Introduccion a CSS

Durante el desarrollo de este curso, **CSS y Diseño Web Responsive**, aprenderemos a utilizar la hoja de estilos en cascada (CSS) para brindarles a nuestras páginas y aplicaciones Web estilo, forma, color, posición y demás elementos que conforman la estética de una página web.

1. Identificar el funcionamiento del **flujo de cajas** en la estructura HTML.
2. Aplicar el **estándar** CSS.
3. Aplicar **propiedades** de forma a la estructura.
4. Utilizar propiedades de **transformación y efectos** especiales.
5. Aplicar propiedades de **animación** de la estructura.

**Hoja de estilo en cascada**

Hoja de estilo en cascada o CSS por su sigla en inglés (Cascading Style Sheets) es un mecanismo simple que sirve para adicionar estilos a un documento web con el fin de modificar su presentación. Al diseñar una página web empleamos el HTML para definir la estructura del documento y CSS para aplicar el formato gráfico.

Con CSS especificamos la manera en que deben renderizarse los diferentes elementos de un documento HTML en las pantallas, los medios impresos o en cualquier otro dispositivo. El estándar CSS cuenta con diversas funcionalidades tales como: un conjunto de selectores avanzados, la relación con tipografías web, propiedades orientadas hacia el mejoramiento de la accesibilidad de usuarios con discapacidades, opciones para manipular elementos HTML en 2 y 3 dimensiones, herramientas para generar animaciones sin usar lenguajes de script, modos de color avanzados e integración con la mayoría de los navegadores.

Las especificaciones del estándar CSS son definidas por el World Wide Web Consortium (W3C) y están orientadas por principios que buscan una web para todos con acceso desde cualquier dispositivo, estipulando una serie de metodologías y estándares para mejorar la experiencia de los usuarios web desde cualquier parte del mundo.

En la actualidad, las opciones del CSS puede ser extendidas y potencializadas a través de preprocesadores y postprocesadores CSS, de la mano de herramientas de automatización de procesos (Gulp) y de frameworks CSS (Bootstrap).

## ****TENDENCIAS****CSS

** Movilidad web**

**Responsive Web Design (Diseño web adaptativo)**  
Consiste en una serie de técnicas y metodologías que buscan que un contenido web se vea correctamente en cualquier dispositivo, a través del uso de media-queries (módulo CSS3). Una media-query permite controlar la presentación de un mismo contenido HTML a partir de condiciones basadas en el ancho, el alto y el color de los dispositivos de salida.

**Mobile first**  
Es una metodología que hace parte del diseño web adaptativo y que se usa para desarrollar los sitios web de manera que se rendericen en condiciones óptimas, primero en dispositivos móviles y luego, según el tamaño, en pantallas de computador.

**Frameworks y librerías CSS**  
Los frameworks son entornos de desarrollo o de trabajo que cuentan con múltiples opciones de herramientas que podemos usar según las necesidades de cada proyecto. Utilizar un buen framework CSS mejorará el resultado final de nuestros desarrollos y al mismo tiempo nos permitirá ahorrar cientos de líneas de código. En la actualidad, la mayoría de los frameworks CSS tienen incorporados sistemas de grid, así como decenas de componentes preconfigurados, tipografías web, entre muchas otras opciones. Algunos de los más populares son Bootstrap, Less Framework, Skeleton y Foundation.

** Multimedia**

**3D**  
La especificación CSS3, que reúne las transformaciones CSS 2D, CSS 3D y SVG, permite construir reglas para poder aplicar a los elementos transformaciones en espacios de dos y tres dimensiones.

**Animate**  
Este módulo de CSS permite animar los valores de las propiedades sobre una línea de tiempo, usando keyframes. Con él, puede especificarse la duración, el número de repeticiones y el comportamiento de los keyframes. Existen librerías especializadas que nos permiten crear animaciones complejas sin necesidad de emplear lenguajes de script.

** Accesibilidad**

Accesibilidad web se refiere a las herramientas y metodologías que tienden a facilitar el acceso de cualquier persona a la web, independientemente del tipo de hardware, software, infraestructura de red, idioma, cultura, localización geográfica y capacidades de los usuarios.

El W3C cuenta con un proyecto llamado **Iniciativa de Accesibilidad Web** que desarrolla y propone actividades que motivan el acceso a la web de personas con discapacidades. El CSS tiene un conjunto de propiedades que llama “aurales o auditivas” y que utiliza una combinación de síntesis de voz y efectos de sonido para hacer que el usuario escuche en vez de tener que leer la información.

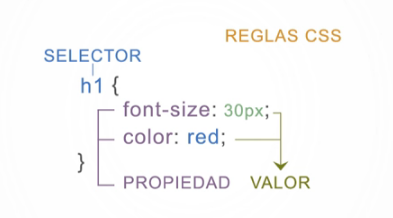
## ****QUE es**** CSS

Html5 nos ayuda a hacer la estructura y contenido.

CSS es la Presentación y aspecto. Es un conjunto de reglas que determinan como lucen los elementos.

Los estilos se definen en una hoja independiente (buena practica) que luego se vincula desde el html

El trabajo en CSS se realiza a través de selectores (formas de referenciar etiquetas o elementos del DOM)



## Validador W3C

Un recurso interesante para los desarrolladores web son los validadores que son herramientas que permiten verificar que el código HTML o CSS sea correcto, es decir, que cumpla con los lineamientos y recomendaciones de la Word Wide Web Consortium (W3C). Este organismo, formado por varias organizaciones, tiene como misión establecer estándares para la web, publicar directrices para crear webs usables y accesibles. Una de las más importantes ventajas de seguir estos estándares es que nos garantiza la interoperabilidad entre navegadores.

La W3C provee una herramienta para realizar las validaciones de HTML y CSS  <http://validator.w3.org/>



1. **Validate by URI:** Es posible validar el HTML y CSS de una URI que ya exista utilizando la opción
2. **Validate by file Upload:**Si aún no se encuentra el recurso que quieres validar en internet, es posible cargarlo y aplicarle la validación haciendo uso de la opción
3. **Validate by Direct input:** En esta opción es posible validar una porción de código.

Cuando validas tu código aumentas su interoperabilidad, te aseguras que la visualización sea la correcta y deseada en la mayoría de los navegadores y facilitas el mantenimiento debido a que cumpliendo los estándares tú código es mucho más legible. Cada día los navegadores actuales están convergiendo y comprendiendo la importancia de los estándares establecidos por la W3C.

## Cómo funciona y sintaxis

### **Hoja de estilos como atributo, etiqueta y archivo externo**

Presentacion de etiquetas html. Hay varias formas

FORMA 1, agregando **atributo** style dentro del código HTML. (estilos inline, es decir, en la misma línea de codigo)

<p> Hola </p>

<p style=”color:red”> Hola </p>

<p style=”color:red; font-family:helvetica”> Hola </p>

Con este método solo afectamos una etiqueta puntual (a la que le estamos poniendo el atributo unicamente)

FORMA 2, a través de **etiqueta** style que ubicaremos en etiqueta head, luego de la etiqueta title.

<style>

h1

{

color: blue;

font-family: helvetica;

}

p

{

color: red;

font-family: helvetica;

font-size: 5px

}

</style>

Con este método afectamos a todas las etiquetas. Si queremos que una etiqueta tenga mas de un estilo, basta con realizar este proceso teniendo en cuenta el nombre de las etiquetas

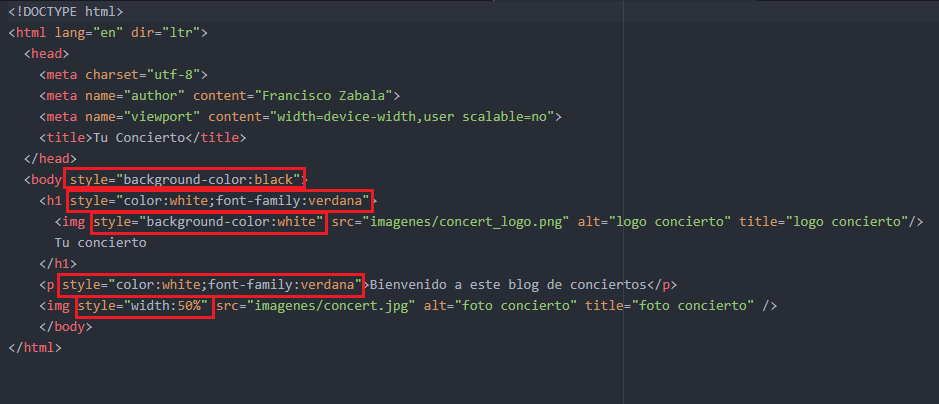
FORMA 3, a través vinculación con archivo externo CSS

1. Primero hay que agregar a nuestro archivo, una carpeta denominada “CSS” que será donde guardaremos las hojas de estilo.
2. Luego en ATOM crearemos un archivo adentro de esa carpeta “CSS”. El archivo que crearemos lo denominaremos estilos.css
3. En ese archivo denominado estilos.css pondremos los códigos de estilo (aquellos que estén comprendidos DENTRO de ETIQUETAS <style>)
4. Vincular desde nuestro código HTML, en etiqueta head luego de title. De esta manera:

<link rel=”stylesheet” href=”css/estilos.css”>

Con este método podemos manejar paginas distintas con el mismo estilo y cambiar 1 sola vez el estilo en el .css afecta a todas las paginas. Con lo cual es mas practico.

Ejemplo forma 1 inline



Ejemplo forma 2 en el head



Ejemplo forma 3 importado de un estilos.css externo



Herencias

Estilos respetan herencias salvo que se indique lo contrario. Los elementos adentro de otro elemento, heredan como hijos las propiedades definidas. A menos que se indique expresamente lo contrario.

Orden jerarquico según tipo de cambio, del menos importante al mas importante:

1.    Estilos del navegador

2.    Hoja de estilos del usuario (archivo vinculado) archivo estilos.css

3.    Etiqueta style <style> h2 { color:blue; } </style>

4.    Atributo style <h2 style="color:red">Subtitulo</h2>

5.    !Important color:grey !important;

Es decir, si para body definiste letra gris, todo adentro de body, y sus elementos también, llevan letra gris. Salvo que hayas definido algún elemento específicamente con otro color. Es decir, termina mandando la instrucción que diste a nivel mas especifico. Y si querés romper con esta lógica, le tenes que asignar !important a la que queres que mande por sobre las demás.

SELECTORES DE ETIQUETA

Estos selectores buscan un elemento DOM con el nombre de la etiqueta. Aplican a todas las etiquetas.

Por ejemplo en estilos.css aplicamos un selector de etiqueta body

Body

{

Background-color:black;

}

Otro ejemplo: aplicamos un selector de etiqueta img

Img

{

Border-radius:10px;

}

SELECTORES ID, CLASS

SELECTOR ID

Si tenemos todos los elementos de nuestro html identificados con un id de esta manera:

<article id=”estudios”>

Y en el estilos.css poner:

#estudios

{ font-size: 20px;}

SELECTOR CLASS

Puede localizar un numero de elementos en el dom y aplicarles un estilo.

<article class="estiloparavinetas">

Y en el estilos.css poner:

.estiloparavinetas

{ font-size: 20px;}

Tanto ID como CLASS, hacen lo mismo, peeeeero: con una clase podrás establecer un mismo estilo a **vaaarios** elementos con la misma clase, en cambio, con el id, sólo a **uno y sólo un** elemento! Esto es porque a cada elemento, le tenes que poner un id diferente. Ejemplo: a cada <li> le tenes que poner un id y no se pueden repetir, con lo cual después vas a tener que especificar el estilo para cada id y es mucho laburo. En cambio muchos elementos pueden tener el mismo class.

SELECTORES DE ATRIBUTOS

Es un selector mas inteligente que te permite buscar y asi seleccionar atributos. Por ejemplo, si queres hacer un estilo determinado con todas las etiquetas “a” que tengan un atributo href, entonces en tu estilos.css pones lo siguiente

a [href],

esto te va a buscar todas las etiquetas a con atributo href

Si querés ser mas específico y seleccionar solo aquellas etiquetas que tengan equis valor en un atributo también lo podes hacer, de esta manera:

a[href=http://www.google.com]

Y también podemos buscar valores al principio, al final, o en el medio del contenido.

Para buscar al final, metes el signo$

Para buscar al inicio, metes el signo ^

Para buscar en el medio con multiples valores separados por espacio ~

Para buscar a todos los que al inicio arrancan igual, usamos |

De esta manera quedan:

* img[src$=.svg]
* a[href^=www]
* img[src~=icono]

Y por ultimo, NO IMPORTANDO de que etiqueta se trate, podes buscar todas las etiquetas que tengan un atributo con valor determinado, ya sea etiquetas variadas o no. Lo haces asi, suponiendo que justo tenes varias etiquetas que tengan un atributo title con valor

* [title\*”ejemplito”]



SELECTORES DE ATRIBUTOS

SELECTOR UNIVERSAL

Con asterisco podemos aplicar un estilo a absolutamente todo junto. Por ejemplo ponele que querramos agregar margen a nuestra pagina.

\* { margin: 1vh 1vw;}

Eso aplica a todo el documento!

SELECTOR DE HERMANOS

Con + hacemos que se de una condición de precedencia. Por ejemplo en este caso:

h1+h2{color:black}

Ahí le estamos diciendo que ponga h2 en negro, SI Y SOLO SI, ese h2 está precedido inmediatamente por un h1. Esto lo que va a hacer es poner en negro esos h2, pero no los h2 que no cumplan con esa regla.

SELECTOR ADYACENTE

Con ~ hacemos que se de una condición de sucesión, por ejemplo en este caso

h2~p{color:black}

Ahí le estamos diciendo que ponga p en negro, SI Y SOLO SI, ese p viene después de un h2. Esto lo que va a hacer es poner en negro esos p, pero no los p que no cumplan con esa regla.

SELECTOR DE HIJOS DIRECTOS

Con > seleccionamos los hijos directos

Article>h2{color:black}

Ahí le estamos diciendo que ponga h2 en negro, si es hijo de article. Sino no.

SELECTOR DESCENDENTE

Otra es para las listas, todos los li que estén dentro de ul, por ejemplo, pero no los li que estén dentro de ol. Eso es asi:

ul li {list-style-type:circle}