

ENTORNOS GRÁFICOS

TRABAJO PRÁCTICO CSS

**Comisión**: 3 (4E03)

**Integrantes**:

|  |  |
| --- | --- |
| Legajo | Apellido y Nombre |
| 45882 | Romaniuk, Federico Nicolás |
| 45878 | Galdeano, Gonzalo |
| 46472 | Raselli, Gianfranco |

# Ejercitación 1.

1. **¿Qué es CSS y para qué se usa?**

CSS son las siglas de Cascade Style Sheet que traducido significa hojas de estilo en cascada. Las hojas de estilo es una tecnología que nos permite controlar la apariencia de una página web.   
En otras palabras, es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura.

Se utiliza para:

* Describir como los elementos dispuestos en la página son presentados al usuario.
* Especificar estilos como el tamaño, fuentes, color, espaciado entre textos y recuadros, así como el lugar donde disponer texto e imágenes en la página.
* Facilitar a desarrolladores el control total sobre el estilo y formato de los documentos.
* Para dar estilo a documentos HTML y XML, separando el contenido de la presentación.

1. **CSS utiliza reglas para las declaraciones de estilo, ¿cómo funcionan?**

Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML. La regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por una propiedad y el valor que se le asigne.

Ejemplo: h1 {color: red}

* H1: Selector
* Color: red🡪Declaración

1. **¿Cuáles son las tres formas de dar estilo a un documento?**

Las tres formas de darle formato a un documento html son:

1. Utilizando una hoja de estilo externa que estará vinculada mediante el elemento <link>, el cual debe estar situado en la sección <head>
2. Utilizando el elemento <style> en el interior del documento al que se le quiere dar estilo y que generalmente se situaría en la sección <head>. Los estilos serán reconocidos antes que la página se cargue por completo.
3. Utilizando estilos en aquellos elementos que lo permitan utilizando el atributo <style> dentro del <body>. Este tipo de estilo pierde las ventajas que ofrecen las hojas de estilo al mezclarse el contenido con la presentación.
4. **¿Cuáles son los distintos tipos de selectores más utilizados? Ejemplifique cada uno.**

Tipo de selectores más utilizados:

* 1. Selector universal:
     1. Sintaxis: \*{Propiedad: valor;}
     2. Ejemplo: \*{color: red} El formato se aplicará a todos los elementos de la página.
  2. Selector etiqueta:
     1. Sintaxis: etiqueta{Propiedad :valor}
     2. Ejemplo: p{color: rojo} el formato se aplicará a todos los elementos <p> del documento.
  3. Selector clase:
     1. Sintaxis: clase {Propiedad :valor}
     2. Ejemplo: .index {color: rojo} el formato se aplicará a cualquier elemento que tenga la clase .index
  4. Selector identificador: utiliza el atributo id para seleccionar un elemento. Solo puede haber un elemento identificado con un id en un documento.
     1. #id {Propiedad: valor}
     2. #cent {color: blue} El estilo se aplicará al elemento que tenga el id #cent.
  5. Selector descendiente: Un elemento es descendiente de otro cuando se encuentra entre las etiquetas de apertura y de cierre del elemento padre.
     1. Sintaxis: Selector1 selector 2… Selector N{Propiedad: valor}   
        Nota: El formato se aplica sobre el elemento N
  6. Combinación de selectores: Nos permite darles el mismo formato a todos los selectores:
     1. Sintaxis: Selector 1, selector 2, … {Propiedad: valor}
     2. Ejemplo: h1, h2, h3 {color: black}
  7. Selector de hijos: Se utiliza para indicar que un elemento es hijo de otro mediante el símbolo (>)
     1. Sintaxis: Selector 1> Selector 2 { propiedad: valor; }
     2. Ejemplo: H1>h2 { color: black; }
  8. Selector adyacente: Se utiliza para dar estilo a dos elementos que se encuentran bajo el mismo padre mediante el símbolo (+):
     1. Sintaxis: Selector 1 + Selector 2 {Propiedad: Valor}
     2. Ejemplo: h2 + h3 { Color: black; }

1. **¿Qué es una pseudo-clase? ¿Cuáles son las más utilizadas aplicadas a vínculos?**

Una pseudoclase es un selector que marca los elementos que están en un estado específico, por ejemplo, los que son el primer elemento de su tipo, o aquellos por los que el cursor les pasa por encima. Tienden a actuar como si hubieras aplicado una clase en una parte determinada del documento y, a menudo, ayudan a reducir el exceso de clases y proporcionan un marcado más flexible y fácil de mantener.

Permiten aplicar un estilo a un elemento no sólo en relación con el contenido del árbol de documento, sino también en relación a factores externos como el historial del navegador, el estado de su contenido o la posición del ratón.

Sintaxis:

Selector: pseudo-clase{ propiedad: valor; }

1. **¿Qué es la herencia?**

Una de las características principales de CSS es la herencia de los estilos definidos para los elementos. Cuando se establece el valor de una propiedad CSS en un elemento, sus elementos descendientes heredan de forma automática el valor de esa propiedad.

1. **¿En qué consiste el proceso denominado cascada?**

El proceso de cascada en CSS es el mecanismo que controla el resultado final cuando, por existir declaraciones múltiples, surgen conflictos sobre qué estilos se aplican a un elemento. Dicho de otra manera: si hay dos o más declaraciones que afectan a un elemento, la cascada determina cuál de ellas tiene preferencia, mayor peso, y se aplicará.

La forma de calcular esta preferencia no se realiza de forma arbitraria, sino que tiene su proceso y reglas en base a los tres conceptos principales que controlan la prioridad en que las declaraciones CSS se deben aplican:

1. La importancia de la regla.
2. El origen o procedencia.
3. El orden de aparición en el código.

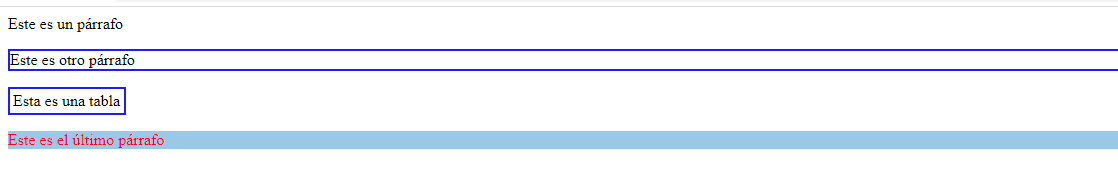
Cuando el navegador recibe una página actúa de la siguiente manera para calcular qué aplicar:

1. Selecciona los estilos a aplicar por el tipo de medio.
2. Aplica los que corresponde por su importancia y origen.
3. Si las declaraciones tienen la misma importancia y origen, prima la especificidad del selector.
4. Si se mantiene el empate por importancia, origen y especificidad manda el orden de aparición en el CSS. La última declarada gana.

# Ejercitación 2.

**Efectos:**

* El primer párrafo tiene un selector id llamado normal que muestra el texto con la fuente arial,helvetica; de tamaño 11 px y en negrita.
* El segundo párrafo tiene un selector id llamado destacado que define un borde alrededor de sus cuatro lados de color azul y tamaño 2 px.
* Tabla de una fila y una columna que también hace referencia al identificador llamado destacado que tiene contorno azul y de tamaño 2 px.
* El último párrafo que hace referencia al selector id llamado distinto que color de fondo celeste y color de texto rojo.

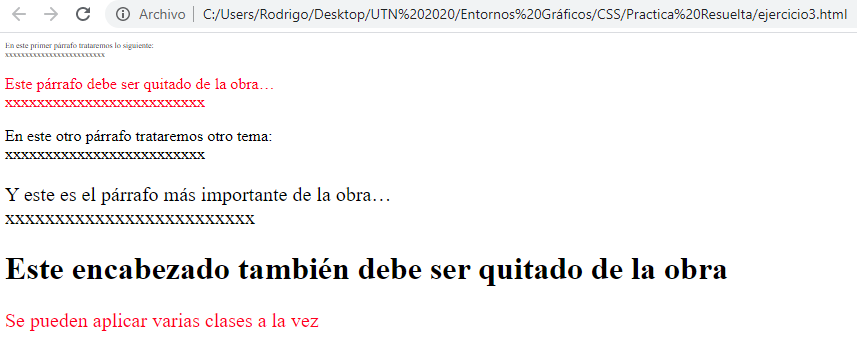
**Gráficamente:**   


# Ejercitación 3.

**Efectos:**

* Párrafo con clase llamado desarrollo que define el tamaño de su contenido de 8px.
* Párrafo con clase llamado quitar que define su contenido de color rojo.
* Párrafo que no tiene estilos declarados por lo que se muestra con tamaño y color de default.
* Párrafo con clase llamado importante que cambian el tamaño de su contenido a 20px.
* Título h1 con clase quitar que como solamente esta está definida para párrafos no va a cambiar a color rojo.
* Párrafo que aplica las clases quitar e importante que cambia su color a rojo y su tamaño a 20px.

**Gráficamente:**



# Ejercitación 4.

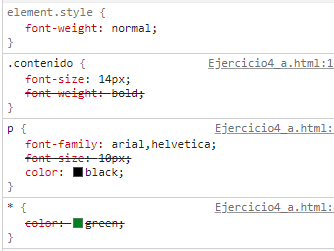
**Efecto en ejercicio 4 a:**

Se define una etiqueta p cuya clase es contenido. También se define una

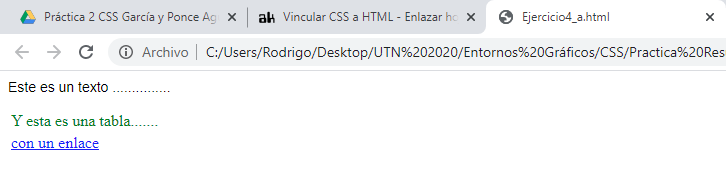
Propiedad estilo Font-weight: normal. Esto funciona de la siguiente manera:

* Primero p adquiere todas las propiedades declaradas para la clase contenido, es decir: .contenido{ font-size: 14px;font-weight: bold; }
* Luego p asume las propiedades declaradas para la etiqueta p, es decir:  
  p { font-family: arial,helvetica; font-size: 10px; color: black; }
* En la etiqueta dentro del body se define una propiedad de estilo donde la declaración es: { Font-weight: normal}

Entonces la etiqueta p tendrá un Font size: 14 (selector .contenido), Font-family: arial,helvética, color: black ( selector p), Font-weight: normal (elemento style).

Utilizando el navegador e investigando el elemento podremos visualizar esto:  


**Gráficamente:**



**Efecto en ejercicio 4 b:**

Se define el body con una clase “contenido”. Esto significa que en primera instancia todos los elementos del body se les aplicará estas propiedades.

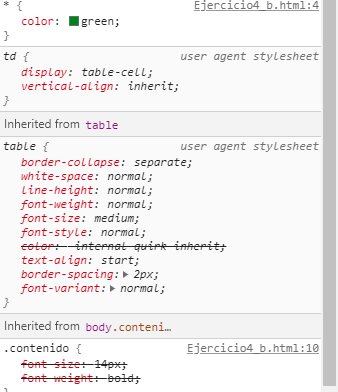
Se define un párrafo p. En las declaraciones tenemos un párrafo p cuyas propiedades se aplican:

* Primero se aplica las propiedades de las clase del body (p hereda de body): .contenido{ font-size: 14px; font-weight: bold; }
* Luego toma las propiedades declaradas para las etiquetas p { font-family: arial,helvetica; font-size: 10px; color: black; }

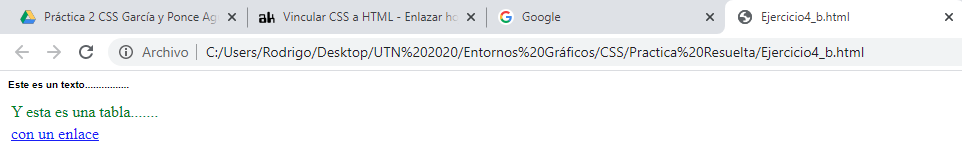
Por eso se puede visualizar en la imagen que el texto asume la fuente, el tamaño y el color definido en la etiqueta p, y también hereda las propiedades de la clase.   
Esto lo podemos ver analizando el elemento desde el navegador:  


Luego el comportamiento de la tabla definida:

* Asume el color definido en el selector universal.
* Asume las características por defecto de la tabla.



**Gráficamente:**



La diferencia en este estos códigos es donde se define la clase, en uno se define dentro de un párrafo y en el otro se define a nivel del body. En base a esto, comienza la aplicación en cascada de las propiedades. Por eso en el primer caso el texto que tiene la clase contenido mantiene el tamaño 14 px definido en la clase. En cambio en el otro caso, al estar definido a nivel de body, la propiedad definida en el selector .contenido es reemplazada por la propiedad definida en selector p.

# Ejercitación 5.

1. Los textos enfatizados dentro de cualquier título deben ser rojos.  
   h1,h2,h3,h4,h5,h6: {color:red;]
2. Cualquier elemento que tenga asignado el atributo "href" y que esté dentro de cualquier párrafo que a su vez esté dentro de un bloque debe ser color negro.  
   P a [href]:{ color: black;}
3. El texto de las listas no ordenadas que estén dentro del bloque identificado como “ultimo” debe ser amarillo pero si es un enlace a otra página debe ser azul.  
   ul li #ultimo {color: yellow;}  
   ul li #ultimo a {color: blue;}
4. Los elementos identificados como “importante” dentro de cualquier bloque deben ser verdes, pero si están dentro de un título deben ser rojos.  
   #importante {color: green;}

h1,h2,h3,h4,h5,h6 #importante {color: red;}

1. Todos los elementos h1 que especifique el atributo title, cualquiera que sea su valor, deben ser azules  
   h1[title] {color: blue;}
2. El color de los enlaces en las listas ordenadas debe ser azul para los enlaces aún no visitados, y violeta para los ya visitados y, además, no deben aparecer subrayados.  
   ol li a{color: blue;}

ol li a:visited{color: violet, text-decoration: none;}

# Ejercitación 7.

Juego completado en https://flukeout.github.io/.