Crash course:

<https://www.youtube.com/watch?v=qz0aGYrrlhU>

Book code:

<https://www.htmlandcssbook.com/code-samples/chapter-01/>

Validator:

<https://validator.w3.org/>

Nomenclature

**Front end** – what you see in the browser. Use HTML, CSS, JavaScript

* HTML defines building blocks of our webpages
* CSS used for styling webpages to make them pretty
* JS used for adding functionality to webpages

**Back end** – what powers the front end but storing and accessing data.

**Reac**t – JS library for building UI

**Version Control System** – like a git system

**Web servers –** the client (website) and the server (computer). Client requests a service, and the serve accommodates. The message is formatted in http – this is the language that client and server would communicate in.

**http request –** sent by the client to the server

**http response -**  sent in response by the server to the client

**DOM –** document object model. Sent by the serve to the client to render the page.

**Tags –** also referred to as elements, have closing, i.e., <p>, and ending, </p>, syntax to specify their start and end and are used to specify what a particular part of a text is for.

**Attributes** – is inserted inside the opening tag to enhance tags functionality by telling us more about the elements. They are comprised of name = “value”.

**HTML**

**Tags**

*Linking to stuff*

<a></a> is used to define a link

* href = <http://www.abcd.com> is an attribute inside the tag that will link the website
* href = <mailto:roman@tanvas.co> will specify the email address
* target = “\_blank” will open the link in a new window
* The text which will hold the hyperlink is defined between the tags

<img /> used to define an image

* src=”dir” specifies the image source
* alr=”description” is used to describe what is being shown
* width = “600”
* height = “500”
* align = “left” will place the image to the left of the paragraph. Can also be “right”, “top”,”bottom”,”middle”

<figure></figure> is used to define a figure in which the caption can be mentioned

* <figcaption> is used to specify the caption

*Structuring the page*

<h1></h1> is a header tag

* Style = “color:blue;text-align:center;”

<p> is a paragraph tag

* Lang = “en-us” is an attribute to specify language

<em> is an emphasis tag and is by default set to italic

<strong> is a strong importance tag and is by default set to bold

<b> is a bold tag to make a specific word or a line of words bold.

<i> is italic tag.

<sub> and <sup> are for subscript and superscript tags respectively.

<br/> is a line break tag

<hr/> will add a horizontal rule line to break up the text

<blockquote> will identify a piece of text as a quotation.

<cite> will indicate where a citation is from

<dfn> indicates defining instance of a new term

<ol> creates an ordered list

<ul> creates an unordered list (bullet points)

<li> is used to add a new element to the list

<dl> creates a definition list

* <dt> specifies term being defined
* <dd> contains the definition

<li> each element in the list needs to be tagged by this tag.

<div></div>will group elements together in a single block element to which certain formatting can be applied

<span></span> is similar to <div> but for inline characters

<iframe></iframe> is used to embed an entity into a page

* Attributes are : width, height, scrolling (“yes”/”no”), frameborder (“0”/”1”)

<meta> is placed inside the <head> element and contains information about the page

* Name is an attribute that can contain a number of descriptive values, i.e. author of the page, keywords, description

*tables*

<table></table> is used to start a table

<tr></tr> is used inside a table tab to start a table row

<td></td> is used to start a new cell in the table, to be used within <tr>

* Colspan=”2” specifies that a particular row should run across two columns (any number is permitted)
* Rowspan=”2” same as above but for rows

<th scope=”col”></th> is used like td but names the columns or rows given the name.

<thead></thead> is a header tab

<tfoot></tfoot> is the footer tab

*forms*

Types of forms: test, password, text area, radio buttons, checkboxes, dropdown boxes, submit buttons, image buttons, file upload.

<form></form> contain form controls

* Action attributes contain the url of the page on the server that will receive the information
* Method attributes – can either be “get” or “post”. Get is used for short forms and are only getting data from a server not sending it back. Post is used for users to upload files, contains sensitive data, exceptionally large information.
* Id attribute is used to identify a form from others

<input/> is used to create controls about the form which will be sent along with the entered data to distinguish each piece.

* Type = “text” will create a single line of text input. = “password” will block out characters entered. “radio” will create round multiple-choice option. “checkbox” will give a checkbox option with multiple possible choices that could be made at once. “file” allows you to upload a file as a user. “submit” Is used to send a form to the server. “image” will specifically require an image. “date”. “email”. “url”. “search” used for single line of text that will be searched in a query.
* Checked attribute will select the box which is selected when the page loads
* Maxlength sets the maxlength of entered data
* Name is the identifier of the data being entered, i.e., “username”
* Value attribute specifies the value that Is sent to the server when a particular option is selected.
* Placeholder is the text that resides in the textbox before user inputs their text.

<datalist></datalist> can be paired with <input list = “name”> where id if datalist is set to name

<select></select> is a dropdown box.

* Name is the attribute sent to the server

<option></option> gives options for the select tag

* Value is the attribute that is sent to the server

<textarea></textarea> creates a multiline text input

* Cols and rows are attributes specifying the size of the textbox

<button></button> will create a button with set text and image

<fieldset></fieldset> places related forms together into a single field

<legend></lengend> is contained in the fieldset and contains the caption which helps identify the purpose of the group

<label> specifies the text in front of the control

* For is used as an identifier attribute

*Commenting*

<!-- Insert the comment here -->

*VIdeos*

<script></script> specifies which script you would like to embed. i.e. swfobject.embedSWF(“’). These settings go into the <head> section of the file. This is one of the methods used in embedding the Flash videos.

* src specifies where the script can find the element in question
* type specifies what language the script is written in

<video></video> adds HTML5 video to the webpage

* src specifies the path of the video relative or absolute
* poster specifies the path to the image which will serve as the video placeholder before video plays
* preload can be “none”,”auto”,”metadata” specifying whether the browser should or should automatically load the video, or if it should partially load it, respectively.
* Autoplay whether the video should start automatically
* Loop whether the playback should loop

<source></source> replaces the src attribute in video and should be used inside the video block. Is used to give a certain video in different formats. i.e. providing the video In both flash and hmtl5

*Audio*

Skipping Flash version

<audio></audio>

* src specifies the path of the audio relative or absolute
* preload can be “none”,”auto”,”metadata” specifying whether the browser should or should automatically load the audio file, or if it should partially load it, respectively.
* Autoplay whether the audio should start automatically when the page is visited
* Controls specifies whether audio file should display any of the controls

**CSS**

*Commenting*

/\* Insert the comment here \*/

*Basic syntax*

selector {declaration}

declaration consists of property : value

* Selector indicates which element the role is applied to. It is literally the name of the tag which would be affected. Several selectors can be grouped by separating them with a comma.
* Declaration indicates how the elements being referred to should be styled. Several can be separated by a semicolon.

Inside the <head> of html file we can specify the .css file location using <link/> tag

* Href is the link to css file
* Type is the description of the file. For css should be set to “text/css”
* Rel specifies the relationship between html and file its linked to. Set to “stylesheet” for css files.

We can also embed the css code in html using <style></style> tag.

* type attribute should be set to “text/css” for actual css files

We can also include the css code inside the html file that will be affected inside the tag we want to be targeted.

Including # in the selector will create the change around entities with a given id, i.e. p#hello will select entities <p> with id hello

Including . in the selector is the same as # for id’s but for classes

**CSS selectors**

\*{} targets all elements on the page

h1, h2, h3 {} matches element names

#id{} will target all tags with a given id

.class{} will target tags with a given class

p.note{} targets only <p> elements whose class attribute has a value of note

li>a{} targets any <a> elements that are children of <li> element (but not other <a> elements in the page)

p a {} targets any <a> elements that sit inside a <p> elements even if there are other elements nested between them

hl+p {} targets the first <p> element after any <h1> element (but not other <p> elements)

hl-p {} if you had two <p> elements that are siblings of an <h1> element, the rule would apply to both

tag[class = “dog”] matches specific tag with dog class

**Properties**

*Color*

color – there are four ways to specify a color. 1) with a name, 2) with hex codes, 3) with rgb, i.e. rgb(10,100,255) and 4) hsl which sets hue (0 – 360), saturation (0%-100%), lightness(0%-100%), and alpha (0-1).

background-color – can take on values that color can. This property can be applied to text, body, blocks

opacity – ranges between 0 and 1. This value can also be set by the 4th entry in the rgb and hsl lists

*Text*

font-family – specifies the type fonts to be used in each text block. You can list several so that a user that doesn’t have one installed on their computer can see the text in other options.

Font-size – can be listed in 12px (pixels), percentages (scaled to 16px), or 1.2em (which is equivalent to width of letter m. lol)

Font-weight – normal or bold

Font-style – normal, italic, oblique

Text-transform – uppercase, lowercase, capitalize

Text-decoration – underline, overline, line-through, blink

Line-height – sets the height of the textbox in which the text resides. Unit in em.

Letter-spacing / word-spacing - units in em.

Text-align – left, right, center, justify

Vertical-align – baseline, sub, super, top, text-top, middle, bottom, text-bottom

Text-indent – indents first line of text.

Text-shadow – creates a shadow behind the word for an effect.

Position: relative (relative to the box in which the block resides), fixed (positions a text element anywhere relative to the page), absolute (positions an element and take it out of being considered for positioning by other elements)

z-index sets the priority of positioning being applied

*Table*

Border sets the border width, color, and shape

Border-collapse: collapse will remove all the padding in the cells and just leave behind the border lines

*Boxes*

Min-width / max-width set the smallest and largest sized bounding box around a text when the window is minimized and maximized. Can also use min-height max-height

Overflow – tells the browser what to do when the text does not fix bounding window. Options are overflow and scroll.

Padding / border / margin all effect properties of the box surrounding a text

Blockquote – will generate a separate block for a quote within the text.

* Float property will identify where to position it
* Width should be used with float to style the quote well

Clear – can left, right, both, none. Specifies which side of a box cannot be touched by a box identified earlier in the code.

Float – tells the box where it should be positioned within the page

Margin – creates a gap between neighboring boxes or images or text

Width/height – makes a box to the specified dimension

Display – block/inline specifies if the box/image/text shall be rendered in a block format or in line with other text.

Background-image: url(“”) – will set the image in the background of a given box element.

Background-repeat: repeat-x, repeat-y, no repeat - will specify how the image will be repeated to fill the box

Background-attachment: fixed, scroll – will specify if the image scrolls as the user scrolls on the page or not.

Background-position: left/center/right top/center/bottom place a given image in the site

*Cursors*

Cursor – crosshair, default, pointer, move, text, wait, help, utl(“link”)

*Wrapping text and content*

Display: flex will use flex wrapping style

Flex-wrap: wrap will wrap the content such that you don’t have to scroll on limited size windows

*Pseudo-elements*

Used as **selector:pseudo-element {}**

:first-letter, :first-line – lets you specify the unique styling information of the first letter or line in a text.

:link , :visited – lets you set styles for links

:hover – sets the type of a text when you hover over them

:active – sets the type of the text when the user interacts with it

:focus – when links on a page are tabbed through, they have focus

@import – can be used on separate CSS style sheets to tag multiple of them. e.g. @import url(“main.css”)

We can also just include multiple of the <link> tags with separate pages

Entities

https://dev.w3.org/html5/html-author/charref

Shortcuts

**!+enter** : will create a basic boiler plate template from which the html file can be written

**Specificity (the order in which attributes will be applied)**

1. Inline – referencing an attributed in the <> brackets of the tag
2. Id
3. Class
4. Type