**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.

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.

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.



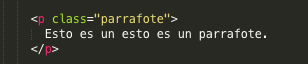
Ahora utilizando un selector de etiqueta, en nuestro archivo css abriremos un bloque de código delimitado por llaves y le asignaremos el valor *“green”* al atributo color de nuestro “*h2”*.

CSS:



**Seleccionar por nombre de clase.**

**HTML**:



**CSS**:



**NAVEGADOR**:

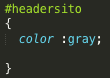


**Seleccionar por id.**

**HTML:**

****

**CSS:**

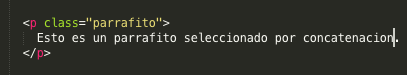
****

**NAVEGADOR:**

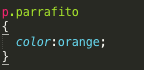
****

**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.

**Ventajas de usar una css style sheet:**

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.

**JavaScript**

JavaScript es un lenguaje de programación que te permite realizar actividades complejas en una página web, cada vez más una página web hace más cosas que sólo mostrar información estática, como mostrar actualizaciones de contenido en el momento, interactuar con mapas, animaciones gráficas 2D/3D etc. puedes estar seguro que JavaScript está involucrado. Es la tercera capa del conjunto de los estándares en las tecnologías para la web, dos de las cuales son (HTML y CSS), de los cuales ya hablamos anteriormente.

Es un lenguaje de programación que te permite crear contenido nuevo y dinámico, controlar archivos de multimedia, crear imágenes animadas y muchas otras cosas más. En definitiva, realizar la programación necesaria en el sitio web.

Hay algo incluso más emocionante, sin embargo, es la funcionalidad construida por encima del núcleo del lenguaje de JavaScript. Llamada Interfaz de programación en aplicaciones - Application Programming Interfaces (API's )

Los APIS son inserciones de líneas, incluso bloques gigantes de código listos para usar que permiten a un desarrollador implementarlo a programas que de cualquier otra forma podría ser difícil o incluso imposible de terminar. Así como las herramientas para construir una casa, es lo mismo para las cosas de programación – Es mucho más fácil tomar los paneles que ya estén cortados y atornillarlos para hacer un estante de libros, ya que es más trabajoso diseñarlo por ti mismo, ir y encontrar la madera correcta, cortarla del tamaño correcto y lijarla, buscar los tornillos del tamaño correcto y ensamblarla para hacer un estante de libros.

**¿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).

El lenguaje JavaScript es ejecutado por el motor del navegador de JavaScript, luego que el código HTML y CSS han sido juntados y congregados dentro de la página Web. Esto asegura que el estilo y la estructura de la página están en su lugar en el momento en que JavaScript comienza a ejecutarse.

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:

El intérprete de JavaScript ignora cualquier espacio en blanco sobrante, por lo que el código se puede ordenar de forma adecuada para entenderlo de una manera sencilla (tabulando las líneas, añadiendo espacios, creando nuevas líneas, etc.)

* Se distinguen las mayúsculas y minúsculas:

En JavaScript se intercambian mayúsculas y minúsculas el script no funciona.

* No se define el tipo de las variables:

En el intérprete de JavaScript al crear una variable, no es necesario indicar el tipo de dato que almacenará. Debido a que una misma variable puede almacenar diferentes tipos de datos durante la ejecución del script.

* No es necesario terminar cada sentencia con el carácter de punto y coma (;):

Aunque JavaScript no es obligación de hacerlo, es conveniente seguir la tradición de terminar cada sentencia con el carácter del punto y coma (;).

* Se pueden incluir comentarios:

Los comentarios se utilizan para añadir información en el código fuente del programa. Este contenido no es visualizado en pantalla, pero se envía al navegador del usuario.

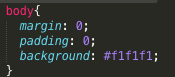
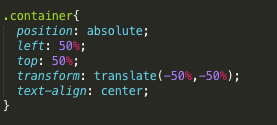
* + Comentario de una línea
  + Comentario de varias líneas

**Ejercicio**

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

HTML

CSS

1. Tag selector al cuerpo y agreado de atributos  
     
   
2. Classname selector al conteiner y atributos  
     
   
3. Classname selector al botón 1 y agregado de gradiente  
     
   

JS