**Introducing HTML**

0.1

* HTML (HyperText Markup Language) provides essential structure and semantics of the web pages for accessibility and seo(search engine optimization)- if we take away HTML nothing is left, no webpage, no content nothing. for proper website we need HTML
* html is not only for readable to the user but also conveyed the structure of the document, the relationship of the content with other and allow the user to link other pages and sites. It does everything bz it is a markup language.
* HTML and css work closely together. Xhtml, html and html 5 all are different versions of html. Html 5 is the latest one.
* CSS-Cascading Style Sheets. Controls the visual presentation of page elements, it is responsible for page layout, color, typography and element dimensions, how page looks- page designs-if we take away css, the page is simply uninstalled.
* JS- page behaviour- if we take away JS, no function perform

0.2

* <p lang="en"></p>
* <p> is opening tag </p> closing tag, lang - attribute, "en"- value which is always in inverted commas.
* Child tag always close before parent tag. for ex. \*<p>Sandhya is a good girl<em>She does her work<strong>carefully</strong></em></p>

0.3 HTML 5

0.4

* <https://webplatform.github.io/docs/html/>
* https://developer.mozilla.org/en-US/docs/Web/HTML/Reference

0.5 Code Editor- brackets.io

**Basic Page structure**

0.1

< ! doctype html > it shows the browser and user which version of html is used...

< html >

< meta charset="utf-8">Every tag does not required a closing tag same as with meta tag. It needs only opening tag. "utf-8" is a standard charset.>

< meta name="describtion" content="A page for exploring basic HTML documents">

< title >we have two version of region</title>

< head > this makes the document better but not define. Head represent functioning content</head>

< body > visual elements are in body tag</body>

< /html >

0.3

<html lang="en"> by putting lang attribute to html tag, it show that this HTML document is going to be use English as a first language. It is nice to write this but it is not mandatory or can say it is not required.

0.4

0.5 **Content Models**

**There are seven main models**

1. **Metadata content** - metadata is a content that sets up the presentation or behavior of the rest of the content, or that sets up the relationship of the document with other documents or that conveys other "out of band" information. ..base, link, meta, noscript, script, style, title.

2. **Embedded content**- is any content that import other resources into the document...audio, convas, iframe, embed, img, math, svg, video, object

3. **interactive content**- is any content that specifically intented for some type of user interaction....a, audio\*,button, embed, iframe, image\*, input\*, keygen, label, object\*, select, textarea, video\*....here \*under certain circumstances

4. **Heading content** - defines the heading of the section...h1, h2, h3, h4, h5, h6

5. **Phrasing content**- this is the text of the document...a\*, abbr, area\*, audio, b, bdi, bdo, br, button, convas, cite, code, data, date, datalist, del\*, dfn, em, embed, iframe, img, input, ins\*, kbd, keygen, label, map\*, mark, math, meter, noscript, object, output, progress, q, ruby, s, samp, script, select, small, span, strong, sub, sup, svg, textarea, time, u, var, video, text\*, wbr...here \*under certain circumstances

6. **Flow content**-this contains the maturity of elements in html 5. An element that would be included in the normal flow of the document... a, abbr, address, area\*, article, aside, audio, b, bdi, bdo, blockquote, br, button, convas, cite, code, data, date, datalist, del, dfn, div, dl, em, embed, fieldset, figure, footer, form, h1, h2, h3,h4,h5,h6, header, hr, i, iframe, img, input, ins, kbd, keygen, label, main ,map , mark, math, meter, noscript, object, ol, output, p, pre, progress, q, ruby, s, samp, script, section, select, small, span, strong, style\*, sub, sup, svg, textarea, time, u, ul, var, video, text\*, wbr...here \*under certain circumstances

7. **Sectioning content**- it defines the scope of header and footer….article, aside, nav, section

**Formatting Page Content**

**0.1**

HTML is a markup language that means, it use tags to identify content on the page…if there is no element is used, the browser shows the test by default setting.

* <pre> this element is used for giving space or can say for writing poems</p>
* <h1>it is the biggest one</h1> there are 6 types of heading h1 to h6 <h6> this is the smallest one</p>
* There should be one h1 on single page . There should be strategy behind using heading.
* <i>italic</i>
* <b>bold</b>
* <em>emphasize</em>
* <strong>strongly emphasize on text</strong>
* <br> break, it don’t need closing tag

**0.2**

Heading help define the structure of the page and control the hierarchy of the content. You can use heading values ranging from a top-level heading of h1 all the way down to an h6. Use heading in an intelligent manner, according to the importance of the content.

**0.3 Formatting paragraph**

**0.4 Controlling the breaks <br> line break**

**0.5 Formatting page content by bold, italic, strong and emphasize**

**0.6 Displaying special characters**

&lang; for left angle bracket < , &rang; for right angle bracket >, &amp; for &, &copy; for copyright, &trade; for trademark,

**0.7 Controlling white space**

&nbsp; for non-breaking space. For ex Formula&nbsp;one

* It is not possible to memorize all character entity so use Wikipedia

List of XML and HTML character entity….

**0.8 Inserting** **image**-

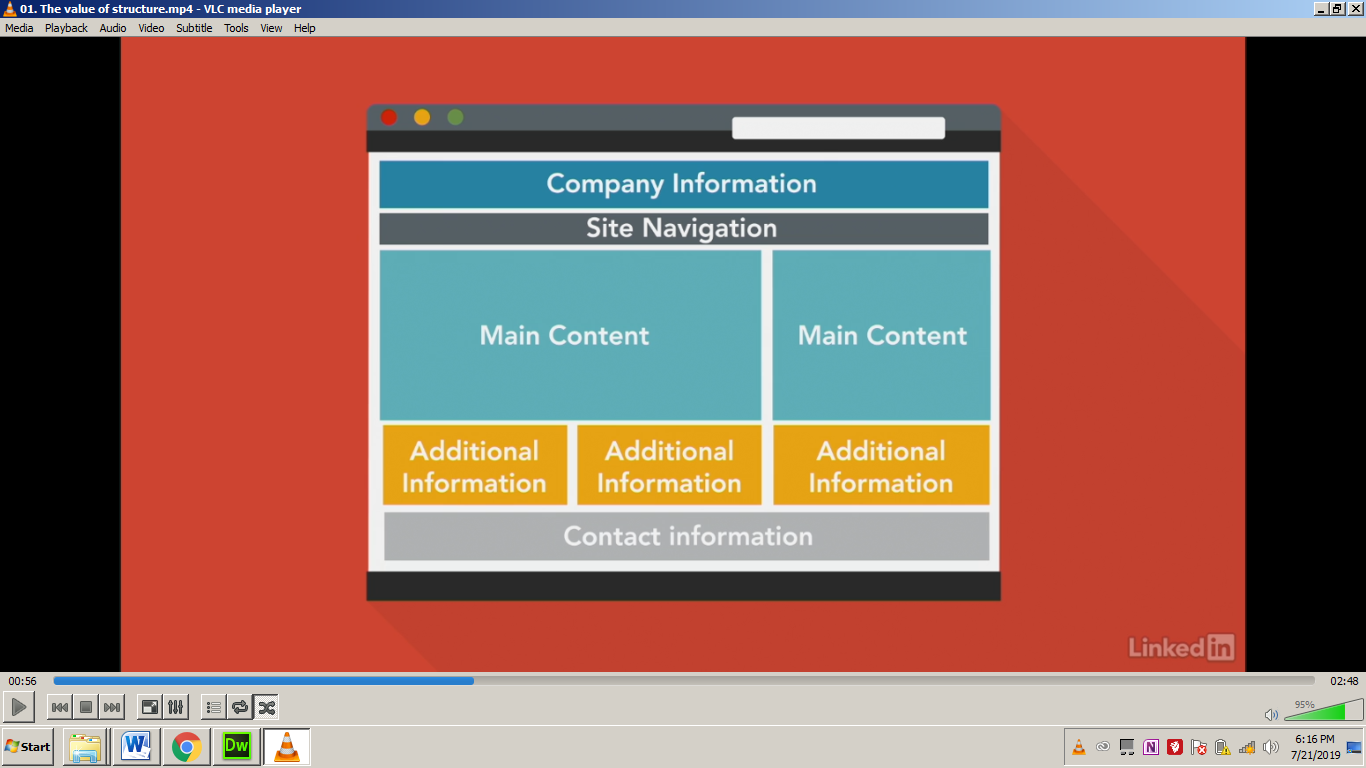
**Src**-it is attribute used to tell the browser where to find the image.

**Alt**-it allows you to pass along descriptive text representing the image.

<img=src”warrior-pose.jpeg” width=”200” height=”200” alt=”warrior-pose”>

**Structuring Content**

**0.1 standard layout for website**





**0.2** **Controlling document heading**

In this, headings are used systematically and how the content is organized has been taught in this video.

**0.3** **The nav element**

The nav element represent the section of a page that links to the other pages or to parts with in the page.

**0.4** **The article element**

Article element is used to give structure to the content.

**0.5 The section element**

The section element represents a genetic section of a document or application. A section , in this context, is a thematic(having or relating to subjects or a particular subject) grouping of content. The theme of each section should be identified, typically by including a heading (h1-h6 element) as a child of the section element.

**0.6** **The aside element**

The aside element is designed for side panel.

**0.7 The div element**

* The div element has no special meaning at all. It represents its children. It can be used with **class**, **lang**, and **title** attributes to markup semantics common to a group of consecutive elements.
* It is a enormous tag. It is used for grouping and structing content.
* <div class=”book”>
* **We can use 1 ID in one style sheet but class can be used as many as we required.**

**0.8 The header element**

The header element represents the introductory content for its nearest ancestor sectioning content or sectioning root element. A header typically contains a group of introductory or navigational aids.

**The footer element**

The footer element represents a footer for its nearest ancestor sectioning content or sectioning root element. A footer typically contains information about its section such as who wrote it, links to related documents , copyright data, and the like.

**The main element**

The main element represents the main contents of the body of a document or application. The main content area consists of content that is directly related to or expands upon the central topic of the document or central functionality of an application.

The main content area of a document includes content that is unique to that document and excludes content that is repeated across a set of documents such as site navigation, links, copyright information, site logos and banners and search forms(unless the document or applications main function is that of a search form).

* It helps in semantic information…seo n all

**0.9 WAI-Aria in HTML**

<header role=”banner”>

<nav role=”navigation”>

<main role=”main”>

<article role=”article”>

<aside role=”complementary”>

<footer role=”contentinfo”>

* With this the page become more accessible…

1. Meaningful structure
2. Semantic(meaningful valid structure) meanings
3. consistency

**Creating Links**

* 1. **Exploring the anchor element**

For creating links we use **<a>** anchor element **</a>**

**href**= hypertext reference

**target**= target is used to open a page in a new window or tab or control where the page opens with in frameset. POINTS to remember- same site pe \_blank use nahi karna, one site se another site k link dene par \_blank use karna hai taki actual site apni identity na khoye.

Target=\_blank

\_parent framename

\_self

\_top

**rel** attribute describes the relationship of a target object to the link object.

rel=alternate next

author nofollow

bookmark noreferrer

download\* prefetch

help prev

license search

tag

<a href= “seed.html” target= “\_blank” rel= “next” title= “planting a seed”> Planting a seed</a>

Title is descriptive text.

**Linking to pages within your site**

**<a href=”about.html” target=”\_blank” title=”branches of yoga”>Hatha Yoga</a>**

**<a href=”../about.html” target=”\_blank” title=”up one directory”>Gyan Yoga</a>**

**<a href=”../../about.html” target=”\_blank” title=”up two directory”>Bhakti Yoga</a>**

**Linking to external pages**

**<a href= "https://8limbsyoga.com/about-yoga/" target="\_blank" title="Yoga"> <b>Know about Yoga</b> </a>**

**Linking to downloadable resources**

**<a href=”\_assets/syntax.zip” title=”HTML syntax reference”> HTML Syntax references</a>**

**For reading**

**<a href=”\_assets/syntax.pdf” title=”HTML syntax reference”>HTML Syntax references</a>**

**For download(It is used for downloading)**

**<a href=”\_assets/syntax.pdf” title=”HTML syntax reference” download=”HTML\_syntax”> HTML Syntax reference! </a>**

**0.5 Linking to page regions**

**<h1 id=”#top”>Yoga</h1>**

**<h2 id=”#meaning”>Meaning of Yoga</h2>**

**<h2 id=”#limbs”>What are the limbs of Yoga</h2>**

**<h2 id=”#benefits”>Benefits of Yoga</h2>**

**<h2 id=”#branches”>Branches of Yoga</h2>**

**How to give link? For ex**

**<p><a href=”#top” title=”back to the top”>Back to top</a></p>**

**Creating Lists**

* 1. **Unordered lists- Grouping of list items in no specific order.**

**<ul>**

**<li>Item one</li>**

**<li>Item two</li>**

**<li>Item three**

**<ul>**

**sublist<li>item 3.1</li>**

**<li>item 3.2</li>**

**<li>item 3.3</li></ul></li>**

**<li>Item four</li>**

**</ul>**

**OUTPUT**

* **Item one**
* **Item two**
* **Item three**
* **Item 3.1**
* **Item 3.2**
* **Item 3.3**
* **Item four**

**Ordered List- Grouping of list items in a specific order**

**<ol start=”1” reversed>**

**<li>Item one</li>**

**<li>Item two</li>**

**<li>Item three**

**<ol type=”i”>**

**sublist<li>item 3.1</li>**

**<li>item 3.2</li>**

**<li>item 3.3</li></ol></li>**

**<li>Item four</li>**

**</ol>**

**OUTPUT**

1. **Item one**
2. **Item two**
3. **Item three**
   * 1. **Item 3.1**
     2. **Item 3.2**
     3. **Item 3.3**
4. **Item four**
   1. **Definition list- Grouping of terms and description**

**<dl>description list</dl>**

**<dt>term</dt>**

**<dd>description</dd>**

**For ex**

**<dl>**

**<dt>Html</dt>**

**<dd>Hypertext markup language</dd>**

**</dl>**

**HTML and CSS**

**CSS- Cascading style sheet**

**<style> </style>**

**<!\_ \_ <style> this way we can off the style sheet and ON the another one.**

**</style>\_ \_>**

**0.2 Inline Style**

**<body>**

**<h2 style=”color:red;”**

**font-weight:”normal;”> Style me using the style element </h2>**

**</body>**

* 1. **Embedded styles**

**<html>**

**<head>**

**<meta charset=”utf-8”>**

**<title>embedded styles</title>**

**<style>**

**h2.alert {**

**color:red;**

**font-weight:normal;**

**}**

**</style>**

**</head>**

**<body>**

**<h1> sandhya </h1>**

**<h2 class=”alert”> she is beautiful </h2>**

**</body>**

**</html>**

* 1. **Controlling typography**
  2. **Adding color 0-255 Red, Green, Blue**

**body{**

**width:80%;**

**margin:0 auto;**

**font:100% Arial, sans-serif;**

**}**

**h1 {color: red;**

**font-family: Cambria, "Hoefler Text", "Liberation Serif", Times, "Times New Roman", "serif";**

**font-size: 2em;**

**font-weight: normal;**

**font-style: italic;**

**}**

**.paragraph{ color: #880808;**

**line-height: 1.6;**

**text-align: justify;**

**width: 70%;**

**margin: 0;**

**margin-bottom: 1em;**

* 1. **External style sheet**

**Basic Scripting**

**Javascript**

* **Validating forms**
* **Remote Scripting**
* **Special Effects**
* **Improving navigation**
  1. **The script element**

**In html document, if we want to add styling to the document we use style element likewise if we want to add scripting to the document we use applinamed scripting element**

**<script type=”text/javascript”>**

**//tabbled panels**

**</script>**

**Points to remember**

       Margin

Padding

Content

Content

**Cascading- It means that styles can fall from one style sheet to another, enabling multiplestyle sheets to be used on one HTML document.**

**Void/Self closing document- Void elements don’t have closing tags and don’t wrap any content because they are the content.**

**\* Previous xhtml required self close <hr/> but html 5 does not.**

**<area>, <base>, <br>, < col>, <embed>, <hr>, <img>, <input>, <keygen>, <link>, <menuitem>, <meta>, <source>, <track>, <wbr>**

**DOM (Document Object Model)**

**Body**

**h1----------------------P-----------------------UL**

**siblings siblings**

**li li li (Children of UL)**

**Text Editor**

**HTML, Notepad, Visual Studio, Sublime Text, Brackets, Atom**

**HTML Comment**

**<!\_\_// Intro\_\_>**

**External CSS**

* **A separate css file with a.css file extension.**
* **Always reference within the <head>**
* **Uses the <link> and two attributes, rel and href**

**<link href="styles.css" type="text/css" media="all" rel="stylesheet" />**

* **Use lower case – index.html**
* **Don’t use spaces or symbol = @,&**
* **Use dashes=\_,-**

**If style is saved in a folder then**

**<link href=”css/styles.css” rel=”styles.css>**

**For ex:- .P {**

**Color:blue**

**}**

**Here, P is selector**

**{ - declaration blocks**

**}**

**Color is property**

**Blue is value**

* **The same class can be used multiple times per page.**
* **Use whitespace to make the css easier to read.**

**ID Selector**

* **Ids can only be used once per page. IDs should have unique value.**
* **The value is the selector started with a # symbol.**
* **Multiples Ids can’t be used in the same Html element.**

**Default font size is 16 px.**

**CSS**

* **Cascading style sheet-Different language from HTML.**
* **Stylesheet language with its own syntax rules.**
* **Type selectors match the html by using the element name**
* **This way we can organize project**

**Project-name**

**!\_ \_css/**

**!\_ \_styles.css**

**!\_ \_images/**

**!\_ \_profile.jpg**

**!\_ \_index.html**

**Multiple classes**

* **Separate multiple classes with a space.**
* **Apply different styles to each class separately.**
* **Combine classes, with no space to select both**

**.style {**

**Color:blue;**

**}**

**.name {**

**Font-size:16px;**

**}**

**.style.name {**

**Background:green;**

**}**

**Pseudo-class Selectors**

**Descendent selectors**

* **Use descendent selectors, separated by a space, to match the descendent elements**

**<header> main h2{**

**<h1>header1</h1> color: blue;**

**<h2>header2</h2> }**

**<main>**

**<section> Output**

**<h2>section2</header> header1**

**</section> header2**

**<article> section2**

**<h2>article2</header> article2**

**</article>**

**</main>**

**The more selector we use the more specific the parent become.**

**For ex:- section p a {**

**Color: red;**

**}**

**Grouping multiple selectors.**

**\*we can also target multiple elements by grouping them in one declaration block. This will apply the same style to every selector. Each selector is separated with a comma. It helps to create more sufficient CSS bz it reduces the number of declarations, it also easier to update one declaration block.**

**<main> h1,h2,h3 {**

**<section> color:green;**

**</section> }**

**<article>**

**</article>**

**</main>**

**Each selector in the group is independent of each other.**

**.class-name h1,h2 { .class-name h1{**

**Font-size: 12px; font-size: 12px;**

**} }**

**h2{**

**Font-size: 12px;**

**}**

**Pseudo-class selectors**

**a{**

**background: purple;**

**padding:16px;**

**}**

**a:link {**

**color:white;**

**}**

**a:visited {**

**background: light blue;**

**}**

* **When you visit any link then its color is changed with above class.**

**Hover is not only used for a tag it’s used in other also like P tag.**

**a:active {**

**border: 1px solid black;**

**}**

**a:hover {**

**background:none;**

**}**

**P:hover {**

**Background: white;**

**}**

* **Add comment in pc**

**Ctrl + / (forward slash)**

**Video 16**

**ID values can only be used once per page; use them for unique or global styles that are not repeated.**

**#global-footer {**

**background:white;**

**}**

**\*use IDs for page linking and class for styling.**

**17. CSS Comments**

**/\* \*/**

**Usage**

* **Leave notes for yourself and others**
* **Organize code blocks**
* **Comment out code to hide it temporarily**
* **Css comments are used to keep the things organized.**

**18. CSS Color value**

**color: black;**

**Hex code: is always start from**

**# followed by six characters**

**Color: #000000;**

**Color :rgb(0,0,0) RGB stands for red, green and blue values on scale between 0 and 255**

**Sources- colours.neilorangepeel.com , randoma11y.com**

**And coolers.co**

**20. CSS Inheritence**

**Styles can be inherited from ancestor (parent) to descendant (child)**

**Elements.**

**But few elements won’t be inherited from parent like link (<a> tag</a>) bz it has its own specific default style.**

**CSS Specificity**

**The selector with the highest specificity will be applied.**

* **Type selectors (h1) – lowest**
* **Class selectors (.example)**
* **ID selectors (# example) – highest**

**For ex- <h1 class=.example id=#example> Selector specificity </h1>**

**Here, Id selector will be applied. It will overwrite both type and class selectors.**

**21. font and font-family property**

block level element -

inline level element-

\*width and height: sets specific size of the content box

\*padding: space inside of the element

\*margin: space outside of the element

\*float, display and position can't be used all together on the same element. If using float, then display is iqnored and if using display, then position is iqnored.

Question