Curso CSS soydalto.

Recordemos siempre abrir la carpeta completa desde el vs code, porque si no se hace un quilombo barbaro (créanme).

El css es front end, lo que quiere decir que es lo que se ve en la pantalla del usuario y existen 4 formas de utilizarlo para cambiar el estilo de los elementos de nuestra pagina. estilo en línea, estilo definidos por una etiqueta , el estilo con la etiqueta , y el estilo con import que no es recomendable.

Con la etiqueta Style lo que hacemos es trabajar en css como si fuera una etiqueta mas, lo cual lo haríamos de la siguiente manera. .

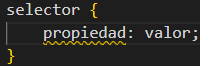
La otra forma, la cual seria con el estilo dentro de las etiquetas, como un atributo, es de la siguiente manera. , en este caso seria para el que texto que contiene el <p> sea de color rojo.

La forma mas optima y que vamos a utilizar es usando la etiqueta , que en este caso se utilizaria de la siguiente manera, 

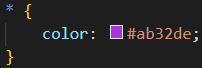
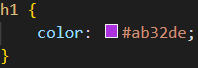
En el que rel seria hoja de estilo, que esta escrita en texto y css con el href donde nosotros vamos a crear nuestra hoja de estilos, la cual, cave aclarar que es un archivo .css

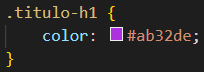
De esta forma lo que hacemos el modulizar nuestro trabajo, haciendo que nuestro codigo no sea un codigo espagueti, que en resumidas cuentas es el que esta todo junto y mal ordenado.

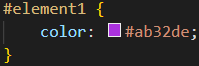
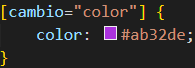
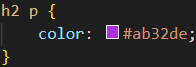
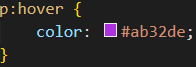
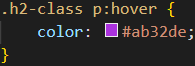
## Selectores

El orden de nuestro archivo css va a ser el siguiente,  donde cada uno de estos 3 items representan lo siguiente.  seria el objeto al que nosotros queremos cambiar el estilo.  seria que le vamos a cambiar, por ejemplo, el color, el tamano, la letra, etc.

Existen diferentes tipos de selectores

* Universal:  que este seria para cambiar todo el archivo y se utilizaria de la siguiente manera.  donde aca vemos, como el selector, seria el , la propiedad que queremos cambiar es  y el valor asignado al color va escrito en lenguaje exadecimal que en este caso es un lila que representamos de esta forma como podemos ver, despues de cada valor, va un punto y coma, en caso de que querramos agregar mas, siempre van separados en punto y coma. De esta forma, si nosotros vemos la pagina de HTML, quedaria de la siguiente manera al seleccionar todos los elementos, el codigo para llamar a ese estilo de nuestra pagina de css seria este. , donde podemos ver que el rel es una pagina de estilo, type, texto css y href donde nosotros tenemos nuestro archivo de css con las caracteristicas visuales que le queremos dar a nuestra pagina.
* De tipo: esta forma de editar seria para cada tipo de etiqueta que usemos, por ejemplo, si nosotros queremos cambiar los titulos de nuestra pagina de color, quedaria de la siguiente manera, , vemos que el selector, es diferente, donde solo se aplica el cambio que nosotros queremos a nuestras etiquetas <h1>.
* Clase>: seleccionar por clase seria agregarle un atributo de la siguiente manera, donde dentro de las comillas, va el titulo con el que nosotros llamaremos a el objeto que queremos cambiarle el estilo. Quedando de la siguiente manera.

Y en nuestra hoja de estilo quedaria de esta forma, como podemos ver, se coloca un punto antes de colocar el nombre que nosotros le asignamos en la .

* ID: Para seleccionar a traves de el ID, lo que hacemos es utilizar el atributo  de esta forma es igual a class, pero en nuestra hoja de estilo lo que usamos es esta forma de llamarlo . Quedando de la siguiente manera. . En este identificador lo que hay que tener en cuenta, es que para utilizarlo correctamente hay que darle un ID a un solo elemento, no porque si se lo damos a varios no funcione, funciona perfectamente, pero cuando nosotros damos un ID se supone que un identificador de este tipo es unico y no deberia repetirse en otro lugar del codigo. Tambien el id con diferencia al class tiene mas rango de especifidad.
* Por atributo: el siguiente selector es a traves de un atributo que nosotros inventemos, puede ser cualquiera, quedando de la siguiente manera, , entonces en nuestra hoja de estilo lo llamamos de la siguiente manera, como podemos ver, el llamado es dentro de corchetes.
* Descendiente: este selector se utiliza llamando a los hijos de algun elemento, por ejemplo un  <p> dentro de un h2, quedando la forma de llamarlo de la siguiente manera, , primero se llama al padre de el elemento y luego al elemento que nosotros queremos darle un estilo. Tambien podemos llamarlo a traves de una class, por ejemplo de la siguiente manera. , podemos ver que nuestro  esta dentro de un h2 con una class=”” con un atributo y nombre h2-class, entonces en nuestra hoja de estilo quedaria de la siguiente manera.  donde vemos que primero le decimos dentro de donde se encuentra el  para luego decirle que le cambio el color.
* Pseudo-clases: las pseudo clases, lo que hace es que selecciona un elemento para que tenga un reaccion a nuestro accionar en la pagina, en esta ocasión lo que usarmos es el elemento  que va en nuestra hoja de estilo y lo que hace es que si nosotros pasamos el mause por encima de el elemento al que le cambiamos el estilo, este cambia de color, pero si no lo hacemos se encuentra por defecto en negro, la forma de utilizarlo queda de la siguiente manera.  ahí le estamos diciendo que a todos los  lo cambie a color lila, cuando nosotros pasemos el mause por encima. Tambien si le queremos dar mas especifidad a el llamado, para que nos cambien un solo , lo hacemos de la siguiente manera,  ahí utilizamos nuevamente el h2 que teniamos con class y lo llamamos asi para que solo cambie de color, nuestro  dentro de ese h2, quedando asi.  ahí vemos que primero ponemos el h2class para luego darle el estilo al  con el hover.

## Especifidad

La especifidad es fundamental en css, por una cuestión de que nosotros al darle estilo a un elemento, podemos aplicárselo a otro sin querer, por cuestiones de que hay muchos que se van a llamar igual o se repiten, la especifidad se encarga de que si nosotros la aplicamos bien, no tengamos ese tipo de conflictos en nuestro código.

Para comprender mejor la hoja de estilo, hay que saber que trabaja en efecto cascada, donde el navegador lo que hace es tomar la ultima modificación en orden descendente de el elemento que nosotros modificamos, por ejemplo, si nosotros hacemos 3 modificaciones a un , siempre va a tomar la que se encuentra mas abajo en nuestra hoja de estilos. El orden de especifidad es el siguiente. Siempre esto va a funcionar de la siguiente manera, si le damos 2 estilos a un mismo elemento, siempre va a tener efecto el ultimo cambio que hagamos en orden de cascada. Pero si utilizamos el orden gerarquico que vemos aquí debajo, el orden de cascada no se tiene en cuenta, salvo que utilizemos un mismo nivel de ordenamiento, en ese caso, vuelve a ser en efecto de cascada, siempre el ultimo, es el que va a modificar al elemento que nosotros le dimos estilo.

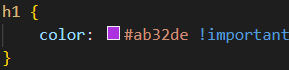


Dandole importancia de arriba hacia abajo, lo menos importante es este caso seria, los elementos y pseudo-elementos que nosotros vimos antes, que serian el <a> <b>.

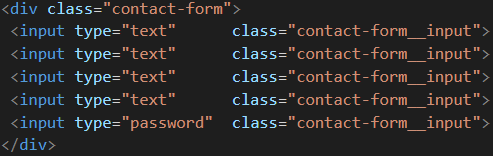
Luego vendrían las clases, las pseudo clases y los atributos.

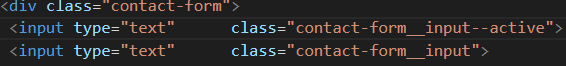
Luego vienen los ID, que es cuando nosotros le asignamos un identificador a un elemento.

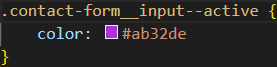
En el siguiente orden tenemos los estilos en línea, que son los que se aplican desde el mismo código html, de la siguiente forma. Dándole un Style como atributo dentro del elemento que nosotros queremos modificar.

El nivel mas alto de gerarquia, es utilizando el  en nuestra hoja de estilo de la siguiente manera , esto lo hacemos y no tiene ningún valor cualquier otro atributo o llamado que tengamos en nuestra hoja de estilos, o modificación en línea, siempre el  va a ser el que tiene el orden jerárquico mas importante. No es tan recomendable usarlo, ya que si le damos esta orden, no vamos a poder modificarlo mas de ninguna manera, es mejor aplicar bien el orden jerárquico de las otras formas a tener que utilizar este.

## Metodología BEM

La metodología BEM, seria una forma ordenada de programar y diferenciar elementos para que a la hora de darle un estilo, tengamos ese elemento solo con una sola clase, esto se trabaja por bloques, por ejemplo de la siguiente manera, si yo hago un bloque, le pongo a todo la misma clase.

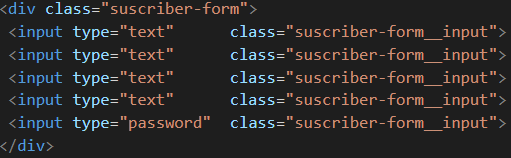
Como podemos apreciar en la imagen, cada elemento de nuestro div, contiene la class contat-form seguido de 2 guiones bajos y el nombre de la etiqueta que estamos utilizando, esto es la llamada metodología bem, si nosotros quisiéramos cambiar el color de algo de solo uno de esos imput, lo que deberíamos hacer es lo siguiente. 

Entonces si nosotros solo le queremos dar un estilo especifico a ese input, lo que hacemos en nuestra hoja de estilos, es llamarlo como lo tenemos ahí, quedando de la siguiente manera. 

Y para cambiarlos a todos, seria de la siguiente manera.

 de esta forma, cambiariamos el color de todo el div que tenga la clase .

Si yo tuviera otro div similar, deberiamos hacerlo con otra clase, por ejemplo de la siguiente manera.



Cabe aclarar que hay muchas metodologias y siempre es recomendable utilizar alguna para darle un ordenamiento mejor a nuestra web y muy util a la hora de modificarla con CSS.

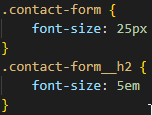
## Teoria de las medidas

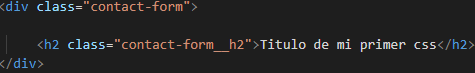
En css existen 2 tipos de medidas, las fijas y las relativas.

Las relativas son las que dependen de algo a la que la quisiéramos aplicar y las fijas son las que se atienen a medidas, como cm, milímetros, pixeles, puntos, etc.

La tipografía por ejemplo, tiene varias formas de medida, mm, px, cm, etc. Que estas serian medidas fijas, pero también si las hacemos relativas, nos sirven mucho mas porque se adaptan al tamaño de dispositivo que nosotros estemos usando en el momento. Em (un em por defecto es 16px) es una medida relativa. Este tipo de medidas, las relativas, hacen que nuestro projecto, sea responsive design, que quiere decir que se adapta al entorno donde nosotros lo estemos utilizando.

## Unidad de medida relativa EM

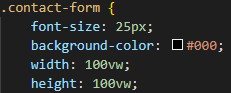
Para modificar el tamaño de la letra, lo que hacemos desde nuestra hoja de estilo es lo siguiente. , en este caso, utilizamos una medida fija que son los px. Si queremos utilizar una medida relativa, lo que hacemos es utilizarla de la siguiente manera, , como dijimos antes, cada em por defecto de navegador, equivale a 16px, pero si nosotros queremos asignarle un valor a nuestro em, lo que hacemos es heredar la caracteristica desde arriba, entonces, quedaria de la siguiente manera. como podemos ver, en el contact form inicial le dijimos que el tamanio de la letra sea 25px, al heredar esa caracteristica nuestro em, lo que hace es que cada em valga 25px por ende, el tamaño de nuestra fuente es este caso, terminaria siendo 125 pixeles.

Esto quiere decir que nosotros en esa caja contenedora, le estamos asignando un valor a em, entiendase una caja contenedora a lo siguiente. , como podemos ver, mas arriba, le decimos a contact form que el tamanio de letra sea 25px y luego le decimos que el tamano de el h2 sea 5em dandole a entender que 1em vale 25px.

## Unidad de medida relativa viewport.

Lo que hacen estas medidas, es ocupar todo el ancho de la pantalla, sea cual sea el tamaño va a depender de donde se lo esté ejecutando.

Para que esta medida sea correctamente utilizada, lo que hacemos es desde nuestra pagina de estilo, utilizar las características siguientes, , acá podemos ver que width es el ancho de la página, y height es el alto, si a estos valores nosotros le asignamos 100 viewports, lo que el navegador interpreta es que es todo el ancho de la pagina que utilizamos.

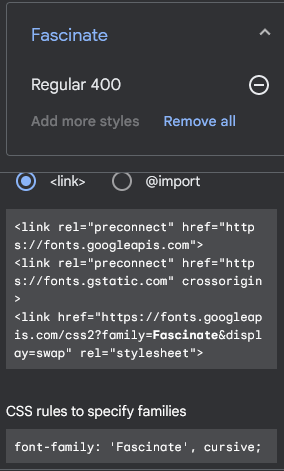
Si utilizamos 50vw lo que va a hacer es utilizar el 50% de la pagina y asi para todos los casos. Quedando utilizado de la siguiente manera en nuestra hoja de estilo. .

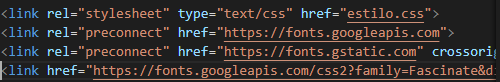
## Estilo y propiedades de texto

Estas son las propiedades de texto que vamos a ver a continuación.

 seria el tamanio de la letra, el que como podemos ver esta en em para que sea una medida variable, pero también podría ser en px, cm, pts, etc.

 es el estilo de letra que va a utilizar, por ejemplo, arial, Georgia, etc. Para conseguir mas tipografías, lo que hacemos es entrar en <https://fonts.google.com> donde ahí nos salen muchísimas fuentes disponibles, seleccionamos alguna y nos dice que copiar en nuestro header y el estilo con el que la llamamos, quedando de la siguiente manera.



Nosotros copiamos los links en nuestro header de la siguiente manera. 

Y luego aplicamos la fuente que queremos desde nuestra hoja de estilos, con el nombre indicado en css rules.  cave aclarar que se pone con una coma, para que en caso de que la pagina no encuentre la fuente que nosotros seleccionamos va a pasar a la siguiente, la cual seria “cursive”. De esta forma podemos aplicar cualquier fuente que veamos en la pagina.

 sería el espacio que ocupa la letra, siempre naciendo desde el medio, sería como el interlineado o el espacio que ocupa a la siguiente línea, en todo caso seria el espacio en blanco que queda abajo y arriba de nuestra letra, como dijimos mas arriba naciendo desde el medio, hay que tener en cuenta que 1 seria 0 espacio, ya que es el tamaño de la letra.

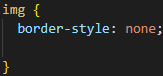
 se enumera de 100 a 900, esto lo que hace ensancha o afina el grosor de el trazo de la letra que nosotros estemos usando.

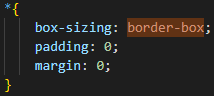
 que seria como ya vimos mas arriba, el color que querríamos utilizar, también deberíamos googlear el tipo de color y código que lleva.

## Normalize y teoría de cajas

Los navegadores traen configuraciones por defecto. Para que todas estas características vuelvan a 0, lo que debemos hacer es utilizar el <https://necolas.github.io/normalize.css/>. Que se aplica de la siguiente manera, se puede descargar desde la misma pagina el archivo css  lo que ponemos en nuestra carpeta de proyecto y lo linkeamos a nuestra pagina dentro de el <head> de la siguiente manera , de esta forma, nuestras páginas, quedaran sin ninguna característica automática de el navegador.

Una vez tenemos el archivo normalize.css de manera local, lo que hacemos es modificar algunas cosas, para darle ciertas características.

Lo primero es ir a este estilo dentro de el normalize.css y modificarlo , agregándole la característica , para que las imágenes en móviles las ajuste al 100% y no se nos pase de los bordes del mismo.

Otro elemento que hay que tener en cuenta, es el , que lo que hace es que cuando nosotros creamos una caja con un <div> y le damos un tamaño especifico, al agregarle cosas dentro, el tamaño se mantenga, ósea que lo que pongamos dentro se ajuste a nuestras especificaciones. Lo que pude ver intentando hacerlo, es que tenemos que hacer una característica nueva nosotros mismos en la stylesheet, con las siguientes características. 

De esta forma lo que podemos decir, es que nuestro archivo esta al 100% sin modificar.

## Teoría de cajas

Existen 2 tipos de cajas en html, las que son en línea, en bloque que serian las cajas naturales, y también existen las box model.

Las cajas que son en bloque, se caracterizan por ocupar todo el ancho de la caja que la contiene, no importa el tamaño de la caja, ya sea de toda nuestra pagina o de un bloque mas pequeño esta caja va a ocupar todo el ancho disponible dentro de la otra caja.

Los bloques que son en línea, el espacio que va a ocupar es solamente el que utilice lo que nosotros le pongamos dentro, ósea que, si nuestra caja es de 100px y el bloque en línea solo ocupa 20px, ese será el tamaño del bloque. Lo que quiere decir es que podemos poner varios bloques en línea seguidos, ya que su tamaño al ser de 20px en este caso, podríamos poner 5 dentro de nuestra caja.

Si nosotros queremos que un elemento en bloque, pase a ser un elemento en línea, lo que demos hacer, desde nuestra stylesheet, es modificarlo con la propiedad de esa forma el elemento pasara a ser en línea.

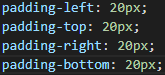
Otra diferencia entre estos 2 tipos de cajas, es que a los elementos que son en línea, no le podemos dar la característica , ya que solo se puede utilizar en los elementos en bloque.

## Propiedades de las cajas

Son las que modifican o aplican algún cambio a nuestras cajas.

Algunas de las propiedades que podemos darle son las siguientes.

 con esta característica lo que hacemos es cambiar el fondo de la caja que creamos, en este caso un h2, que al ser un elemento en línea, nos ocupa el fondo, todo el ancho de la caja completa, nosotros en este caso le agregamos la propiedad  que hace que el fondo solo se aplique a el texto que nosotros escribimos, lo que hace que sea un elemento en bloque.

Luego tenemos el padding, que lo que hace es darle un margen al texto con respecto a la caja, tanto de arriba, abajo, derecha e izquierda utilizando la siguiente propiedadsiendo cada una de estas, la correspondiente.

En caso de que queramos el mismo padding en todos los lados, lo que podemos hacer es utilizar la propiedad de la siguiente manera.  con solo usar esa, aplicamos los 25px a los 4 lados antes mencionados.

Si queremos ahorrar todo el tema de el padding-left-right-bottom-top, lo que hacemos es utilizar el padding de la siguiente manera  siendo el primero el top, luego el rigth, bottom y por ultimo el left. Siempre tienen ese orden si nosotros no le indicamos cual es. Osea que seria en castellano, arriba, derecha, abajo e izquierda el ordenamiento, tambien podriamos decir que es en el sentido de las agujas del reloj comenzando desde arriba.

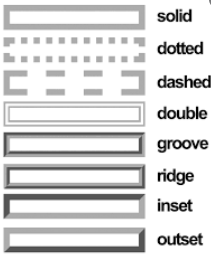
Si solo ponemos 2 medidas de padding, que hace es que toma los ejes X e Y, siendo Y la primer medida que pongamos y X la segunda, osea que Y seria el padding vertical tanto arriba como abajo y el padding X seria derecha e izquierda.

 esta propiedad lo que hace es el margen o distancia que queremos darle a esa caja, por ejemplo alejarlas, de la siguiente manera  ahí vemos que con margin las separamos, de no tener esta propiedad las 2 cajas quedarían de la siguiente manera, y el motivo porque el margin de el medio es un poco mayor, por decir un poco es decir el doble, porque se aplica 10px de margin a cada caja, entonces entre caja y caja hay 20px.

Para aplicar caracterisiticas a cada margin individual, se utiliza igual que el padding, en orden arriba, derecha, abajo e izquierda. De la siguiente manera.

 esta característica lo que hace es hacer redondeado el borde, si le damos px va a quedar un rectángulo de esquinas redondeadas pero si aplicamos % lo que va a pasar es que sea va a ir haciendo un circulo.

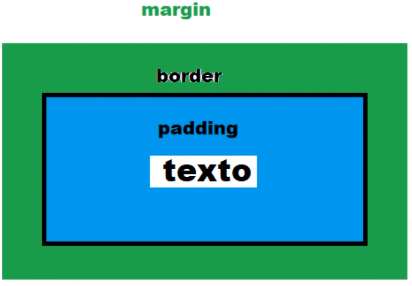
 esta propiedad lo que hace es darle un borde a nuestra caja, existen muchos tipos de bordes, pero lo que vemos ahí es, 5px de ancho el borde, el estilo del borde y el color. De esa forma lo modificamos.

 aca vamos a dejar una serie de bordes para tener.

 esta característica lo que hace es darle una sombra a la caja, el orden de las características es eje X, eje Y, tamaño del desenfoque (que seria el largo de la sombra), cuanto borde va a tener (por Gral. se le da 0) y por ultimo el color. Esta característica se puede utilizar con el texto también, si usamos la propiedad , hay que tener ojo que esta no tiene propiedad de borde, si la colocamos, anula a el resto de características y no va a hacer el sombreado.

otra característica que podemos modificar es la rotación, se utiliza esa propiedad, en la que vemos el tag, la rotación y los grados que la queremos rotar.

Por último lo que queda observar es que estas características se deben hacer en un orden determinado, para que queden con una prioridad correspondiende y seria de la siguiente forma  como primer característica, luego el , después el  y por ultimo el . Se hace de esta forma debido a que se llama “box Model” en el que la caja se va haciendo desde adentro hacia afuera.



## Outline

Como vimos antes, el outline, es otra propiedad acortada