

Hola, vamos a ver un concepto que es importante que entiendan, bueno, en realidad todos son importantes, ¿no?. ¿Qué ocurre cuando tenemos un elemento y a ese elemento podemos acceder de diversas maneras y poner un estilo? Y un estilo podría ser contradictorio, es decir, podríamos poner el mismo estilo, pero con distintos valores, o sea, la misma propiedad con distintos valores, y de acuerdo a cómo seleccionemos el elemento, se aplicaría uno y no otro. Estas cuestiones las vamos a resolver a continuación.

A nivel técnico, podríamos decir que hablaremos sobre la precedencia o importancia de las declaraciones, ¿correcto? Comenzamos en nuestro código simplemente con un HTML, un "index.html" que sólo tiene aquí, bueno, pues un título en la pestaña y un título aquí general para hablar sobre el tema de esta clase de hoy, ok, hemos vinculado nuestro archivo "index.html", con un archivo CSS, que está ubicado en esta carpeta, aquí les voy a mostrar la estructura tenemos la carpeta CSS, con el archivo "Style.ss" dentro y eso es lo que estamos llamando aquí ya hemos vinculado los archivos, por lo tanto estamos preparados para darle estilos, utilizamos el "reset" de margen y "padding" en el comienzo. Noten que si comentamos esta línea, es decir, ahora esto es un comentario, no se va a leer como código noten que aquí vamos a tener un cambio porque los navegadores no todos igual, pero todos si tienen su propio padding y su propio margen por defecto por eso reitero esta línea es tan común, es tan común utilizar esta este receta de "padding" y de "margin" cuando comenzamos a trabajar. Ahora sí, le quito el comentario y va a ser tenido en cuenta por lo que es, como código CSS. Y volvemos a quedarnos sin ningún "padding", ningún margen, por defecto. Antes de hablar sobre precedencia de las declaraciones, me gustaría hacer esto un poquito más prolijo, nada más, vamos a seleccionar nuestra etiqueta "h 1", mes esta que está aquí, la única que tengo en la página por ahora y vamos a centrarlo. Hay una propiedad que se llama "text align", o sea alinear el texto, vamos a enviarla al medio y ahí está, en el medio, podríamos separarla un poco desde arriba, ¿verdad?, claro que sí, como no, démosle un "margin top", por ejemplo, vamos a utilizar medidas relativas, "232 rem" equivale, en este caso, a 32 píxeles, porque un "rem" equivale por defecto a 16 píxeles. Bueno, ahí está un poquitito más prolijo.

Y luego vamos a aplicar el tema de la precedencia de las declaraciones sobre un texto, pero no quiero el texto suelto, sino que lo voy a poner dentro de una caja. En CSS todo es una caja, por eso hablamos del "box model", el modelo de cajas. ¿De dónde viene esto? Bueno, miren, si inspeccionamos nuestro código, vamos a hacer todo un poco más grande, noten que cuando elijo, por ejemplo, alguno de estos elementos, fíjense que aquí todo me lo muestra como si fueran cajas, en este caso rectángulos. Podrían ser cuadrados también, por eso hablamos de cajas. Cada elemento es una caja. Bueno, ahora tengo uno solo, ¿verdad?, pero cada elemento es una caja, bien, de aquí viene entonces hablar de el modelo de cajas, o "box model" cuando hablamos de CSS. Los diseñadores de páginas web, siempre ven todo como una caja, porque efectivamente, cada una de estos elementos termina siendo, en el fondo, una caja, o mejor dicho, termina pareciéndose a una caja. Ok, entonces pues, hagamos un contenedor que iba a ser simplemente una caja un elemento "div", perdón, estoy en el archivo CSS, tengo que ir al HTML, aquí debajo pongamos un "div" y un "div" es, simplemente, una caja contenedora que no tiene una estructura, ni un peso semántico. Puede haber aquí adentro cualquier cosa, es indistinto y este tipo será el que contenga el texto sobre el que vamos a trabajar aquí.

Me dan dos estilos, ¿ok?, obviamente en nuestra caja contenedora, nuestro "div" apenas si logra verse y ahora vamos a ver porque en principio nuestro div vamos a elegirlo porque hay

uno solo así que bueno no voy a tener conflictos con ningún otro, vamos a darle un color de fondo. Ahora para ver exactamente qué espacio está ocupando, sólo para eso, bien, es un elemento de bloque, por lo tanto como dijimos, antes ocupaba el ancho completo de mi pantalla y el alto está dado por la cantidad de texto que contiene como el párrafo, también es un elemento de bloque, solamente tengo lo que ocupa el párrafo en la altura de mi caja vamos a darle un tamaño por defecto a nuestra caja, por ejemplo, hagamos un cuadrado, un cuadrado que tenga 400 píxeles de ancho x, bueno, 400 píxeles cuadrados, es decir, 400 de ancho por 400 de alto.

Bien y ahí ya cambia un poquito la cosa, porque estamos indicándole un tamaño a esa caja contenedora que va más allá de lo que reclama el elemento que está contenido dentro del "div" que es el párrafo, ok. El párrafo reclama sólo este tamaño, pero estoy diciéndole a mi caja, "bien, tú tienes esta medida independientemente de lo que contenga", me gustaría ponerlo un poquito más en el medio de esta caja, eso podemos hacerlo con la propiedad "margin" y vamos a poner un poco de margen desde arriba. Pongamos unos a 30 píxeles y, de izquierda y derecha, pongamos la propiedad "auto", esto lo que va a hacer es distribuirme en partes iguales el espacio remanente y va a terminar centrando nuestra caja en la pantalla, bien, centrándola horizontalmente, correcto un poco más de margen desde arriba no vendría mal, ahora sí, muy bien.

Nosotros queremos aplicarle a este párrafo los estilos, está un poco pequeño, así que para que se vea bien, vamos a indicar algunos estilos para ese párrafo y recuerden, a pesar de que es el único párrafo en mi HTML, puedo ser más específico, puedo indicar qué voy a apuntarle al párrafo, que esté únicamente dentro de un "div", eso es más específico y aquí podría también decir "margin auto", perdón, aquí adentro no puedo establecer el margen de esta manera, lo que voy a hacer con el texto, es indicarle por que es un texto y hay una propiedad específica para texto que ya la usamos aquí arriba que siempre está line vamos a indicar "text align", "center", ahí tenemos el párrafo en el medio podríamos poner un margen desde arriba, ok, podríamos indicar el margen desde arriba, vamos a ponerle un valor específico, no, perdón, estoy trabado con el "auto", quedé con el "margin auto" desde la caja anterior. Vamos a indicarle unos 100 píxeles desde arriba, bueno en realidad 200, porque mi caja tiene 400 de alto, así que esto le va a dejar aquí, aproximadamente, en el centro. ¡Ajá!, pero le estoy aplicando también los estilos de esta forma a mi "deep" contenedor y no quiero hacer eso, por lo tanto, esta propiedad no la voy a aplicar de esta manera, la vamos a aplicar de otro modo, pero lo primero, porque nos estamos yendo por las ramas, pero no es culpa de ustedes, obviamente, en mi culpa. Vamos a ir al tema de la clase que es la precedencia de las declaraciones, si quiero aplicar un estilo efectivamente a este párrafo, podríamos hacerlo de varias maneras, podríamos indicar, vamos a quitar de aquí y vamos a indicar "tamaño de letra", un "font size". El "font-size" va a ser, vamos a usar las que ya están definidas aquí, la propiedad de predefinidas, "extra large", con esto me va a alcanzar, ¿qué quiero mostrarles?, bueno, pues, lo siguiente: ya hablamos sobre la cascada, ahora imagina en esto tenemos un párrafo aquí, un selector de párrafo que indica estas propiedades y luego tenemos otro selector de párrafo que indica las mismas propiedades, pero además vamos a añadir un color por aquí, "red" y aquí arriba también vamos a poner un color, de color "white", bien, ¿cuál de estos dos colores?, estas propiedades no van a tener conflictos, porque son iguales, por lo tanto, no va a haber un problema aquí, pero aquí tengo, para el mismo selector, es decir, para el mismo párrafo, en rigor, de verdad, estoy seleccionando todos los párrafos que haya en mi documento, primero digo "color blanco" y luego digo "color rojo" y, ¿de qué color va a quedar al final?, vamos a develar el misterio. Guardamos el archivo y es

rojo, ¿por qué?, por que si tenemos dos elementos, dos electores iguales, siempre se aplica el último, siempre se va a aplicar el último y como en el último tengo color rojo, ese es el color que va a quedar, ¿ok? Bien, por supuesto, si en el último no tuviese con los rojo, el texto sería blanco, no el negro por defecto, porque a pesar de que este segundo selector está luego del primero, bueno, lo cual está así de pero grullo, verdad, siempre el segundo viene después. El primero no tiene una propiedad de color, por lo tanto, no hay conflicto, no se sobre-escriben. Las propiedades cuando no hay conflicto entre ellas se suman, se agregan, pero cuando hay conflicto es decir cuando tengo dos propiedades iguales, se aplica siempre la última, ténganlo presente, siempre se aplica la última.

Ok ahora quiero mostrarles lo siguiente: dejemos seleccionado nuestro párrafo de esta manera, pero además, vamos a ponerle, para explicar ciento por ciento la cuestión de la presidencia de las declaraciones un "ID" al párrafo y también una clase, le puedo poner el mismo nombre, no importa, por qué, porque esto es un "ID", es una categoría y esto es una clase, es otra, por lo tanto no hay problema. Ahora vamos a poner que el "ID" o mejor dicho, vamos a empezar con la clase, recuerden que para seleccionar dentro de CSS por nombre de clase, tengo que poner el nombre antecedido por el prefijo punto, entonces: punto, párrafo, y esto va vamos a vamos a jugar con el color aquí. El párrafo va a ser de color "violet", vamos a cambiar este color aquí, porque realmente no tiene un buen contraste, cierto, un poquito más claro o mejor dicho, un "deep pink". Bien, ahora si, seleccionamos el párrafo por clase. Vamos a poner un color distinto, pongamos un color por ejemplo, "blue" y ahora se aplica el "blue", ¿por qué?, porque la clase, el selector de clase, se impone al selector de etiqueta, al selector de, digamos, que lleva el nombre de la propia etiqueta. Fíjense que ahí me muestra la especificidad del selector, es un dato técnico, pero ténganlo en cuenta, ¿ok? Fíjense que aquí es 0 0 1, aquí es 0 1 0 y ahora vamos a llamar a nuestro "ID", "párrafo". Recapitulemos, tenemos un párrafo, el párrafo tiene un "ID" y el "ID" lleva ese nombre y tiene una clase con el mismo nombre, pero recuerden, no importa esto, porque son cosas distintas, categorías distintas, es decir, siempre estamos apuntándole a esto, a este texto, primero por el nombre de etiqueta, después por la clase y ahora vamos a apuntarle al "ID". Vamos a decir que el color es "red" y ahora es rojo. Esto les habla de la especificidad, se aplica el estilo de párrafo, mejor dicho, bueno, sí, de párrafo, pero me refiero al nombre de la etiqueta. Si hay una clase aplicada, la clase se impone a la etiqueta, pero si hay un "ID", el "ID" se impone a la clase y a la etiqueta. Esa es la presidencia de las declaraciones y hay una última que, bueno, tratamos de no utilizarla, pero se las voy a mostrar, porque por supuesto existe y se llama "estilo en línea", podemos aplicar un estilo directamente en el elemento, ¿cómo aplicamos un estilo en línea?, directamente al elemento, bueno, con el atributo "style" que tiene la siguiente sintaxis, bueno, como todos los atributos de HTML, el nombre del atributo, igual, y entre comillas, los valores, y aquí vamos a poner color, a ver, ¿qué color podemos elegir?, distinto al que ya hemos elegido antes, bueno, vamos a volver a poner "black" y ¿qué ocurre ahora?, se aplica el color que indicamos aquí, entonces, cuál es la precedencia de las declaraciones que se aplica, bien, podríamos resumirlo con este diagrama: primero el estilo aplicado a la etiqueta, el estilo aplicado a la clase, sobre-escribe el estilo aplicado a la etiqueta, el estilo, el estilo aplicado con un "ID" se impone a los otros y si aplicamos un estilo en línea, que reitero no es una buena práctica, pero puede hacerse, sobre-escribe todos los demás selectores.

Bien, espero que haya sido lo suficientemente claro para explicarles la cuestión de la precedencia de las declaraciones.