**Diseño Visual Web**

Los métodos mas comunes para modificar la apariencia de un HTML son dos. El primero se lo conoce como *“inline style”*mientras que al como *“CSS”.* A continuación, se explicará cada uno.

**Inline Style**

Consiste en agregar atributos directamente en la etiqueta de un elemento. Ejm:



En este fragmento se le ha agregado el color azul al contenido de la etiqueta “*h1*”.

Es bastante útil cuando se desea trabajar con pocos elementos y se desea agregar

**CSS ( Cascade Style Sheet)**

Es una tecnología de diseño gráfico con estilo de cascada que utiliza lenguaje de marcado para darle propiedades gráficas a los elementos de una pagina web.

**Enlace**

Para utilizar un archivo CSS es necesario establecer un vinculo. Esto se hace agregando un tag de enlace o “<link>” en el “header” de nuestro archivo.



*“style.cc”* es el tipo de vinculo con nuestro html , mientras que “style.css” es el nombre del archivo dentro del directorio raíz.

**Ventajas**

1. Mejora la consistencia visual
2. Reduce el tamaño del archivo HTML mejorando los tiempos de ejecución.
3. Hace que los sitios web sean más fáciles de actualizar y modificar.

Ahora que esta hecho el enlace ya podremos empezar a editar. Para referenciar elementos html se utilizan los *“selectores”*.

**Selectores**

Son identificadores que permiten abrir bloques de código en los que agregaremos atributos. A continuación, se explicarán algunos de los mas comunes.

**Selector por etiqueta**

A los archivos css se les conoce como cascada style sheet por la forma en que aplican los estilos que se asemeja a una cascada. Eso quiere decir que el estilo asignado por un selector de mayor *“especificidad”* puede sobrescribir el estilo de otro y que un selector grupal puede aplicar estilo a varios elementos. Entonces teniendo eso en cuenta nos guiaremos en el siguiente esquema de especificidad.

HTML**:**



Aquí tenemos a un h2 con un poco de texto en su interior que genera el siguiente resultado en la pagina web.



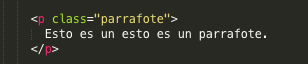
Ahora utilizando un selector de etiqueta, en nuestro archivo css abriremos un bloque de código delimitado por llaves y le asignaremos el valor *“green”* al atributo color de nuestro “*h2”*.

CSS:



**Seleccionar por nombre de clase.**

**HTML**:



**CSS**:



**NAVEGADOR**:

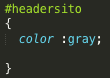


**Seleccionar por id.**

**HTML:**

****

**CSS:**

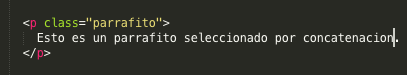
****

**NAVEGADOR:**

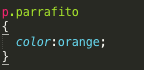
****

**Seleccionar por concatenación.**

**HTML:**

****

**CSS:**

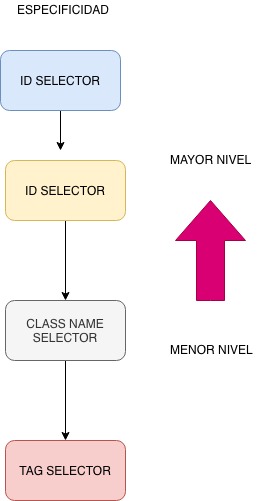
****

**NAVEGADOR:**

****

**Especificidad**

Es el nivel de prioridad con la que el navegador despliega los estilos. Eso significa que sobrescribirá el estilo contenido en selectores de menor especificidad. A continuación, un esquema con el nivel de especificidad de los selectores vistos anteriormente.



**Especificidad máxima**

Existe un elemento denominado ¡important flag

A diferencia de las reglas anteriores, este se aplica directamente a un atributo y hace que ningún otro selector pueda aplicarle estilo.

Es recomendable trabajar con la menor especificidad posible, así será fácil aplicar cambios posteriores de ser necesario.

**JavaScript**

**¿Qué es lo que JavaScript hace en tu página web?**

Cuando cargas una página Web en tu navegador, tu código (HTML, CSS y JavaScript) es leído dentro de un ambiente de ejecución (pestaña del navegador). Esto es como una fábrica que coge la materia prima (Las líneas de código) y lo presenta como el producto final (la página Web).

Orden de ejecución

Cuando el navegador encuentra un bloque de JavaScript, generalmente lo corre en orden, de arriba hacia abajo. Esto significa que tienes que tener cuidado en qué orden pones las cosas.

**El Interpretador entre el código compilado**

Quizás debiste escuchar sobre los términos interpretador y el compilador en contexto de programación, JavaScript es un lenguaje de interpretación – el código se hace funcionar de arriba hacia abajo y el resultado de leerlo de esta manera hace que inmediatamente responda. No tienes que transformar tu código en algo diferente antes de que el navegador lo haga por ti.

**La sintaxis de JavaScript** es muy similar a la de otros lenguajes de programación como Java y C.

Normas básicas que definen la sintaxis de JavaScript:

* No se tienen en cuenta los espacios en blanco y las nuevas líneas:
* Se distinguen las mayúsculas y minúsculas:
* No se define el tipo de las variables:

No es necesario indicar el tipo de dato que almacenará.

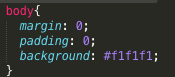
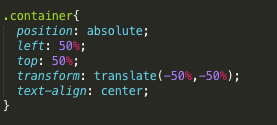
* No es necesario terminar cada sentencia con el punto y coma.
* Se pueden incluir comentarios:
* Los comentarios se utilizan para añadir información en el código fuente.

**Ejercicio**

Botones con efecto transición de gradiente que llaman algoritmos de JS.

**HTML**

**CSS**

1. Agregado de margen e interespaciado entre elementos utilizando tag selector “body”.  
     
   
2. Classname selector a “conteiner” en el que se modifica:
   1. Posición en relación con el marco del navegador.
   2. Alineación de texto  
        
      
3. Classname selector al botón 1 en el que se agrega la transición de colores.  
     
   

**JS**