**HTML CSS by freeCodeCamp**

**HTML 🡪** It is a markup language, not programming language. It is responsible for webpage structure:

Syntax: **<element>** content **</element>**

* Install Chrome, Install VS Code, some extensions too
* Create a folder, open it in VS Code and add a HTML file
* Una vez en VS Code asociamos el Repository Git: https://github.com/PadillaTom/FrontEndJourney.git

(Log in user.name // user.email // git init // git remote origin LINK //git pull origin master // git push –u origin master)

* Una vez Asociado Git con Remote y Folder comenzamos los tutoriales

Structure and Elements:

<!DOCTYPE html> 🡪 HTML version.

<html> 🡪Root Element.

<head> 🡪 Information about the page (Meta, Links, Title, etc) .

Content won’t be visible.

<title> xxx 🡪 Page title (shows in tab).

<body> 🡪 What will be displayed on the page.

* There are Parent Tags and Children Tags:  
  <head>

<title> **Titulo 1**

* If something is too long we can use **ALT + Z** (word wrap)
* What is Emmet? Something that speeds up my work. In VS Code it’s already built in. EJ: Every time I want to create a HTML element I don’t need to type the < >, Emmet will fill with the suggested abbreviation. <html></html>
* Vemos la estructura general y una vez armada podemos comenzar: Se recomienda utilizar Live Server (right click on VS Code + open with Live Server)

**Headings <h>:**

They are a kind of title. There are six types, being h1 the bigger size:

<h1> <h2> <h3> <h4> <h5> <h6>

**Paragraph <p>:**

Where we are going to put some kind of Text component. HTML it is “white space collapsing” (HTML is going to ignore extra spacing) : <p> **Hello**

**this is a paragraph**</p> 🡪 HTML will ignore all the spaces and print it all together.

* Si queremos crear SPACES tenemos que usar un tag especial
* Por momentos sabemos que tenemos content, pero no lo tenemos, podemos usar **LOREM X** (x = number of words):

<p> **Lorem 20** </p>

**Images <img src = “ “ alt=” “ >:**

We can upload images for every kind of content. They must be in the same folder as the website or can be uploaded by URL().

SRC = ” “ 🡪 Path (Folder **./** or URL() )

ALT = “ “ 🡪 What we can see when the image is broken, like a short description

./ 🡪 “in the same directory”

../ 🡪 “outside the directory”

URL() 🡪 Get to the image file (Jpg, Png, etc) and copy link.

Debemos usar website de busqueda para copyright free images (Pixabay, Gratisography, etc).

* Render size 🡪 via HTML (we will use CSS or even cropping

them before)

<img src=”” width=”260 alt=”””> 🡪 Width =” “ // Heigth will be

automatically adjusted.

* Ahora … si tenemos muchas imagines el usuario va a experimentar un website muy lento…. Podríamos ajustar las imágenes previamente Photoshop, Mac, Etc. o **utilizar CSS.**
* Creamos una IMG folder y dentro sub folders según utilidad.

**Comments <!—xxx -->:**

We should comment at the Start or End of every section.

**CTRL + }** 🡪 Shortcut para añadir comentario in windows. It also

converts a line into a comment In and Out.

**Line Breaks <br>:**

This will break the white space collapsing. Just like print(“ “) in python. It adds lines.

**BR\*6** 🡪 Shortcut that will add 6 times the <br>

**Links <a href=” “>:**

It is called an Anchor Tag:

<a href= ” # ” > **Some clickable Text** </a> 🡪 **#** it is a placeholder that will be replaced later in time for a real link

* Target =” **\_blank** “ 🡪 It will open link on a new Tab
* What about internal surfing? Creamos un **About.html** y agregamos un **<a>** que nos lleve a **Index.html**.
* “Back to the top” 🡪 Usamos **src =” # “. #** Como en CSS # representa un ID, podríamos poner cualquier **#ID** y nos llevaría hasta él.
* Podemos usar distintos elementos para clickear un Link:

<a>

<img>

**<Sup> and <Sub> Elements:**

These elements will be places in top or bottom of the RENGLON.

<h1> **Hello I am** <sup>**1st**</sup> **John to arrive in** <sub>**town**</sub></h1>

* 1st and Town will be shown NOT IN LINE.

**<Strong> and <Em>phasis Elements:**

Within some text we can have STRONG and EM.

<p>**lorem20**<strong>**Bold Text**</strong><em>**Italic Text**</em></p>

Strong 🡪 Bold text

Em 🡪 Italic Text

* The ideal is to have a CSS for style and HTML for structure. Not to mix them both.

**Special Characters” &char; “ :**

Por ejemplo el logo de Copyright:

<h1> Copyright **&copy;** </h1> 🡪 Replace copy for another character, Emmet will suggest the names.

**List Structures <ul> <li>:**

1.39.20