Formspree.io

MongoTufi\_23

Hack academy

<https://plataforma.ha.dev/enrollments>

Antes de los editores de texto se usaba el bloc de notas.

***Visual studio code.***

***Live Server.*** Permite visualizar los html en el navegador.

***Prettier***. Da ayuda para que el código se vea bien. Ayuda a corregir errores.

***Material Icons.*** Para visualizar iconos a la izquierda.

HTML está presente en todos los sitios web. Está compuesto de etiquetas y atributos. No es un lenguaje de programación, porque carece de lógica y funciones, entre otras cosas.

Estructura de un sitio web: Frontend y Backend. Los navegadores solo interpretan 3 lenguajes: HTML, CSS y JAVASCRIPT. Los lenguajes de programación no. El backend necesita ser compilado e interpretado antes de llegar al sitio web.

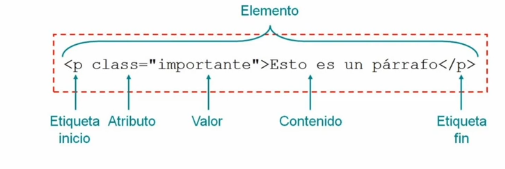
Frontend:

HTML= Estructura + semántica

CSS= Presentación + diseño

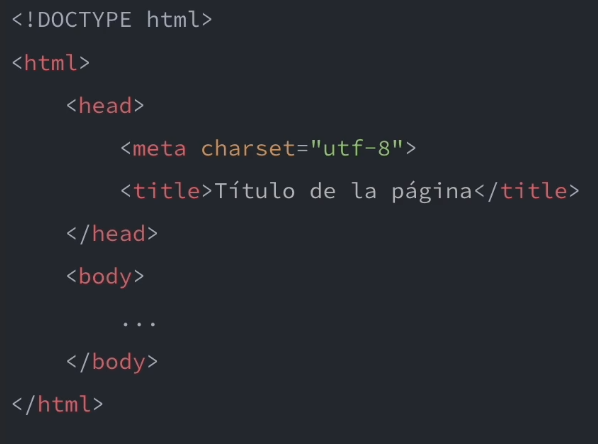
JavaScript= Interacción

Etiquetas:



**Etiquetas estructurales**

Son las que deben ESTAR SIEMPRE. Podría faltar alguna y no haber problemas, pero no debe ser así.



DOCTYPE html Declara la versión del lenguaje que estamos utilizando.

HTML. Marca que todo lo que ponemos dentro es html.

Head. Aporta valores informativos del documento. Se anidan ahí otras muchas etiquetas. Meta (idioma, entre otros atributos). Title (implementa un título en la pestaña del navegador).

Body. Aquí va todo el contenido.

En VS Code genera el code estructural de modo automático tocando ! (Emmet abreviation).

Antes debí guardar el archivo con la extensión .html Dice que también podría ponerse html:5 y lograr el miso resultado.

Con click derecho. Open with live server.

HTML lang=”es” ahí queda en español.

**Etiquetas de texto**

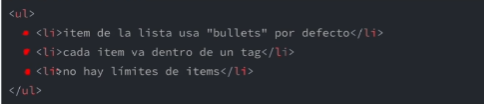
6 etiquetas h (heading). Solo debería haber un h1.

P (paragraph). Pueden haber tantos como uno quiera.

Las etiquetas van en el orden que uno quiere.

Listas sin enumeración. Se genera con 2 etiquetas Ul y Li. Ul es unordered list (Lista sin ordenar).

Anidadas van las Li (list item).



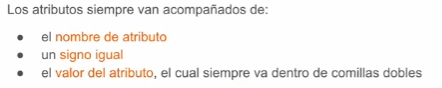
Etiqueta OL. Ordered list. La viñeta es un número.

Cursivas <em> y negritas <strong>

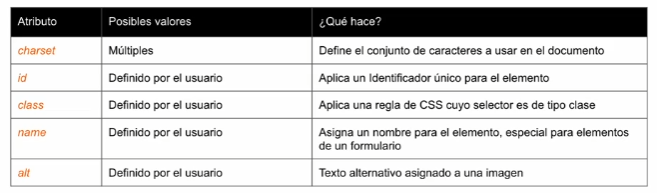
<em> Lanús es el <strong> mejor </strong> </em> == Lanús es el ***mejor***

***Atributos***

**Proporcionan info adicional sobre los elementos. Modifican la funcionalidad o agregan.**

****

A todas se les puede agregar atributos. Algunos no surten efectos en algunas etiquetas.



Hay atributos para dar forma. Todo eso está obsoleto, antes se usaba, ahora se hace con CSS.

**Rutas**

La imagen (sonido o video) no se inserta en el HTML, es referenciada, la ruta es la ubicación de un archivo. Puede ser interna, externa o relativa.

Rutas absolutas. Se indica, o bien dentro del disco duro, o bien desde un dominio https.

Rutas relativas. Indica la ubicación con respecto al archivo HTML con el que se trabaja. Si comparten carpeta, solo se pone el nombre del archivo a añadir.

**Imágenes**

Etíqueta **img** y atributo **src** (source u origen). Luego se pone la ruta, absoluta o relativa.

Agregar el atributo **alt** es de buen uso. Por si la imagen no se carga se puede leer lo que ahí se pone. También vale para personas que no pueden ver, eso podría leerlo un software.

Rutas absolutas para imágenes, no pueden hacer referencia al disco duro, solo a internet.

Recomienda guardar las imágenes en una carpeta, dentro de la carpeta en la que se trabaja el HTML.

Enlaces

Etiqueta **a** (ancor o ancla). Relativos o absolutos.

El atributo es **href**. Ahí se indica a dónde se quiere ir. 3 tipos: externo, interno y local.

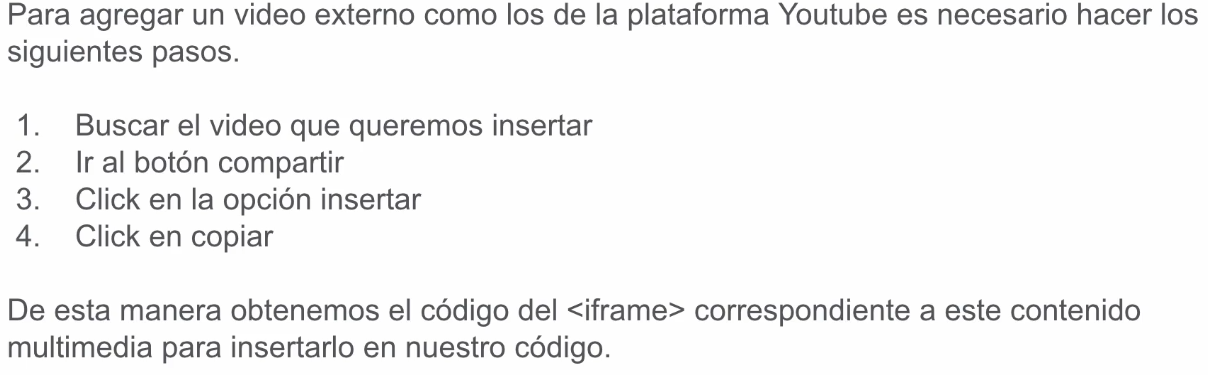
Externos\_ Se va a algún lugar fuera del sitio web. Https. Se abre en la misma pestaña, por lo que se abandona tu página. Para evitar esto se agrega el atributo Target=\_“blank” inmediatamente después de poner la dirección.

Locales. Conecta con otro documento del mismo sitio web (pero otro HTML). Solo cambia que el href es distinto. Se usa una ruta relativa.

Internos. Se navega internamente dentro de un HTML. El href se inicia con un numeral **#**, el cual hace referencia a lo nombrado en un “id”, el texto tiene que ser el mismo.

**Insercion de contenido externo**

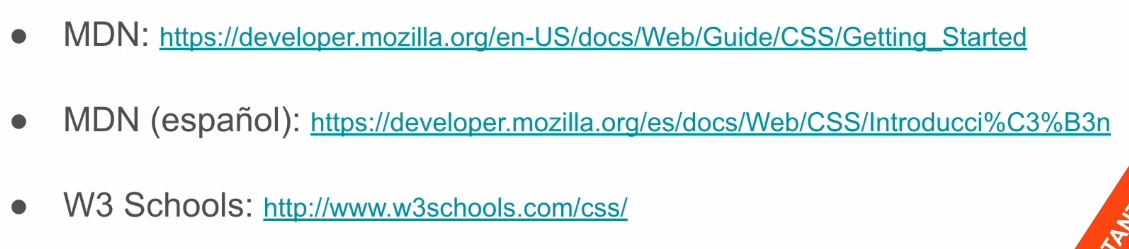
Etíqueta iframe. Permite incrustar un HTML dentro de un HTML principal. Usa el atributo src, entre otros.



No lo hice exactamente así, aunque si es copiar y pegar. Con Google maps es casi igual.

CSS

Cascading style sheets. Hojas de estilo en cascada.



Ahí hay cosas adicionales. Como por ejemplo, para meter alguna animación.

Reglas de estilo. Cada una posee un **selector**, que tienen una **declaración**, o sea, un par **propiedad** y **valor**. Por ejemplo:



Vincular CSS con HTML:



Etiqueta link debe ir dentro de head con 2 atributos, rel y href.

**Selectores**

Es la parte de la regla que permite seleccionar a qué elemento se le va a aplicar el estilo. 4 tipos: element, class, id, pseudoclass.

-Elemento. El nombre es igual a la etiqueta, como p o h1. Se aplica a todos esos.

P{ Color:red;}

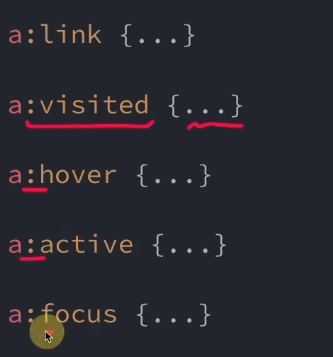
-Clase e Id. Primero se le inventa un nombre al selector. Id se puede usar una sola vez en un HTML, en tanto que el selector de clase se puede aplicar a varios elementos. Recomienda por esto no usar id.



Hay que retocar el HTML. En CSS se le antepone el punto (.). Usa 2 selectores. Al primer párrafo le adosa los 2 selectores. Con el id se antepone no un punto sino el numeral (#).

Pseudoclase. Selectores que se aplican sobre otro selector. Se le antepone al selector a cambiar los 2 puntos (:).

Por ejemplo, se escribe a:visited{blabla} Significa que cualquier etiqueta a que ya fue visitado, va a tomar la regla de estilo blablá.



Así se escribe y las describe algo. Hover es la única que se puede aplicar a elementos que no son enlaces, sino a cualquier elemento.

Jerarquía de selectores. El id es mas especifico que la clase. Esta es mas especifica que element. Si son igual de específicas, se impone la que está escrita mas abajo. Si no, gana la mas especifica. Id le gana a clase y así.

**Propiedades tipográficas**

Familia tipográfica.

Font-family: Courier, “Lucida Console”, monospace;

Pone 3, ya que si el navegador no encuentra una, pasa a la siguiente.

Font-size. Se puede usar pixeles (px), el cual no recomiendan. (em) Es relativa y puede cambiar y los (%). Recomiendan usar ***rem***, que permite establecer el tamaño de forma relativa al tamaño del texto establecido en el root de HTML(O sea, se pone al ppio del CSS), en el Font size que se ponga allí.

Interlineado. Line-height. Espacios que queda entre línea y línea. Se puede expresar en px o em

Trazado. Font-weight (normal o bolt). Refiere a cómo se ve la tipografía. Si es negrita o no.

Estilo Font-style (cursiva o normal). Italic.

Text-align (center, left, right, justify). Una sola.

Decoración. Text-decoration (underline, line-through, none). Lo que está subrayado parece un enlace; usarlo solo para eso.

Fondos

De color o de imagen.

Formas de indicar un color:



Palabra clave. Background-color: blue; y listo. Es sencillo, pero hay pocos colores (140 aprox).

Funciòn RGB. Aplica color a partir de 3 valores, R G y B. Red, green y blue. Tambièn está RGBA (Alpha u opacidad). Valores desde el 0 al 255 o del 0% al 100%. Se pone rgb (120,80,10);

Codigo RGB hexadecimal. Es el mas usado por desarrolladores web. Hexa viene de 6 valores. Se antepone a los 6 dígitos, el numeral #. En visual studio code habría una paleta de color.

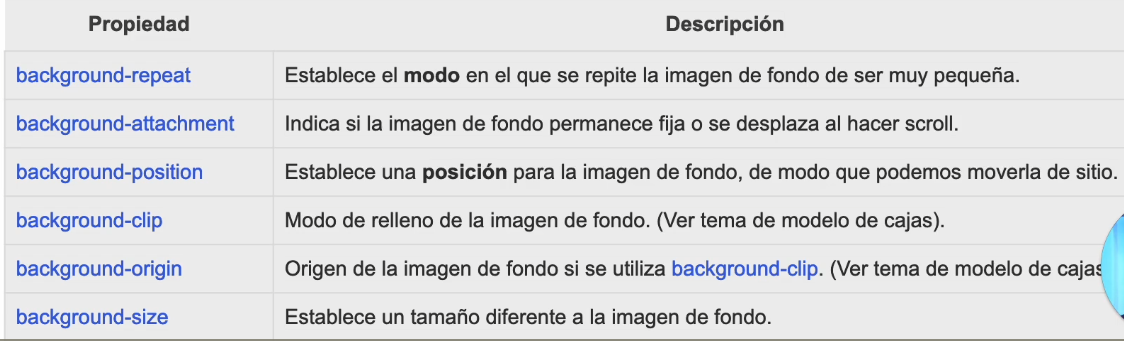
Función HSL. Hay 3 valores, que hacen referencia a saturación y luminosidad. La h es hue (matiz de color), s es saturación y l luminosidad o brillo. H va de 0deg a 360deg. S y L van del 0% al 100%.

El Alpha aquí sirve para fusionar colores que uno vaya sumando. Va del 0 al 1.

Fondo de imagen

Va al CSS no al HTML.

Background-image se usa. Reciba la dirección URL. Absoluta o relativa. Se puede modificar la imagen.



Attachment se mueve cuando se scrollea la pantalla.

Background-size. Si se pone un solo valor es el ancho, el alto sería proporcional. Si no, se pasan 2 valores, ancho y alto. En ese orden. Con valor contain, crece y crece, hasta que choca con un límite, en el ancho o el alto. Con cover se expande, puede pixelarse la imagen.

Para imágenes de fondo recomiendo algo así como tapices, mosaicos.

Manejo de URL. Con / se mete en la carpeta. Para ir para atrás, salir de una carpeta se pone “../”

Background position. Pïde 2 valores. El primero del eje X, el segundo del Y.

Elementos block, inline e inline block

Elementos de bloque. Generan un salto de línea, empujando todos los elementos hacia abajo. Ocupan todo el ancho del que disponen. Elementos block mas usados h1, etc, div, p, ul, ol, form.

Inline, de línea. Ocupan solo el ancho que ocupan y no generan saltos de línea. Elementos inline: a, em, strong, button, span.

Cambiar la visualización de un elemento. Con la propiedad display: (block; inline, none (oculta el elemento), inline-block (Se le puede aplicar ancho y alto, no genera saltos de línea, es un mix de los otros 2)).

Hay unos márgenes del body que no se ocupan. Salvo que se los saque luego.

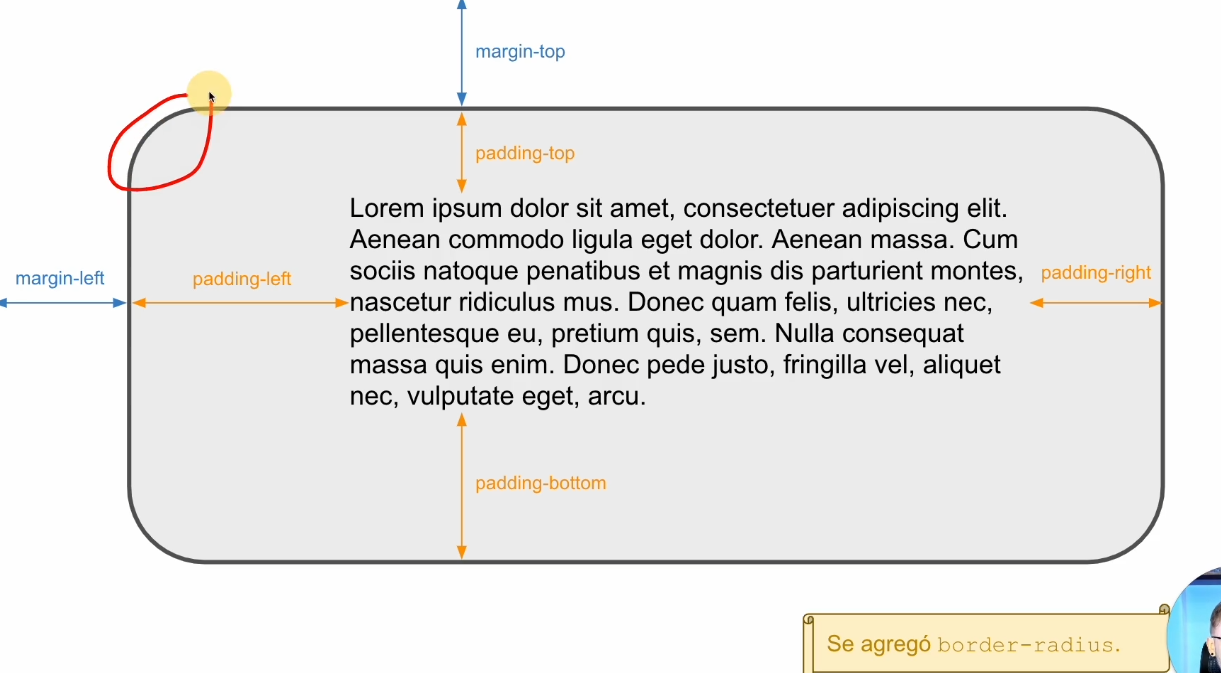
**Width y height**

Width Permite cambiar el ancho de un elemento. Px y %, rem y vw. Recomiendan usar porcentajes.

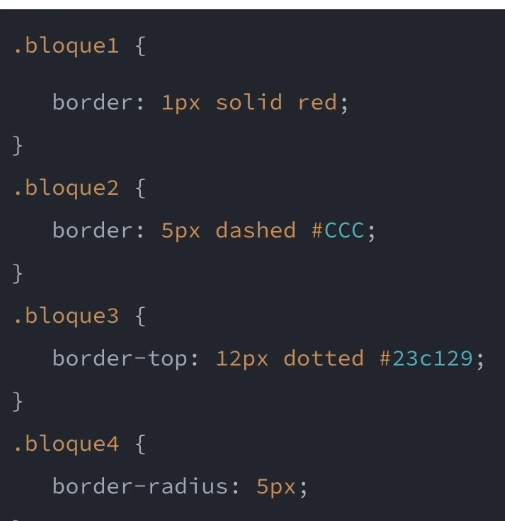
Height. Alto. Recomiendan no aplicarlo al elemento.

**Margin, padding y border**





Border siempre recibe 3 valores:



Si al margin se le pasa un solo valor, será igual para los 4 lados. Si no, se pasan 4 valores. Con Css se le puede poner margin:0; a body y así se pega a los bordes de la pantalla.

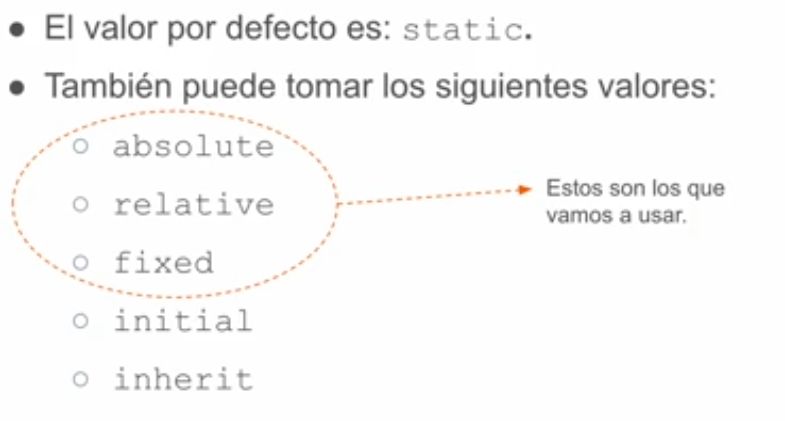
Padding=Relleno=espaciado interno. El margin es un espaciado externo.

Al border se le puede dar color. Si no se lo hace, por defecto, el color del borde es el mismo que el de la tipografía.

Chrome dev tools: Muestra la estructura HTML y qué pasa con CSS

Posicionamiento de los elementos

Propiedad position.



Fixed (FIJO). Nada lo va a poder desplazar, por mas que se scrolle.

Static. Por defecto.

Relative. Desplaza en función de su ubicación original, no en relación a la pantalla.

Absolute. Similar al fixed. Los demás elementos asumen que este no existe, por lo que pueden ocupar su espacio (al igual que con el fixed, no con los otros), solaparse. Pero al scrollear, cambia su ubicación en pantalla.

Coordenadas. Left, right, bottom y top.

Flexbox

Es un conjunto de propiedades.  
Flex ítems y flex containers. Dado el container (padre) se posiciona el item.

Flex-direction:row. Es para que los ítems se posicionen horizontalmente. Column los ubica uno debajo del otro.

Flex:1; es una forma de dar tamaño. 1 es que va a tomar un tercio del tamaño total. Podría ponerse width 50% o lo que sea, si te complican los flex.

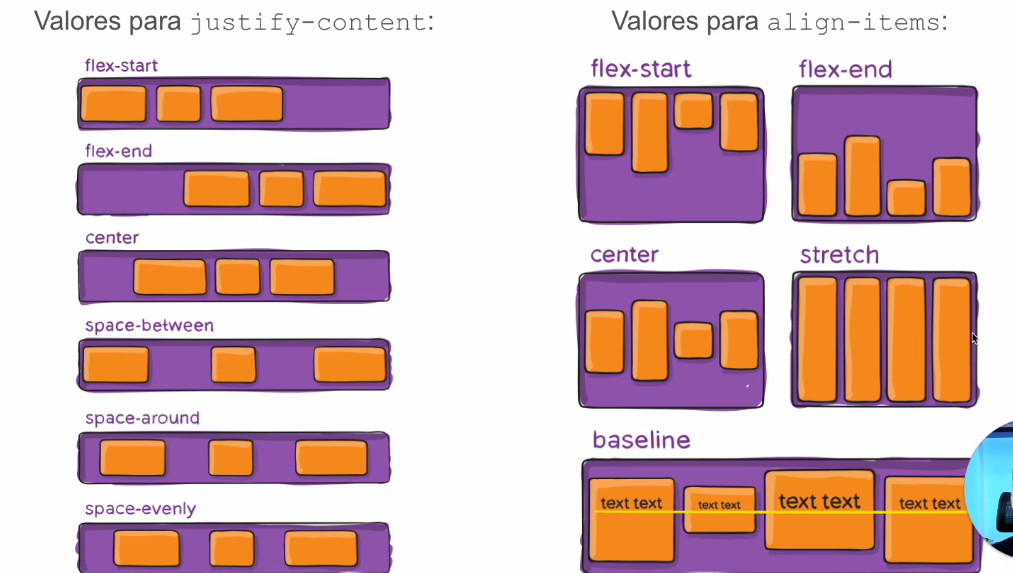
Align.items:center. Alinea, sobre el eje vertical, a la mitad. Align-items: por defecto viene con strech, significa que los ítems van a tomar todo el alto del container. Se le puede poner flex-start, flex-end, flex-center

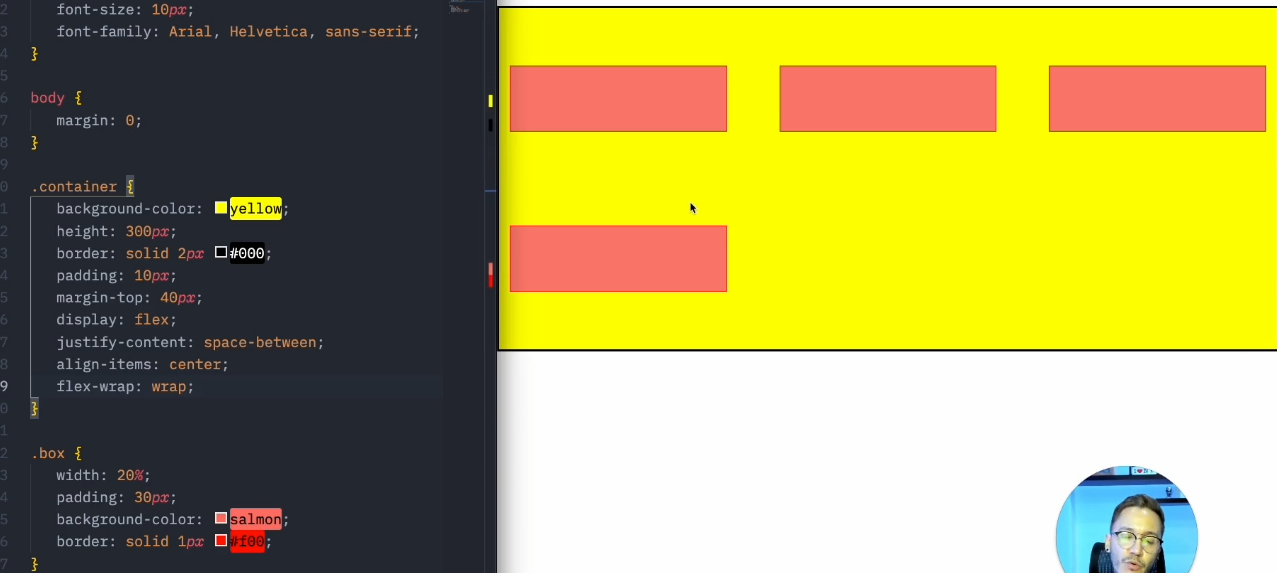
Justify-content:center. Alinea, sobre el eje horizontal. Space-between genera un espacio entre los ítems, el space-around es parecido, también separa de los costados del container.

Flex-wrap:wrap. Funciona cuando no hay espacio horizontalmente. Da un salto y va para abajo

Display flex. Se le pone al container, no al item.

Los porcentuales que se ponen en el ítem son en relación no a body, sino al container.





Flexboxfroggy.com hay un juego para practicar los flex.

Formularios

Etiqueta <form> en HTML. Necesita como mínimo 2 atributos: Action y Method

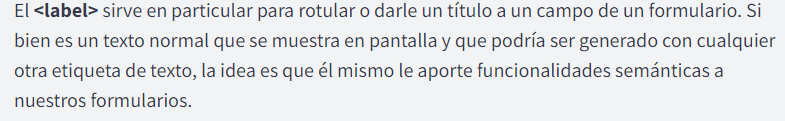
Action. Quién va a procesar la información. Formspree.io. Haciéndose una cuenta ahí se puede hacer, es una forma sencilla, sin saber de back-end. Te da una url que es la que se copia ahí.

Method. A través de qué protocolo vamos a enviar los datos. 2 protocolos: HTML get y post. Aquí siempre se escribe post.

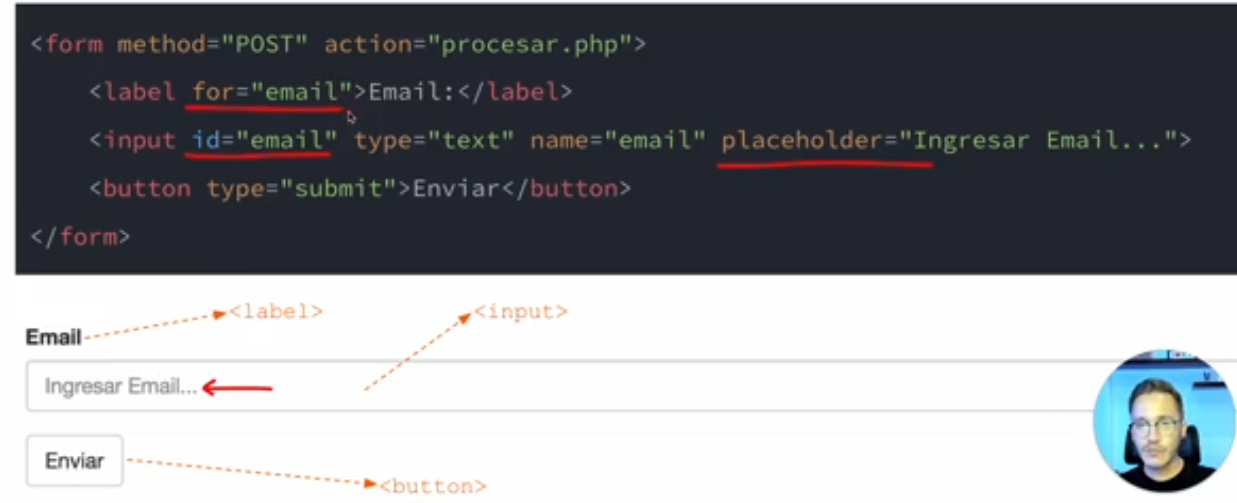
Etiquetas Label, input y Buttom

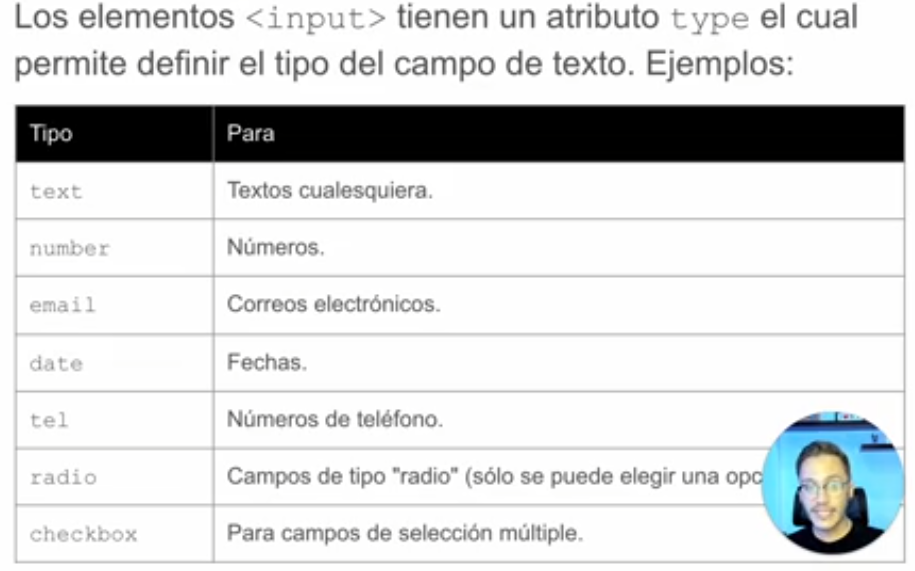
Input. Ingresar. Atributo type y name: rótulo, identificador.

Label: es el texto que aparece:



Buttom: Es el botón: submit (enviar formulario)





Placeholder es un texto de ayuda. Al escribir el usuario la primera letra, se borra.

El id está conectado con el for del label.

Etiquetas select y textarea

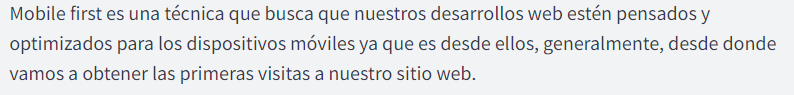
Select + option= menú desplegable.

Cada option es una de las opciones que se despliegan. También lleva atributo name. El id también, que se conecta con el for.

Textarea es área de texto. Name, id, y row, que hace al tamaño del área.

Type:radio son esas pelotitas que uno marca si la quiere elegir de entre varias opciones. Llevan el mismo atributo name. Se les define un value. El value es lo que uno recibe cuando se procesa.

Diseño responsive/ mobile first



Responsive. El sitio web se ve bien en todos los dispositivos. Se adapta a ellos.

Mobile first, lleva a que primero se desarrolle la resolución para celulares, luego tables y luego pcs. Así sería mas sencillo.

Si se hace primero la versión de pc hacen falta muchas adecuaciones para pasar a la versión de celular.

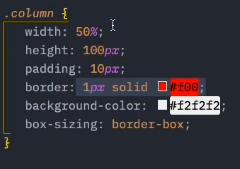
Etiqueta meta, con atributo name viewport. Viewport es la cajita donde se muestra el contenido (no es exactamente el ancho de la pantalla). También está el atributo content, width=device-width. El contenido aquí se adapta al ancho del dispositivo. También está initial scale=1; para que el zoom esté en un 100%.

Toca meta:vp y aparece toda la estructura:



Medidas relativas/Porcentajes

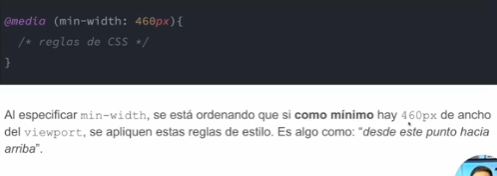
Porcentajes se suelen usar para establecer el ancho de los elementos. La referencia de dicho ancho es su elemento contenedor. Si no tiene contenedor, su referencia será el body. El alto manejarlo con pixeles.



Afirma que cuando uno pone width 50%, eso no se respeta posteriormente ya que después se le suman 20 pixeles por el padding, mas 2 pixeles del borde. Por eso al final pone lo del box sizing. De esa forma, esa caja si o si va a medir el 50% de su contenedor.

Media queries

Conjunto de reglas de Css que permiten reorganizar el contenido dependiendo de las condiciones de visualización del documento. Se suelen poner al final del Css.



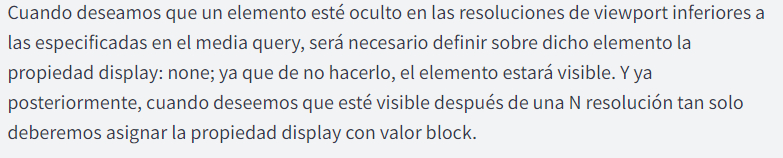
Si, la resolución como mínimo tiene X cantidad de Px (460) voy a aplicar la regla que tengo dentro de la llave.

Siempre se escribe igual @media (min-width:…)

Podría usarse max-width. Si empezás con la versión de escritorio y querés armar la versión de celular. Iría, por ejemplo, de 960 hacia debajo de Px.

Cuando se supera tal numero de pixeles se pueden cambiar muchas cosas con el media querie. Es una condición, un if. Si se cumple la condición, se aplica tales cambios.

Recomiendo escribir como máximo 4 media queries.

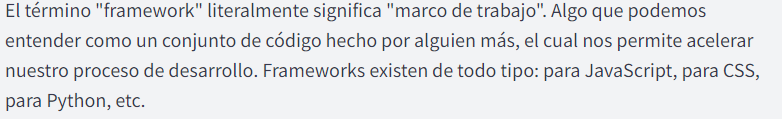


Bootstrap

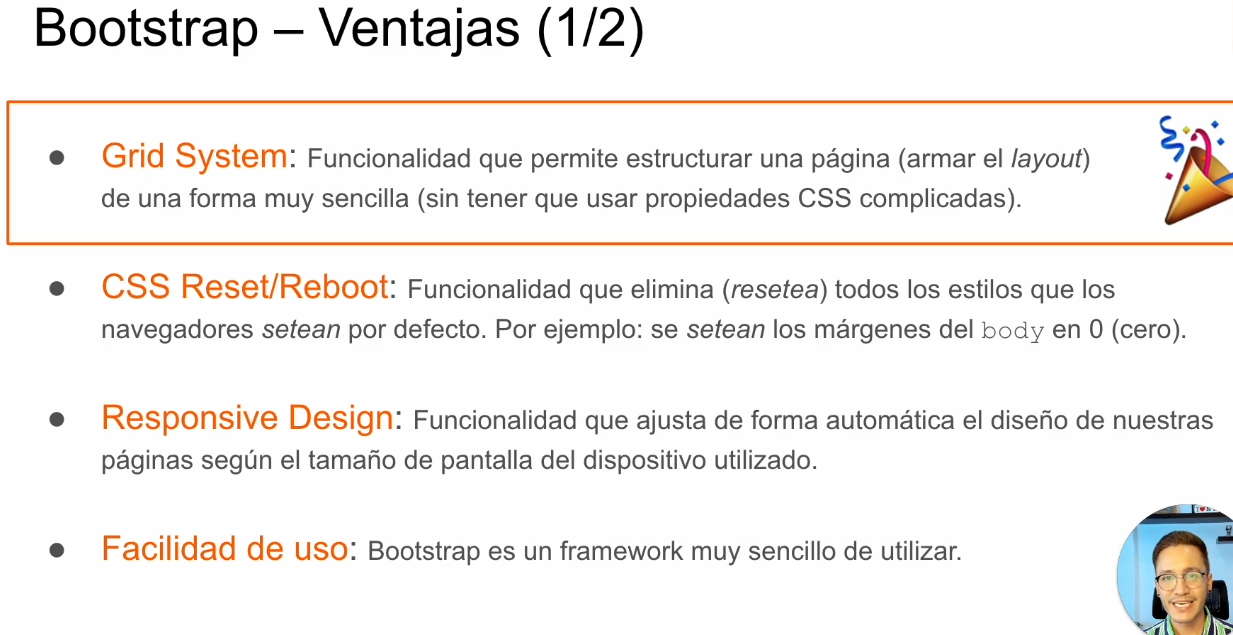
Uno de los frameworks mas usados. Open source. Solo para CSS la dan. En 2020 salió su versión 5.

Framework. Es fragmento de código ya escrito y listo para usar.

Framework no es librería. No entendí bien la diferencia.

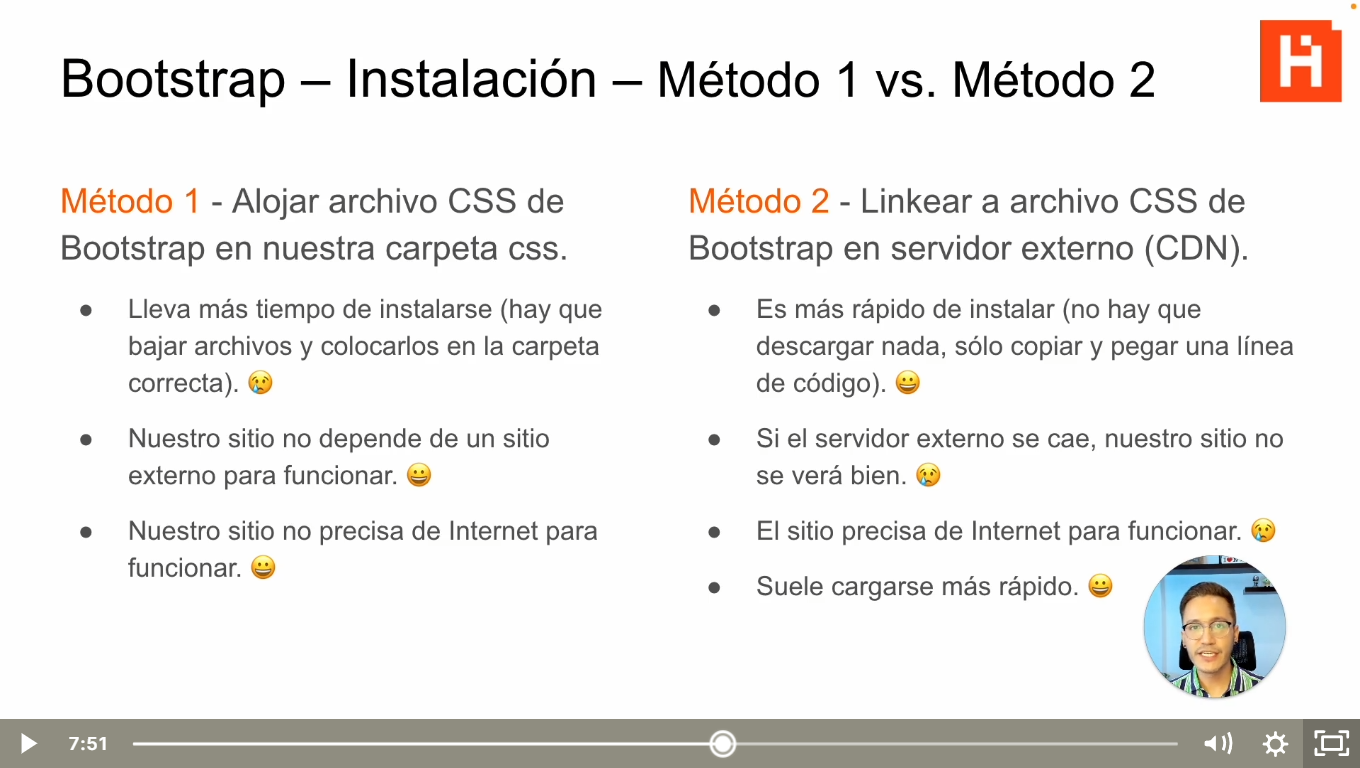


Getbootstrap.com





Hay muchas formas de usar la página. Usan un método de instalación en el que copian la etiqueta al head del HTML.



Copia el CDN:



Y ahí, ya cuando se ponga a escribir el HTML ya aparecerá con un estilo diferente.

Después, copia y pega las líneas HTML de un botón, lo pega y ya queda armado. La parte de CSS está armada en la página de Bootstrap, por eso no se escribe.

Sistema de grillas (Bootstrap)

Bootrastrap da 2 clases: .row (contenedor) y .col

En una fila (row) pueden existir hasta máximo, 12 elementos. O sea, el ancho no puede sumar 12.

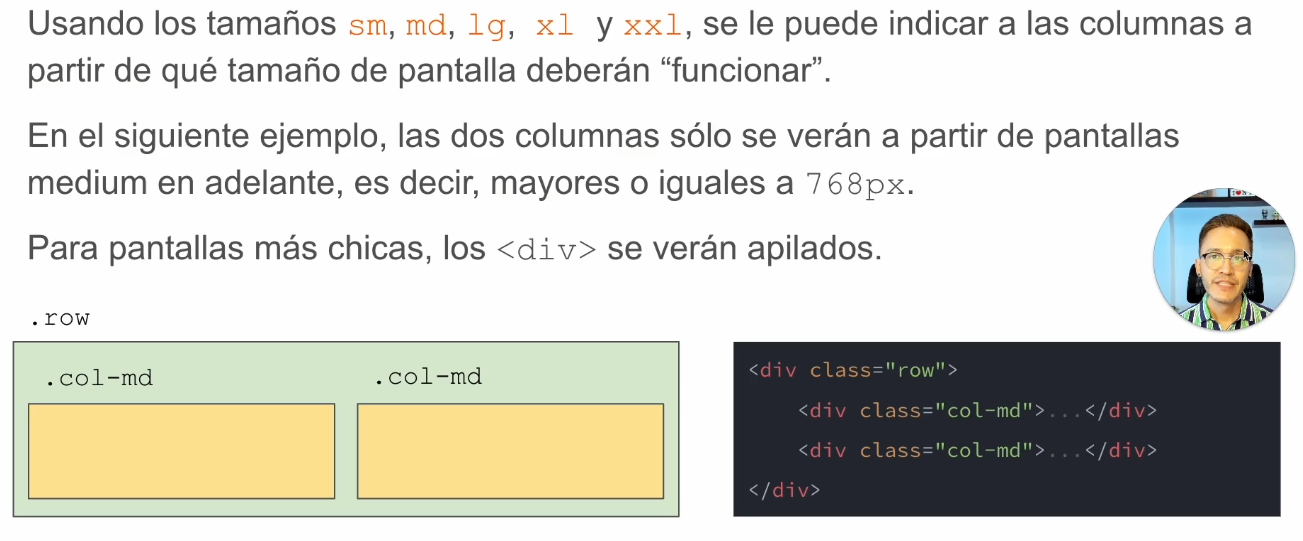


Poniendo col-4 y col-8 definís el ancho de las columnas. La de 8 es el doble de la otra.

El contenido se agrega dentro de las filas, no de las columnas. Adentro de las columnas, solo pueden haber filas.

Se puede meter al .row adentro de un .container que está definido en Bootstrap.

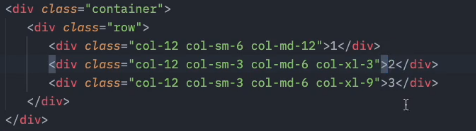
Para que sea responsive, no se puede tocar el CSS. Lo que se hace es:



Los 5 de arriba son los modificadores de tamaño.



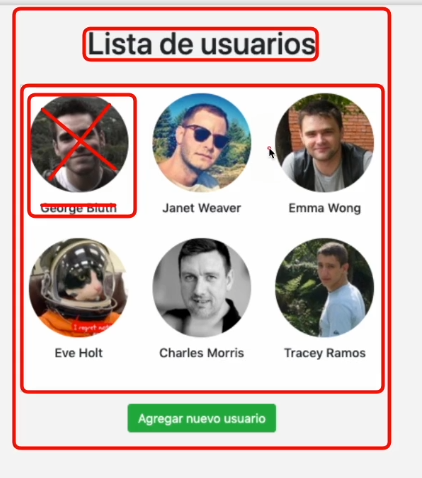
Esos son los mediaqueries que trabaja Bootstrap. Esos números de corte son los breakpoints.



Col-12 implica que cada columna va a ocupar todo el ancho de la fila.

Luego, cuando la resolución sea sm, la primera columna va a ocupar la mitad de la fila, y cada una de las otras dos, un cuarto.

Luego, cuando sea md, la primera va a ocupar toda la fila, y las otras 2 columnas, estarán juntas en la fila de abajo.



Hace cajas (containers) dentro de cajas, dentro del body.

Recomienda free icons “flaticon.es” Para descargar imágenes y tener una demo de lo que se va a hacer.

Siempre hay que asumir que piden mobile first si no aclaran.

Selector descendente:

.container h1

Eso significa que la regla de estilo CSS se aplica al h1 que está dentro de container.

.dibujos img {

    width: 100%;

}

Así, las imágenes que forman parte de dibujos, van a ocupar el 100% de su contenedor.

Font-size Em. Multiplica el tamaño de letra del HTML. Por defecto HTML viene de 16 pixeles, por lo que Em:1; sería 16 px. Em 1.3 sería 20,8.

Flex wrap:wrap. Le dice al contenedor que respete el valor que tengan sus contenidos.

Luego puso al ppio

\* {

Box-sizing: Border-box

}

El asterisco es SELECTOR UNIVERSAL