HTML

What is HTML?

* A markup language the tells web brosers how to structure the web pages you visit.
* Consists of a series of elements, which you user to enclose, wrap, or makrup different parts of content to make it appear or act in a certain way.
* The enclosing tags can make contant into a hyperlink to connect to another page, italicize words, and so on

Anatomy of an HTML Element

* <Opening tag>Content<Closing tag> = 1 element

1. The opening tag : Consists of the name of the element wrapped in opening and closing angle brackets. This opening tag marks where the element begins or starts to take effect.
2. The content : The content of the element
3. Closing tag : Same as the opening tag, except that it includes a forware slash before the element name.

Nesting elements

* Elements can be placed within other elements called nested elements.

ex) <p>My cat is <strong>very</strong> grumpy.</p>

* The tags have to open and close in a way that they are inside or outside on another

Block vs. Inline elements

* Block-level elements: Appears on a new line following the content that precedes it. Any content that follows a block-level element also appears on a new line. ex) heading, paragraphs, lists, navigation menus..etc.
* A block-level element wouldn’t be nested inside an inline element, but it might be nested inside another block-level element
* Inline elements : Contained within block-level elements, and surround only small parts of the document’s content.
* Will not cause a new line to appear in the document. ex) <a>, <em>, <strong>..etc

Empty Elements

* Some elements consist of a single tag, which is typically used to insert/embed something in the document. ex) <img>

Attributes

* <p class=”editor-note”>some content</p>
* class=”editor-note” is called an attribute. It can contain extra information about the element that won’t appear in the content.

Boolean Attributes

* Can only have one value, which is generally the same as the attribute name. ex) “disabled”
* <input type=”text” disabled = “disabled”>

Anatomy of an HTML document

1. <!DOCTYPE html>: The doctype. A historical artifact that needs to be included for everything else to work right.
2. <html></html> : Wraps all the content on the page. (Root element)
3. <head></head> : Acts as a container for everything you want to include on the HTML page, that isn’t the content the page will show to viewers such as keywords and a page description that would appear in search results, CSS to style content, character set declarations, and more.
4. <meta charset=”utf-8”> : Specifies the character set for your document to UTF-8, which includes most characters from the vast majority of human written languages. With this setting, the page can now handle any textual content it might contain.
5. <title></title> : Sets the title of the page, which is the title that appears in the browser tab the page is loaded in.
6. <body>/body> : Contains all the content that displays on the page, including text, images, videos, games, playable audio tracks, and whatever else.

<!DOCTYPE html>

<html>

<head>

<meta charset=”utf-8”>

<title>My title</title>

</head>

<body>

<p>some content</p>

</body>

</html>

Entity references

* The character <,>,”,’ and & are special characters. You need to use other character to represent these characters.

|  |  |
| --- | --- |
| Literal character | Character reference equivalent |
| < | &It; |
| > | &gt; |
| “ | &quot; |
| ‘ | &apos; |
| & | &amp; |

Comments

* use <!- - and - - > .

Metadata

* Metadata is data that describes data by using <meta>

Adding custom icons to your site

* <link rel=”icon” href=”some\_pictre.png” type=”image/x-icon>

Applying CSS and JavaScript to HTML

* <link rel=”stylesheet” href=”my-css-file.css”>
* <script src=”my-js-file.js” defer></sciprt> : defer instructs the browser to load the JavaScript after the page has finished parsing the HTML.

HTML layout elements

1. <main> : For content unique to this page. Use <main> once per page inside body
2. <article> : Encloses a block of related content that makes sense on its own without the rest of the page
3. <section> : Similar to <article> but is more for grouping together a single part of the page that constitue one single piece of functionality or a theme
4. <aside> : Contains content that is not directly related to the main content but can provide additional information indirectly related to it
5. <header> : Represents a group of introductory content.
6. <nav> : contains the main navigation functionality for the page.
7. <footer> : represents a group of end content for a page

Span, div and br

1. <span> : An inline non-semantic element, which you should only use if you can’t think of a better semantic text element to wrap your content.
2. <div> : A block level non-semantic element, which you should only use if you can’t think of a better semantic block element to use.
3. <br> : Creates a line break in a paragraph.

Images in HTML

* Use <img src=”some\_image.jpg”>

Alternative text

* Value supposed to be a textual description of the image, for use in situations where the image cannot be seen/displayed or takes a long time to render because of a slow internet connection.

Width and height

* Use the width and height attributes to specify the width and height of your image.

Image title

* Use title attributes to images to provide further supporting information if needed

Video and audio on the web

* Use <video src=”” controls> : controls allows to user to be able to control video and audio playback.
* Or you can use <source> elements. This allows the browser to go through the <source> elements and play the first one that it has the codec to support.

Video Features

1. width and height: Control the video size either with these attributes or with CSS
2. autoplay: Will play the video right away, while the rest of the page is loading
3. loop : Makes the video start playing again whever it finishes
4. muted: Causes the media to play with the sound turned off by default
5. poster: The URL of an image which will be displayed before the video is played.
6. preload: Used for buffering large files; it can take one of three values:
   1. “non” does not buffer the file
   2. “auto” bufferes the mdedia file
   3. “metadata” buffers only the medtadata for the file

Audio

* <audio> element does not support visual component, width/height attributes.
* Does not support poster attribute
* You can also use <track> and “subtitles” or “captions” or “timed descriptions” to cues.

SVG

* An XML-based language for describing vector images. SVG is a markup except that you’ve got many different elements for defining the shapes you want to appear in your image, and the effects you want to apply to those shapes.
* SVG is for marking up graphics, not content.

ex) <svg version=”1.1” baseProfile=”full” xmlns=<http://www.w3.org/2000/svg>> …..</svg>

* SVG uses for text in vector images remains accessible, lend themselves well to styling/scripting, because each component of the image is an element that can be styled via CSS or scripted via JavaScript.

Tables

* A table is a structured set of data made up of rows and columns.
* The content of every table is enclosed by <table></table>
* The smallest container inside a table is a table cell, which is created by <td></td>
* Use <tr> for each row.
* Use <colgroup> and <col> for styling all of the columns of rows in pattern.
* <caption> : Give your table a caption by putting <caption> just below the opening <table> tag.
* You can add <thead>, <tfoot> and <tbody> to make your table more structural.
* <thaed> usually wrap the first row of the table
* <tfoot> usually wrapt the foot the of the table
* <tbody> usually wrapt the table content
* These are usefuly when styling with css.

HTML Tags Note

https://avalangche.tistory.com/64