs, which cam emphasized using <p> tags and <em> tags.

**HTML HEADLINES**

Headlines in HTML come in six types: h1, h2, h3, h4, h5 and h6. Each has a distinct visual effect and hierarchy. The largest is h1, while the is the h6. Selecting the appropriate headline level is straightforward, starting with the opening tag and ending with the closing tag.

Hierarchical headlines provide meaning to the browser and ensure consistency for screen reader users. Follow use cases for each level of headline and consider how they will be utilized across the site. Longe teams often have design systems for sematic design.

**HTML BOLD AND ITALIC**

The use of italics in HTML is used to make a strong point or visually distinguish between two use case. The word “Favorite’+ is italicized to emphasize a specific character, while ‘’Sesame Street” is italicized to highlight the title of the TV show.

The browser should understand when to verbally emphasize or visually distinguish these two cases. Two elements are used to convey this distinction: the <i> element for visual italic and the <em> element for adding emphasis.

These elements serve different purposes, communicating semantic and human meaning.

The ‘’<strong>’’ element emphasizes importance, seriousness, or urgency, while the ‘’<b>’’element is more genetic and neutral, allowing for bold visuals without implying on alternative voice or mood.

**HTML LISTS**

In html there are three types of lists: unordered lists, ordered lists and definition lists.

Unordered lists are commonly used in web design, such as a recipe list. Each item is enclosed in an <il> element, with a dot or marker before it. The entire list is wrapped in an <ul> element. Indentation can be added for readable code, but the default appearance is determined by the browser.

An ordered list is a list with a specific order of items. If differs from on unordered list in that t uses <ol> to wrap the list items, displaying numbered steps.

A definition list in html is a key-value pair in computer science, consisting of terms and their corresponding description, and is wrapped in a tag, representing the entire list. Multiple descriptions can be added, and the tags are placed side by side.

**HTML QUOTES**

Html elements like paragraphs, headlines, bold and italic styles, and listed help format text on webpages. However, including a quote can be challenging, as seen Jeremy Keith’s quote, which can be enclosed in tags.

these elements, <cite> and <blockquote>, serve a semantic purpose and they provide a convenient way to apply custom styling.

The ‘’<q> ‘’ element in HTML stands for quote, and it automatically provides the appropriate quote marks for users. Inline elements like <strong>, <b>, <l>, and <em> wrap around inline phrase of text.

Block-level elements like block quotes, paragraphs, and unordered lists create separate blocks on the page. CSS can switch the layout behavior of elements from block to inline or vice versa. Black quotes and the ‘’<q>’’ element serve different purpose, with one used for inline phrases and the other for creating a block context. Understanding the elements is crucial before implementation layout or CSS.

The datetime attribute in HTML allows us to specify a date or time in a machine-readable format, such as May 8, 2025. The format must be specific, starting with the year, followed by the month and date.

Website often use phrases like ‘’five hours ago’’ or ‘’10 days ago’’ for time stamps, but they still use a machine-readable format. Machines prefer 24-hour clock format and can specify time zones like New York city’s time zone, which is five behind Greenwich Mean Time.

The datetime attribute allows for the combination of date and time, with options for separation them using a T or space. This format is acceptable in various programming languages and can be used in various combinations, ensuring time is easily understood by computers.

**HTML DATE AND TIME INPUTS**

HTML simplifies programming by using a single element called <time> to mark specific time ranges, such as day, date, or duration. This element consists of an opening and closing tag and can be used in any human-readable format. The main purpose of the <time> element is o convey the exact date or time to computers, using an HTML attribute called ‘’datetime.’’

The datetime attribute in HTML allows us to specify a date or time in a machine-readable format, such as May 8, 2025. The format must be specific, starting with the year, followed by the month and date. Examples include ‘’2025-05-08’’, ‘May 8, 2025’ or ’05,08’.

The datetime attribute allows for the combination of date and time, with options separating them using a T or space. This format is acceptable in various programming languages and can be used in HTML and other formats. It allows for machine-readable time communication.

**HTML CAPABILITIES**

A demo was demonstrated in various HTML elements and can be accessed using the “inspect element” or “tools” menu.

The left plane displays the HTML, including the document object model created by the browser. The inspector shows the DOM tree, a family of elements, Popular browser like Safari, chrome, and edge have similar developer tools with similar HTML panels, using indentation and triangles to understand the DOM tree. This allows users to inspect websites and learn from other developers’ use of HTML.

The code displays an unordered list with four items, but the count is incorrect. The browser is adding an extra set of tags, causing the DOM tree to believe there should be five items. To fix this, add a missing slash before the third item and use the developer tools in your browser.

**HTML ATTRIBUTE**

Attributes that enhance the capabilities of any element. These attributes, such as day, are specific to specific elements like the time element. They also work with multiple elements, such as images and videos.

Class and ID attributes are essential for naming HTML elements and referencing them in other parts of code stack. They can be used for CSS targeting but are more specific, causing issues. IDs are useful for addressing specific JavaScript elements or targeted links. In a demo, classes and IDs were added to CSS, and a table of contents with links jumped to ID-element.

HTML attributes enhance user interaction and provide browser powder. Content editable attributes allow interaction with screen, keyboard, and assistive devices. The “lang” attribute specifies language of content using a short code, while the “dir” attribute indicates text flow direction. These attributes are Global Attribute and can be used on any HTML element.

**ARIA ROLES**

ARIA roles are additional attributes added to HTML elements to enhance their meaning and help browsers understand their content. However, sometimes compromises must be made to ensure accessibility. The accessibility tree, a companion to the DOM tree, is crucial for assistive devices like screen readers. To improve accessibility, add an ARIA label to the HTML and hide individual letters within a div element. This removes the div and its contents from the accessibility tree while keeping them in the DOM tree. ARIA is a powerful tool for web accessibility, especially for teams working with semantic HTML or complex application interfaces.

**FORMATTING HTML**

HTML files have an opening and closing tag for most elements, with newer elements having both. Older elements, like the image element, have no closing tag. Back then people favored adding a slash at the of elements without a closing tag, but around twenty ten, people started reducing their importance. Browsers now accept various formatting options, but it’s up to the user to decide.

**UNSUAL CHARACTORS**

HTML symbols <, >, and are essential for content formatting. However, they disappear when written with spaces, causing the browser to assume it’s part of HTML code. To handle this, use a character entity in HTML, such as an ampersand, short code, or semicolon. These entities convert characters into desired ones and can be used to replace greater than or less than symbols. The W3C provides a reference chart for these entities. Non-breaking spaces in HTML prevent the browser from breaking the line between two words, ensuring to create multiple spaces between words, allowing for more precise formatting. Overall, HTML characters are displayed on webpages.

**HTML NAVIGATION AND LINKING**

Web links have become an integral part of our daily lives, with links appearing in various online forms such as navigation bars, menus, and article titles.

The concept of linking dates to the 1980s, when computer innovators focused on hypertext, hypermedia, and hyperlinks. The web’s birth is a testament to this obsession. To create a link, use the A element and an href attribute with a URL enclosed in quotes.

The A element can be inline or wrapped around text, images, or complex elements. Absolute URLs, which point to a precise location on the web, can be included without a trailing slash.

The HTTPS part, or Hypertext Transport Protocol, defines the rules for communication on the web and is crucial for linking. Nowadays, experts recommend using HTTPS enhanced security.

**HTML URL PATHWAYS**

Absolute URLs are used for linking to the same site and domain as the page contains the link, but relative URLs can be used when linking to something within the same site and domain. For example, if a website is launched at http://www.awesomedogs.com, all links should point to the local copy of the files, while links on a testing server should direct to files on that server.

Relative URLs are useful for referencing images files, video files, CSS, JavaScript files, or any other file in the images folder can be written as /images/logo.gif. This creates a URL relative to the location of the file, allowing the browser to locate the file. However, this trick only works for HTML files, not for images or CSS.

**SUMARRY FROM UNIT 5-9**

**HTML Working with Graphics and Images**

An image can be embedded into a web page using the HTML <img> tag. Technically, photos are linked to web pages rather than put into them. The relevant picture is held in place by the <img> tag.

The <img> tag lacks a closing tag, is empty, and just has attributes.

The <img> tag necessitates two attributes:

src: Indicates the image's path.

alt: Indicates a different text for the picture.

2 ways to insert images on the webpages

by giving the complete address (URL) or path to a file on the internet.

by giving the file path in relation to the web page file's current position.

In addition to understanding each of the strategies, we will first talk about embedding the image into the webpage.

Embedding images on a webpage: You can add or embed images on a webpage by using the tag. The "img" tag lacks a closing tag and is an empty tag, meaning it can only have a list of attributes. The inclusion of the photographs enhances the webpage's appearance and design structure while also improving quality. These days, pictures are linked to other pages on a website rather than being included directly on the page.

Adding an Image as a Link: If an image has a URL contained in it, it can function as a link. Using the "img" element inside of a "a" tag will accomplish this. To render the image on the webpage, we must provide the file location. When linking external resources like pictures, movies, style sheets, JavaScript, displaying other web pages, etc., file paths are utilized. A file's source must be known in order to insert it onto a web page.

There are two kinds of file paths**:**

Absolute file paths: They always include the whole directory list needed to find the file in addition to the root element.

Relative File Paths: On a file system, the file or folder is located using the hierarchical path representation.

**Working With Media**

 The audio element is a modern and flexible element with an opening and closing tag, providing more power and flexibility. It uses a source attribute to provide the URL of the audio file and can be used to specify multiple audio files, like the picture element.

The audio element can be used to link to an MP3 file, which allows the browser to provide controls like a play button, timeline, and volume control. Using JavaScript and the HTML media element API, the "controls" attribute can be added to the audio element, allowing users to play, pause, adjust volume, see the time, and navigate through the timeline.

Other attributes can be used with the audio element, such as "loop" or "autoplay". The source element can be removed from the audio element and placed on a separate source element, allowing for the addition of other source elements with alternative audio file formats. MP3s are widely supported in modern browsers, while OGG has some advantages but has not gained much popularity.

There is no second audio format recommended now, but understanding the syntax for supporting multiple formats is crucial in the future.

The web has revolutionized the way we connect and share content, including movies, TV shows, and teaching. The HTML video element allows users to easily put videos on web pages using an opening and closing tag. To display a video, use the source attribute to specify the video file, and if the controls attribute is added, the browser will automatically create a video player. However, there are two issues that need to be addressed.  
  
The first problem is with video encoding. H.264 is the dominant codec used for internet videos, but it is not open source and is patented by a consortium. This raises the question of why video codecs should be treated differently from other web technologies like HTML and CSS.  
  
The WebMD versus AV1 debate remains uncertain, but recent developments suggest that AV1 may be the superior video codec compared to H.264 and is also believed to be free of royalty charges. The video element in HTML allows for the simultaneous use of different codecs like H.264, WebMD, and eventually AV1. This flexibility extends to audio and picture elements as well, where multiple file formats can be specified using the source element with the appropriate attributes.

Captions and subtitles are essential for making content accessible to all users, regardless of their hearing or ability to understand it. The web provides multiple ways to provide content, such as using the track element and linking it to a text file to add captions to a video. This allows viewers to toggle captions on and off or switch between different subtitle options.  
  
On the web, a file format called WVTT (web video text tracks) is used, which follows a specific convention for providing information. Each line of text is accompanied by a time code, indicating when it should be displayed in the video. To display these captions on the video, insert a track element within the video element, specifying the file, kind attribute, label attribute, source lang attribute, and default attribute.  
  
A captioning icon appears, allowing users to turn captions off, enable automatic captioning, or select English captions. If a Spanish translation is required, another file for Spanish can be created and another track element to the same video element can be added.

**HTML Content Identification**

HTML language support is crucial for websites to be understood by search engines and for browsers to pronounce words correctly. The lang attribute is used to specify the language of a webpage, which can be set on the main element that wraps everything else, usually the HTML element. It has many options to indicate language or regional versions of a language, as well as other qualities like the writing system used. If a webpage has multiple languages, specify the language for each part of the content using the lang attribute on any element.  
  
The “dir” attribute is also important to specify the content's direction, as most languages flow from left to right horizontally, but some flow from right to left. When there is a mix of content, ensure to indicate the change in direction for each phrase.  
  
The meta-attribute is another important aspect of HTML language support. It is important to set the charset for your project, which is the set of characters or alphabets used by each language. Unicode, particularly UTF-8, is widely used and encodes content to support a vast range of characters, scripts, and emojis. It is essential to inform the browser about the character set being used to avoid confusion if the computer expects one thing and receives another.

HTML elements like div and span are useful when an element is needed that does not exist. Divs and spans were heavily used before HTML5 introduced semantic elements, but they are not good practice and can negatively affect users. Instead, developers should opt for the appropriate HTML element that serves the purpose.  
  
Divs are block-level elements, while spans are inline elements that do nothing until CSS or JavaScript is applied. For example, a simple article with a headline and four paragraphs can be grouped together to add a background colour to the paragraphs using a div with a class called "boxes". The changes taking effect can be seen with CSS.  
  
For a specific phrase in the text, the inline element span can be used to mark the desired phrase. Both div and span can use global attributes like class, id, lang, and aria roles. Div and span are useful when there is no other suitable element to use, acting as a last resort option. However, they should be used wisely.

**HTML INTEGRATION**

An html is fundamental building block of the world wide web.

The HTML page consists of several key components.

Document type declaration (<! DOCTYPE HTML>): specifies that the document adheres to the html standard.

<head> element contains meter information about the page such as the title.

<html> element the root element of an html page

<title> element specifies the title of the page displayed in the browsers title bar or tab.

<body> element defines the visible content of the page, including heading, paragraphs, images, links, and more.

Document head inside the head of a webpage, you put important information that the browser needs to know about the website.

<head> element is a container for metadata (data about data) and is placed between the <html> tag and the <body> tag.

Metadata typically define the document title, character set, style, scripts, and other meta information. And other meta information/

The <style> element is used to define style information for a single HTML page.

The rel attribute specifies the relationship between the current document and the linked resource.

Href specifies the location of the link document.

Hreflang indicates the language of the text in the linked document.

Media defines on which device the linked document will be displaced.

Sizes specify the size of the linked resource.

Title describes a preferred or alternate stylesheet.

Type specifies the media type of the linked document.

Main element is used once per webpage and tells the browser where and tells the browser where the main content is located.

Header the header and footer elements mark the header and footer cross on the page.

The Footer signifies that there are an extra thing to convey, regardless of its possible on the page.

Section the section element is used to mark section content. E.g. in a long essay with subheading each segment can be wrapped in a section element.

**WORKING WITH FORMS AND INTERATIVE ELEMENTS**

An html form is used to collect user input. The user input is most often sent to a server for processing.

<h2> h html form <h2>

The <form> element is a container for different types of input elements, such as text fields, checkbox, radio buttons, submit buttons.

The html <input> element is the most used form element.

It can be displayed in many ways, depending on the type of attribute.

Here are some examples.

<input type=”text”>displays a single-line text input field.

<input type=” checkbox”> displays a checkbox for selection zero or more of many choices.

<input type=” submit”> display a submit button

<input type>” button” displays a clickable button.

The HTML br element produces a line break in text (carriage-return)

It is useful for writing a poem or an address, where the division of lines is significant.

The <label> element defines a label for many forms elements.

It is useful foe screen-readers.