

CSS: CASCADING STYLE SHEETS / Hojas de estilo en cascada

CSS, Hojas de estilo en cascada, es el estándar de la W3C para definir la presentación de un documento HTML. Por la Presentación nos referimos a la manera en que el documento es visualizado por el usuario en un pc, celular, tv, etc. [CSS es un lenguaje separado con su propia sintaxis.](#)

Como funciona CSS?

1. Empezamos con el documento HTML.
2. Escribimos reglas de estilo para como nos gustaría que luciera el HTML.
3. Adjuntamos las reglas de estilo al documento.
4. Cuando el navegador muestra la página, sigue las reglas de estilo para dibujar los distintos elementos como parrafos, listas, headings, etc.

Como se escriben las reglas de estilo?

- Una hoja de estilo en cascada esta formada por una o mas REGLAS que describen como un elemento o grupo de elementos debería ser mostrado.
- Cada regla SELECCIONA un elemento y DECLARA como debería lucir.

Veamos un ejemplo:

```
h1{
  color: green;
}
p{
  font-size: 15px;
  font-family: sans-serif;
}
```

Este ejemplo contiene dos reglas, la primera hace que todos los **h1** en el documento sean verdes, la segunda especifica el tamaño y la fuente de los parrafos, esto produce algo como:

La Selección ya se instaló en Singapur sin Messi, Higuaín y Otamendi

Los tres futbolistas no jugarán el amistoso del martes ante el seleccionado local. Debido a la exigencia en sus clubes el mister decidió otorgarles algo de descanso. Por otra parte el *resto* del plantel sigue enfocado en el amistoso de la proxima semana.

El arribo



A menos de 24 horas de la victoria 1 a 0 frente a Brasil en Melbourne, que marcó el punto de partida del ciclo de Jorge Sampaoli como entrenador, el seleccionado argentino para la segunda y última estación en esta gira: el duelo que el martes disputará ante el combinado local.

El plantel argentino arribó a Singapur en un vuelo chárter que lo trasladó desde Melbourne. A ese avión no se subieron Nicolás Otamendi y Gonzalo Higuaín, quienes participarán del duelo del martes (a las 9, hora argentina) en el estadio Nacional de Singapur.

Hay fiesta señores

Las partes principales de una REGLA son:

- **SELECTOR**
- **DECLARACIÓN (DECLARATION):**
 - Propiedad.
 - Valor.

El selector identifica el elemento o los elementos que van a ser afectados. La declaración esta formada por una propiedad (en el ejemplo anterior **color**) y un valor (**green**) los cuales se separan por dos puntos (:). Cada declaración finaliza en punto y coma. Además un selector puede contener varias declaraciones.

declaration

```
selector { property: value; }
```

declaration block

```
selector {  
  property1: value1;  
  property2: value2;  
  property3: value3;  
}
```

Selectores

En el ejemplo anterior (imagen al costado) **h1** y **p** son usados como selectores, este es el tipo mas básico de selector llamado **selector de tipo de elemento**, como lo indica el nombre, las propiedades definidas para una regla con este tipo de selector aplican a todos los elementos del mismo tipo en el documento, en este caso todos los h1 van a ser verdes, todos los parrafos van a tener fuente sans-serif y 15px.

```
h1{  
  color: green;  
}  
p{  
  font-size: 15px;  
  font-family: sans-serif;  
}
```

Cuando se proveen medidas a los valores, por ejemplo 15px siempre va el numero pegado a la unidad, no puede haber espacio en blanco.
En caso de que el valor sea cero, no es necesario escribir la unidad.

Declaraciones

Cada declaración esta formada por una propiedad y valor. Cada declaración debe finalizar con punto y coma para mantener separada una declaración de la otra. Si se omite el punto y coma la declaración y la siguiente van a ser ignoradas.

declaration

```
selector { property: value; }
```

El poder de CSS esta en la variedad de propiedades y valores que ofrece, a su vez, los valores dependen de la propiedad. Algunas propiedades toman colores y otras tamaños.

Ejercicio 7-1:

Dentro de la etiqueta `style` agregar las siguientes reglas de css:

1. Hacer que los `h1` tengan color verde (green).
2. Hacer que los párrafos tengan font-size de 15 px y font-family de tipo sans-serif.
3. Hacer que los `h2` tengan color azul (blue).
4. Hacer que los párrafos tengan un margen izquierdo de 100px (margin-left).
5. Hacer que los `h2` también tengan 100px de margen izquierdo.
6. Hacer que la `ul` tenga 130px de margen izquierdo.
7. Agregar un borde bajo azul de 1px al `h1` usando esta declaración:
`border-bottom: 1px solid blue;`
8. Mover la imagen al margen derecho y hacer que el texto la envuelva, a su vez agregar un margen de 0 px arriba y abajo, y un margen de 12 px a la derecha e izquierda. Para ello usamos:

```
img{
    float:right;
    margin-top:0;
    margin-bottom:0;
    margin-left:12px;
    margin-right:12px;
}
```
9. Ver los resultados en el navegador.

Adjuntando los estilos al documento

En el ejercicio anterior y otros que vimos anteriormente, embebimos la hoja de estilo dentro del documento html usando el tag **style**. En total existen 3 formas para añadir estilos a un documento html:

1. **Hojas de estilos externas:** Una hoja de estilo externa es un archivo separado que contiene reglas de estilo, este archivo debe tener el tipo de archivo **.css**. Este archivo css, puede ser linkeado por uno o mas archivos HTML, esto permite que varios archivos de una misma página web compartan la misma hoja de estilo y así el estilo en si. Esta es la forma ideal de añadir hojas de estilo.
Para esto usamos la siguiente etiqueta:

```
<link rel="stylesheet" href="css/style.css">
```

donde el atributo `rel` indica que el recurso relacionado es una hoja de estilo y el atributo `href` indica el archivo de la hoja de estilo que queremos adjuntar.

2. **Hojas de estilos embebidas:** Este es el método que venimos usando en los ejercicios, las reglas de estilo se sitúan en el elemento html **style** y las reglas de estilo dentro de esa etiqueta **solo aplican al documento html que las contiene**. La etiqueta **style** se situá dentro de la etiqueta **head**.

```
<head>
  <meta charset="utf-8">
  <title>Titulo</title>
  <style type="text/css" media="screen">
    /*INSERTAR LAS REGLAS CSS AQUI*/
  </style>
</head>
```

El atributo type ya no es requerido desde html5, la etiqueta style también puede tener o no el atributo media, el cual indica el medio específico tal como celular, pc, tv etc.

3. **Estilos en líneas:** permite aplicar reglas de estilo a un solo elemento html usando el atributo **style**, Los estilos en líneas solo aplican al elemento en el cual se define el atributo:

```
<h1 style="color:green;">Hola a todos</h1>
```

para agregar varias propiedades solo se las separa por punto y coma (;)

```
<h1 style="color:green; margin-top:20px;">Hola a todos</h1>
```

ESTA FORMA DE ESPECIFICAR ESTILOS DEBE SER OMITIDA, ES PROBLEMÁTICA, AL MOMENTO DE HACER CAMBIOS CON RESPECTO A LAS HOJAS DE ESTILOS.

Ejercicio 7-2:

1. Agregar en un h2 un estilo en línea para cambiar el color a orange, actualizar el navegador, ver los cambios y luego borrar este estilo en línea.
2. Reemplazar la hoja de estilo embebida por una hoja de estilo externa:
 1. crear una carpeta llamada CSS.
 2. Dentro de esta carpeta agregar un archivo llamado style.css
 3. Dentro del archivo style.css copiar las reglas de estilo definidas en la etiqueta style en el archivo html.
 4. Borrar la etiqueta style del archivo html.
 5. Agregar una etiqueta link que referencia al archivo CSS recién creado.

Algunas cosas importantes a tener en cuenta

Herencia

Al igual que los padres de una persona le pasan a sus hijos rasgos físicos como el color de pelo, ojos, entre otros, los elementos HTML que tienen estilos aplicados le pasan a los elementos html que contienen ciertas propiedades de estilo, por ejemplo nuestra pagina actualmente luce así:

La Selección ya se instaló en Singapur sin Messi, Higuaín y Otamendi

Los tres futbolistas no jugarán el amistoso del martes ante el seleccionado local. Debido a la exigencia en sus clubes el mister decidió otorgarles algo de descanso.

Por otra parte el resto del plantel sigue enfocado en el amistoso de la proxima semana.

El arribo

A menos de 24 horas de la victoria 1 a 0 frente a Brasil en Melbourne, que marcó el punto de partida del ciclo de Jorge Sampaoli como entrenador, el seleccionado argentino ya se instaló en Singapur para la segunda y última estación en esta gira: el duelo que el martes disputará ante el combinado local.

El plantel argentino arribó a Singapur en un vuelo chárter que lo trasladó desde Melbourne. A ese avión no se subieron:

- Nicolás Otamendi
- Gonzalo Higuaín
- Lionel Messi

Estos jugadores fueron desafectados y no participarán del duelo del martes (a las 9, hora argentina) en el estadio Nacional de Singapur.

Hay fiesta señores

Por su parte, Lionel Messi viajó junto a sus compañeros, pero al llegar al Aeropuerto Internacional Changi los caminos se separaron, ya que el rosarino, quien tampoco disputará el segundo amistoso de la gira, regresará a Barcelona para ultimar los detalles y firmar un nuevo contrato con el club catalán (sería hasta el 2022 y con una cláusula de rescisión de 400 millones de euros) y luego se trasladará a Rosario para casarse el 30 de junio con Antonella Roccuzzo.

Según el diario español Sport y Mundo Deportivo de España, la Pulga tiene previsto invitar a prácticamente todos los hombres del elenco catalán y ya habría expresado en su círculo íntimo lo que desea como regalo de bodas.



Solamente los parrafos tienen aplicado una fuente específica de letra, si ahora en lugar de aplicar el tipo de fuente a los parrafos se lo aplicamos al **body**, todos los elementos contenidos dentro de la etiqueta body van a heredar el valor de esta propiedad.

```
body{
  font-family: sans-serif;
}

p {
  font-size: 15px;
  margin-left: 100px;
}
```

Ejercicio 7-3: aplicar la regla de estilo font-family al body con el valor sans-serif y borrar la que posee párrafo.

Si guardamos los cambios y actualizamos la ventana del navegador ahora la página se vería algo como:

La Selección ya se instaló en Singapur sin Messi, Higuaín y Otamendi

Los tres futbolistas no jugarán el amistoso del martes ante el seleccionado local. Debido a la exigencia en sus clubes el mister decidió otorgarles algo de descanso.

Por otra parte el resto del plantel sigue enfocado en el amistoso de la próxima semana.

El arribo

A menos de 24 horas de la victoria 1 a 0 frente a Brasil en Melbourne, que marcó el punto de partida del ciclo de Jorge Sampaoli como entrenador, el seleccionado argentino ya se instaló en Singapur para la segunda y última estación en esta gira: el duelo que el martes disputará ante el combinado local.

El plantel argentino arribó a Singapur en un vuelo chárter que lo trasladó desde Melbourne. A ese avión no se subieron:

- Nicolás Otamendi
- Gonzalo Higuaín
- Lionel Messi

Estos jugadores fueron desafectados y no participarán del duelo del martes (a las 9, hora argentina) en el estadio Nacional de Singapur.

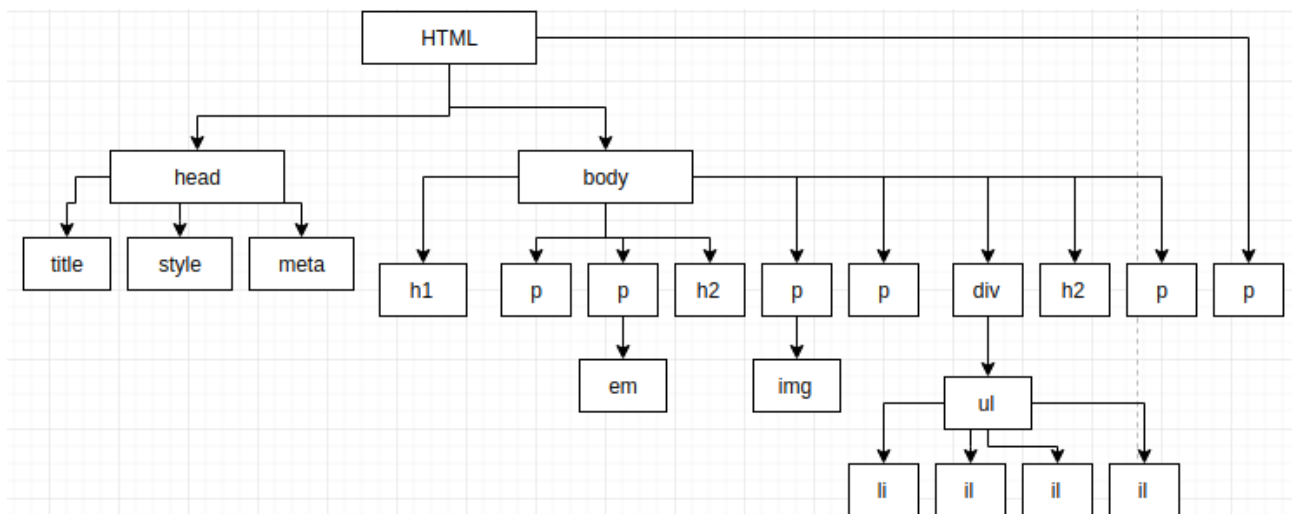
Hay fiesta señores



Acá podemos observar que no fue necesario escribir reglas específicas para que cada elemento tenga la propiedad `font-family` con el valor `sans-serif`, solo la especificamos una vez y gracias a la herencia este valor de la propiedad se hereda.

Estructura del documento

Para entender mejor la herencia, todo documento HTML tiene una estructura o jerarquía de elementos. Por ejemplo, la página de nuestro ejemplo tiene un elemento `html` que contiene un `head` y un `body`, y estos dos contienen a su vez otros elementos, los cuales a su vez algunos contiene otros elementos, esto lo podemos ver con la siguiente imagen:



Esta estructura se denomina **árbol**, todos los elementos contenidos en un elemento se denominan **descendientes**. Por ejemplo, los `p`, `h1`, `h2`, `div`, `ul`, `em`, `img`, `li`, son todos **descendientes** del `body`.

Todos los elementos que están contenidos directamente dentro de un elemento se denominan **hijos (child)** de ese elemento, a su vez, un elemento que contiene a otros elementos es el **padre (parent)** de estos elementos, por ejemplo, el `ul` es hijo del `div` y el `div` es el padre del `ul`.

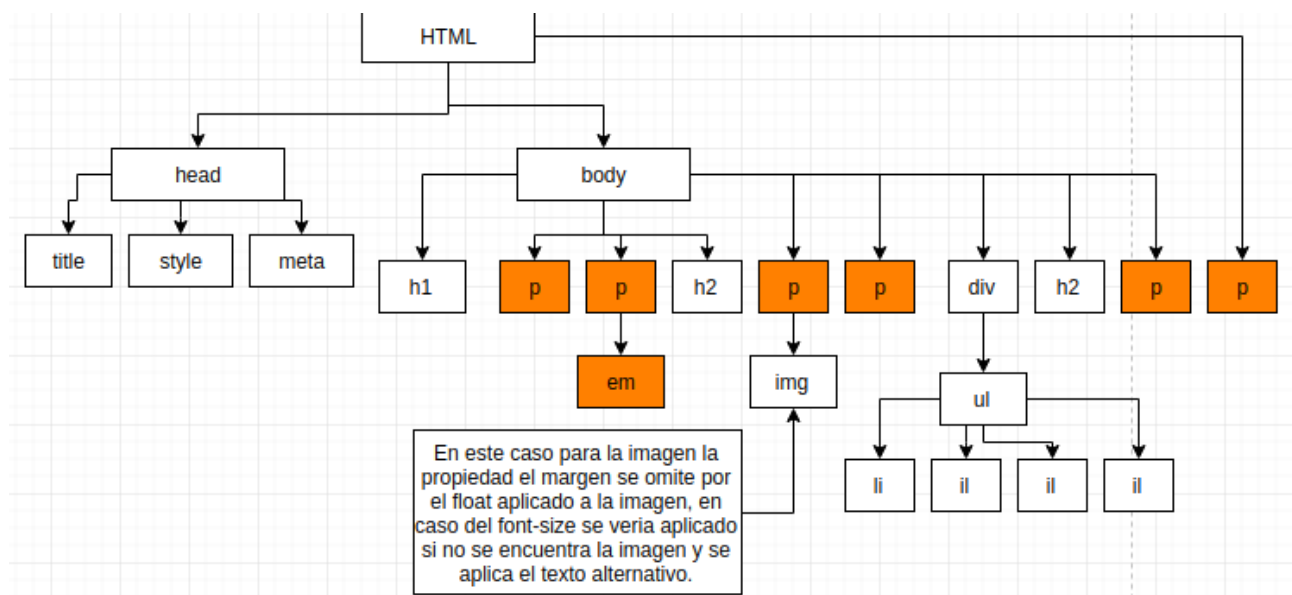
Todos los elementos que están mas arriba que un elemento en la jerarquía son sus **ancestros (ancestros)**. Dos o mas elementos con el mismo padre son **hermanos (siblings)**.

Posteriormente veremos que es posible crear selectores teniendo en cuenta esta estructura.

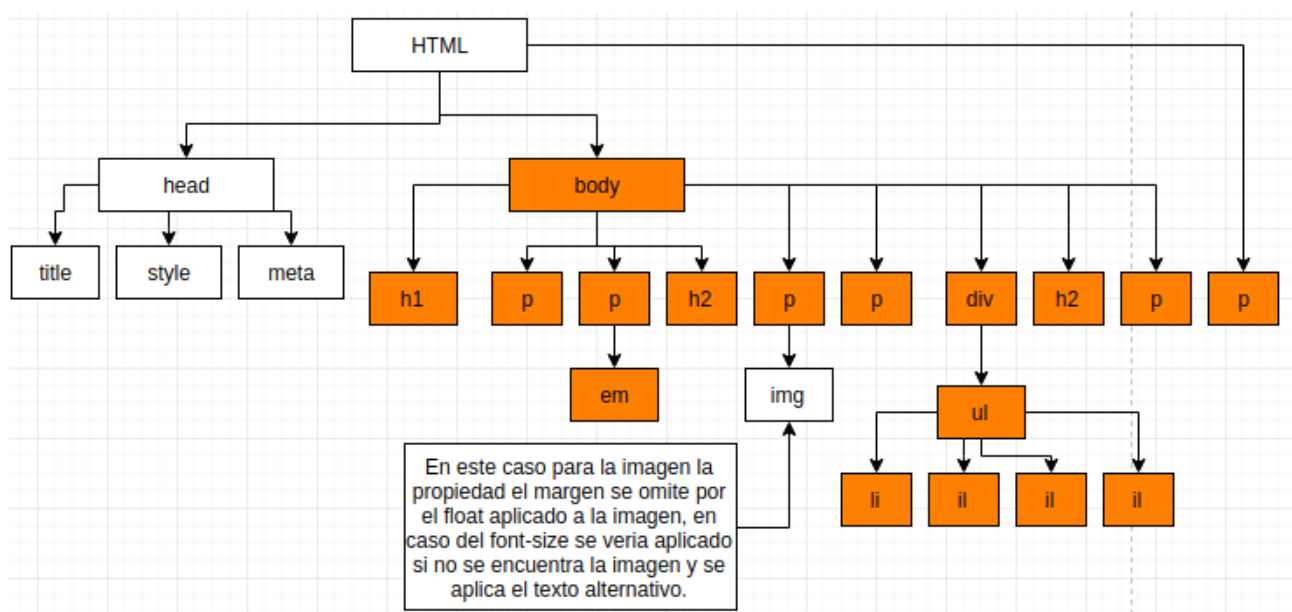
Como se transmiten los valores de propiedades a través de la herencia?

```
p {  
  font-size: 15px;  
  margin-left: 100px;  
}
```

Esta regla aplica para todos los parrafos del documento y para todos los elementos en linea contenidos dentro de los parrafos.



```
body{  
  font-family: sans-serif;  
}
```



Un aspecto a considerar en la herencia de las propiedades es que **ciertas propiedades son heredadas**, es decir, **algunas son heredadas y otras no**. Por lo general las propiedades referidas a estilo de texto como color, fuente, tamaño entre otras son heredadas. **Propiedades como bordes, márgenes y fondos son aplicados a lo que se llama el modelo en caja del elemento (box model) y no son heredadas.**

Esto tiene sentido por ejemplo, si a un párrafo le pongo un borde no me gustaría que cada línea de texto dentro del párrafo tenga un borde. **Ejercicio 7-4 (Probar `p{border: 2px solid green;}` y luego borrar)**

La herencia es una ventaja importante, en el caso de la regla aplicada al body:

```
body{
  font-family: sans-serif;
}
```

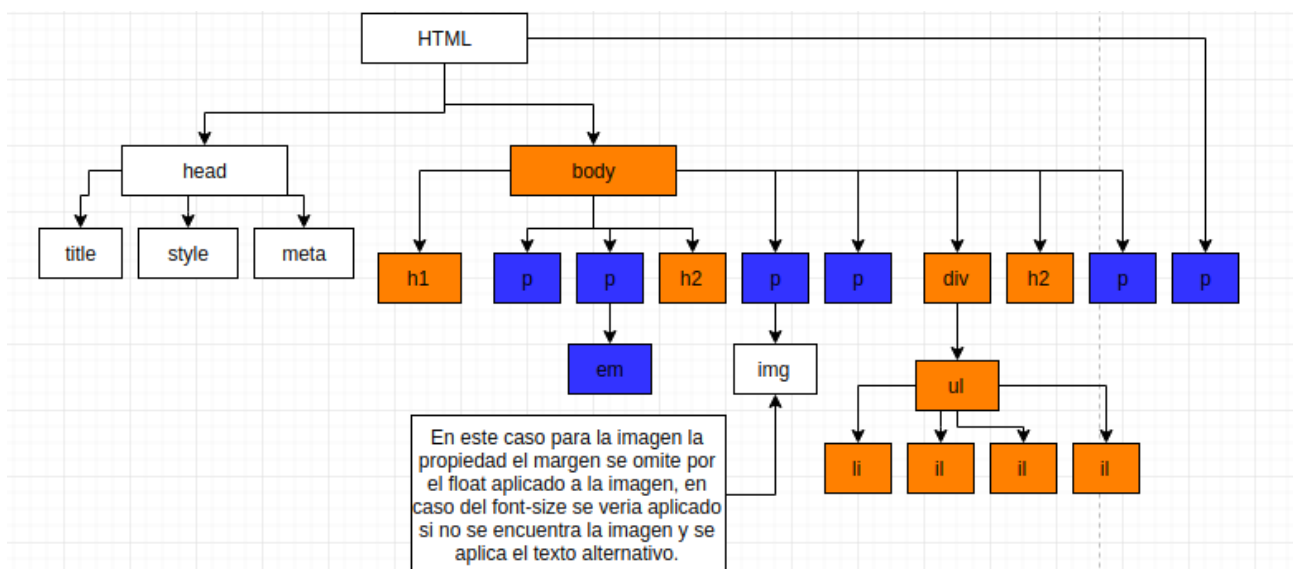
nos ahorramos de tener que escribir mas selectores para indicar que todos los elementos dentro de la página tengan que tener la fuente sans-serif.

Para sobrescribir valores heredados, cualquier propiedad aplicada a un elemento específico va a sobrescribir el valor de la propiedad heredado con el nuevo valor.

Ejercicio 7-5 supongamos que queremos que los párrafos tengan la fuente **Georgia**, agregamos al selector de párrafo una regla de estilo mas:

```
p {
  font-size: 15px;
  margin-left: 100px;
  font-family: Georgia;
}
```

En este caso todos los elementos del body van a tener la fuente sans-serif pero los párrafos sobrescriben esta propiedad y usan la fuente Georgia.



De donde viene el hoja de estilo en cascada – conflictos con estilos

CSS nos permite aplicar varias hojas de estilo a un mismo documento, por lo que es probable que se encuentren conflictos entre las reglas. Por ejemplo que debería mostrar el navegador si importamos dos hojas de estilo, la primera dice que los h1 deben ser rojos y la segunda dice que los h1 deben ser azules.

A su vez es probable que en una misma hoja de estilo varias reglas quieran modificar la misma propiedad para un elemento, como se hace para determinar cual se termina aplicando.

Cuando se ideó CSS se anticipó que este problema podría ocurrir y se implementó un mecanismo de jerarquías que asigna diferentes pesos a las reglas de estilo para determinar cual resulta aplicada.

El término **cascada**, se refiere a lo que pasa cuando varias fuentes de información de estilo compiten por el control de los elementos en una página: **la información de estilo se transmite ("cascadas" hacia abajo) hasta que se anula por una regla de estilo con más peso.**

Por ejemplo:

- Si no aplicamos ningún estilo a una página, va a ser mostrada de acuerdo a la hoja de estilo propia del navegador.
- Si adjuntamos una hoja de estilo a la página entonces va a sobrescribir las reglas de la hoja de estilo del navegador.

Existe una **jerarquía para determinar que regla de estilo termina siendo aplicada**, generalmente mientras mas cerca esta la hoja de estilo del contenido, mas peso se le da a las reglas de estilo. **Por ejemplo las hojas de estilo embebidas en la etiqueta style tienen mas pesos que las hojas de estilo externas ubicadas en archivos .css .**

Los estilos en linea son los que mas peso tienen ya que son los mas cercanos posible a un elemento, pero no es buena practica usarlos.

Para prevenir que una regla sea sobrescrita o no se aplique se puede usar el operador ! important en una regla.

En este caso los parrafos van a resultar en color negro y no azul.

Ejercicio 7-6 escribir esta nueva regla de estilo, guardar el archivo y actualizar la página para ver cambios.

```
p {
  font-size: 15px;
  margin-left: 100px;
  font-family: Georgia;
  color:black !important;
}
p{
  color:blue;
}
```

Jerarquía de las hojas de estilo

Como dijimos antes la información de estilo puede venir de varias fuentes, y hay reglas que van a sobrescribir a otras, **enumeradas de general a específico, los elementos de mayor numero de la lista anulan a los elementos que se encuentren por encima de ellos:**

1. Hoja de estilo propia del navegador.
2. Una hoja de estilo externa (linkeada o agregada con el tag **link**).
3. Hojas de estilo importadas (agregadas con **@import** dentro de una hoja de estilo)
4. Hojas de estilo embebidas (agregadas con el elemento **style**)
5. Estilos en linea (agregados con el atributo **style** en el tag de apertura de un elemento).
6. Reglas de estilo marcadas con **!important**.

Detalle de los selectores

Cuando dos o mas reglas en una hoja de estilo entran en conflicto el tipo de selector es usado para determinar la ganadora. Mientras mas específico o detallado es el selector, mas peso se le da al mismo y por ende a las reglas que quiere aplicar.

Por ejemplo

```
p {  
  font-size: 15px;  
  margin-left: 100px;  
  font-family: Georgia;  
}  
div>p{  
  font-family: sans-serif;  
}
```

En este caso el segundo selector es mas específico, selecciona a todos los párrafos que sean hijos de un div y les aplica el tipo de fuente sans-serif. Mientras que el primer selector es mas genérico selecciona a todos los párrafos. Por ende **el segundo selector es el ganador**.

Ejercicio 7-7: Aplicar el selector y regla de estilo de la imagen anterior y verificar resultados.

Orden de las reglas

El ultimo factor a considerar para resolver conflictos entre las reglas de estilo que tengan igual peso asignado es el orden de las reglas, en este caso si dos o mas reglas tienen el mismo peso de importancia, la regla aplicada es la ultima en aparecer, por ejemplo:

En este caso los párrafos van a tener color verde

Ejercicio 7-8: Aplicar las reglas de estilo de la imagen y verificar. (Comentar o borrar la regla donde se uso **!important** para el color del párrafo)

```
p{  
  color: red;  
  color: blue;  
  color: green;  
}
```

Selectores en grupo

Como vimos antes existen varios tipos de selectores, vamos a agregar un selector mas, hasta ahora vimos el selector de elemento específico, que nos permitía seleccionar un tipo de elemento html, **si queremos seleccionar varios elementos html específicos** podemos usar el **SELECTOR DE GRUPO**.

El selector de grupo permite aplicar reglas de estilo como si fueran varios selectores de elemento, solo que se los separa por coma:

Por ejemplo:

El primer selector de grupo aplica el color rojo a los h1, h2, h3, p y ul.

El segundo selector aplica el color azul a los h4, h5, h6 y span.

Ejercicio 7-9: Aplicar los selectores y verificar.

```
h1, h2, h3, p, ul{  
  color:red;  
}  
h4,h5,h6,span{  
  color:blue;  
}
```

Formateo de texto – Otros selectores

En esta parte vamos a ver las propiedades CSS que se pueden usar para cambiar la apariencia del texto y le vamos a sumar nuevos selectores para aplicar reglas de estilo.

Propiedades de fuente (font)

Las propiedades mas importante dentro del formateo de texto son las propiedades referidas a la fuente (font). Estas se refieren a aspectos como el tipo de fuente, el tamaño de la letra, negrita, cursiva, etc.

Las propiedades en cuestión son:

- font-family
- font-size
- font-weight
- font-style
- font-variant
- font

Propiedad font-family

Esta propiedad permite elegir el tipo de fuente para el texto, se usa para especificar una fuente o lista de fuentes por su nombre.

Valores posibles	<ul style="list-style-type: none">• Uno o mas fuente o familia de fuentes genéricas separadas por coma• inherit (indica que se hereda)
Valor por defecto	Depende del navegador
Aplica para	Todos los elementos html
Se hereda?	SI

Teniendo en cuenta este ejemplo:

```
body{
  font-family: Arial;
}
h1{
  font-family: Courier, monospace;
}
p{
  font-family: "Duru Sans", Verdana,sans-serif;
}
</style>
```

- Todos los nombres de fuentes, excepto los nombres de familias de fuentes genéricas deben estar capitalizadas (primera letra en mayúscula).
- Se usan comas para separar los distintos valores de la propiedad font-family.
- Los nombres de fuentes que tienen espacios en blanco deben ir entre comillas dobles.

Aspectos a tener en cuenta al usar fuentes

Limitaciones de las fuentes

En el ejemplo anterior vimos que en algunos selectores especificamos más de una fuente, esto se hace más que nada porque **los navegadores solo pueden mostrar las fuentes a las que tienen acceso**.

Esto antes significaba que el tipo de fuente tenía que estar instalada en la máquina del usuario del navegador. Pero con el nuevo estándar de CSS3 se introdujo la regla **@font-face** la cual permite proveer nuestras propias fuentes.

Que los navegadores puedan mostrar las fuentes a las que tengan acceso **significa** que si especificamos que la fuente debe ser Arial y el navegador no puede encontrarla (no está instalada en la máquina del navegador o no es provista) **el navegador usa la fuente por defecto**.

Por este motivo es que CSS permite especificar una lista de fuentes, si la primera opción no está disponible, se prueba con la segunda, si esta no está disponible con la tercera y así hasta que encuentra una disponible.

```
p{
  font-family: "Duru Sans", Verdana, sans-serif;
}
```

En este ejemplo va a probar con Duru Sans, supongamos que el navegador no la puede encontrar, entonces se prueba con Verdana, el navegador ve que esta disponible entonces se usa Verdana.

Anexo

Al igual que con las demás cosas, si queremos especificar nuestra propia fuente, se puede hacer en archivos con varios formatos, woff, ttf, svg, entre otros, esto se debe a que los navegadores no interpretan los mismo tipos de archivos de fuente.

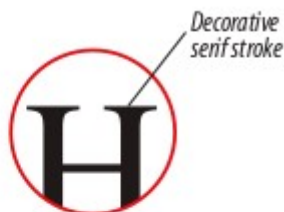
```
@font-face {
  font-family: 'Font_name';
  src: url('myfont-webfont.eot?#iefix') format('embedded-opentype'),
  url('myfont-webfont.woff') format('woff'),
  url('myfont-webfont.ttf') format('truetype'),
  url('myfont-webfont.svg#svgFontName') format('svg');
}
```

Familias de fuentes Genéricas

Como dijimos antes las familias de fuentes no necesitan ir con la primera letra mayúscula, en total hay 5 familias de fuentes, cuando usamos alguna, lo que hace el navegador es elegir una fuente disponible que pertenezca a esa familia. Las familias son:

- **serif**
 - incluye fuentes como Times, Times New Roman, Georgia
- **sans-serif**
 - Arial, Arial Black, Verdana, Helvetica, entre otras.
- **monospace**
 - Courier, Courier New, Andale Mono, entre otras.
- **cursive**
 - Apple Chancery, Zap-Chancery, Comic Sans, entre otras.
- **fantasy**
 - Impact, Western, entre otras.

Serif



Hello
Times

Hello
Georgia

Hello
Times New Roman

Hello
Lucida (Mac)

Sans-serif



Hello
Verdana

Hello
Trebuchet MS

Hello
Arial

Hello
Arial Black

Monospace

W i
Monospace font
(equal widths)

W i
Proportional font
(different widths)

Hello
Courier

Hello
Courier New

Hello
Andale Mono

Cursive

Hello
Apple Chancery

Hello
Comic Sans

Hello
Snell

Fantasy

Hello
Impact

HELLO
Stencil

HELLO
Mojo

Recomendaciones para usar fuentes:

- Empezar con la que mas te guste, especificar algunas alternativas que se parezcan y por lo general finalizar con una familia genérica.
- No hay limitaciones para el numero de fuentes, pero es recomendable que sean menos de 10.

Ejercicio 8-1

- Crear un archivo style.css y agregar el link en el archivo menu.html
- Al body darle la fuente Verdana y la familia sans-serif.
- Agregar en el head el siguiente link, el cual importa una fuente llamada Marko One:
`<link href="http://fonts.googleapis.com/css?family=Marko+One" rel="stylesheet">`
- A los h1 darle las fuentes Marko One, Georgia y la familia de fuentes serif. En este caso se debe escribir "Marko One" ya que tiene espacio.

Propiedad font-size

Esta propiedad nos permite especificar el tamaño del texto

Valores	<ul style="list-style-type: none">• Una unidad de medida.• Porcentaje• xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger• inherit
Valor por defecto	medium
Aplica para	Todos los elementos html
Se hereda?	Si

De acuerdo al cuadro de valores se puede especificar el font-size de varias formas:

- Usando una unidad especifica de medida , por ejemplo en este caso se usa la unidad de medida pixeles.

```
p{
  font-size:15px;
}
```

- Como un porcentaje, el cual aumento o disminuye el tamaño heredado, por ejemplo si el body tiene 10 px de font-size y quiero que los parrafos tengan 15px, con porcentaje queda como:

```
body{
  font-size: 10px;
}
p{
  font-size:150%;
}
```

- Usando alguna de las palabras claves como **xx-small**, **x-small**, **small**, **medium**, **large**, **x-large**, **xx-large**. El valor por defecto de los navegadores es **medium**.

```
body{
  font-size: medium;
}
```

- Usar las palabras **smaller**, **larger** para agrandar o achicar el texto según el texto que lo rodea.

Unidades de medida

Como vimos en la propiedad anterior es posible usar unidades de medidas existen muchas en CSS entre estas son **px**, cm, mm, cm, in , **em**, **rem**, vm, entre otras.

Los valores mas usados en la actualidad son los siguientes:

em: es una unidad de medida, en la cual 1em es igual al font-size heredado del padre

px (pixeles): es igual 1 pulgada/96.

rem (root em): es igual al tamaño en em pero se basa en el tamaño del elemento root `<html></html>` y no en el tamaño que hereda del padre.

Porcentaje: tamaño igual al porcentaje especifico aplicado al tamaño heredado.

Tanto em, porcentaje son medidas relativas, es decir se basan en otro font-size, el cual es el font-size del elemento padre que contiene al elemento sobre el cual se aplica la regla de estilo.

Rem también es una medida relativa pero no se basa en el tamaño heredado sino que se basa en el tamaño aplicado al tag `<html></html>`.

Valores en porcentajes

En el ejemplo anterior, el parrafo hereda los 10px del body y al aplicar un size de 150% resulta en 15px

```
body{
  font-size: 10px;
}
p{
  font-size: 150% /* 150% de 10px=15px*/
}
```

Valores en em

Teniamos que em, es una unidad de medida, en la cual 1em es igual al font-size heredado del padre.

Em funciona similar al porcentaje, en el siguiente ejemplo, el body se setea a 16 px, y ponemos a los h1 con 1.5ems, el cual hace que los h1 sean una y media veces su tamaño heredado o 24 px.

```
body{
  font-size: 16px;
}
h1{
  font-size: 1.5em; /*1.5 x 16px=24px*/
}
```

Mejores practicas con EM

Por lo general si se quiere usar, la solución es hacer que las medidas en em se muestren consistentes de acuerdo al tamaño setado al body de 100% (100% de acuerdo al tamaño por defecto que le dan los navegadores). Luego, se usa las medias en em para dimensionar los elementos de texto. **Esto mantiene la preferencia del usuario en cuanto al tamaño del texto y asegura que los elementos de texto son dimensionados proporcionalmente.**

Por ejemplo queremos que el body tenga el tamaño del 100% del que le setea el navegador, supongamos que resulte en 16px, de acuerdo a esa suposición, queremos que los article tengan 14px y luego que los h2 que aparecen dentro de los article tengan 18 px. El resultado es el siguiente:

```
body{
  font-size: 100%;
}
article{
  font-size: 0.875em;
  /*Este em se basa en el tamaño del body*/
}
article h2{
  font-size: 1.2857em;
  /*Este em se basa en el tamaño del article*/
}
```

Por la definición de em, es un porcentaje aplicado al tamaño heredado del padre, entonces si queremos que el article sea de 14 px con respecto a los 16px del body (ya que es el que lo contiene) se debe multiplicar por 0.875.

En cambio para los h2 contenidos en los article, hay que tener como referencia el tamaño de el elemento article y no el body ya que el article es el elemento mas cercano que contiene a los h2, entonces para que los h2 sean de 18px con respecto a los 14px de los article se debe multiplicar por 1.2857.

La formula por lo general aplicada para em es:

tamaño objetivo / tamaño del contenido (heredado) = resultado en em

Medidas en rem

Una medida que surgió en base a em es rem, la cual difiere en que, en lugar de basar se en el tamaño heredado, se basa en el tamaño del elemento html.

```
html{
  font-size: 100%;
}
article{
  font-size: 0.75rem;
  /*Este em se basa en el tamaño del html*/
}
```

La fórmula por lo general aplicada para rem es:

tamaño objetivo / tamaño del html = resultado en rem

Lo interesante de esto es que al aplicar em, rem al final el resultado es en pixeles por lo general.

Ejercicio 8-2

Como continuación de los estilos agregados en el ejercicio 8-1, realizar lo siguiente:

- Setear el tamaño del body al 100% según las reglas aplicadas por defecto por el navegador.
- Setear a los párrafos en 14px.
- Usando la medida em setear a los h1 en 1.5em y los h2 en 1em

Propiedad font-weight

La propiedad font-weight nos permite especificar si queremos que el texto aparezca en negrita o no.

Valores	Normal, bolder, bold, lighter , 100, 200, 300, 400 , 500, 600 , 700, 800, 900, inherit
Valor por defecto	Bold para los h1, ..., h6 Normal para todo los demas.
Aplica para	Todos los elementos
Se hereda?	si

Se usan solo dos valores:

- **normal** si queremos que el texto no aparezca en negrita.
- **bold** si queremos que el texto aparezca en negrita.
- Con los demás valores los navegadores suelen tener conflictos con las fuentes.

Ejemplo:

```
p{
  font-weight: bold;
}
```

Ejercicio 8-3:

A la hoja de estilo anterior ahora sumarle que los h4 no aparezcan en negrita.

Propiedad font-style

La propiedad font-style afecta la postura del texto, indicando si la forma debe ser vertical (normal) o inclinada (italic o oblique).

Valores	Normal, italic, oblique, inherit
Valor por defecto	normal
Aplica para	Todos los elementos
Se hereda?	SI

La diferencia entre italic y oblique es que la versión italic suele ser un diseño de tipo de letra separado con letras curvas mientras que el texto oblicuo adopta el diseño normal de la fuente y sólo se inclina. La verdad es que en la mayoría de los navegadores, no hay diferencias.

Ejemplo

```
p{
  font-style: italic;
}
```


Ejercicio 8-4:

A la hoja de estilo anterior sumarle:

- Los h2 en oblique.
- Los h4 en italic.

Propiedad font

La propiedad font permite especificar en una sola propiedad todas las propiedades de texto que vimos anteriormente

Valores	font-style font-weight font-size font-family inherit
Valor por defecto	Valores por defecto de cada propiedad listada.
Aplica para	Todos los elementos
Se hereda?	Si

Como vemos la propiedad font tiene una lista de valores de todas las propiedades que vimos antes, separados por espacios. El orden en que aparecen los valores es importante:

```
p{  
  font: style weight size font-family  
}
```

Como mínimo la propiedad font debe incluir el font-size y el font-family para usarla, en ese orden, de otra manera la regla es invalida.

Los otros valores deben aparecer antes que estas dos y lo pueden hacer en cualquier orden, **si son omitidos se les da el valor por defecto que es normal.**

Ejercicio 8-5:

Modificar la hoja de estilo anterior para los h1 usando la propiedad font (tener en cuenta las propiedades que no son obligatorias).

Mas selectores

Antes de seguir con las propiedades de fuentes vamos a ver un par mas de selectores:

Selector de descendientes

Este selector permite seleccionar elementos que están contenidos dentro de otro. Este selector se especifica como una lista de elementos separados por espacio en blanco. Por ejemplo, el siguiente selector:

Selecciona a los parrafos que aparecen dentro de un div.

```
div p{  
  color:blue;  
}
```

Este selector selecciona a los parrafos que aparecen dentro de un div y a su vez el div aparece dentro de un articulo

```
article div p{  
  color:blue;  
}
```

Como vemos podemos se puede anidar selectores de descendientes varios a través de varios elementos contenidos dentro de otros.

Selector hijo

Selecciona un hijo directo de un elemento, se usa el símbolo >, por ejemplo:

Selecciona a todos los p que tiene como padre a un div.

```
div>p{  
  color:blue;  
}
```

selector hermano adyacente

Selecciona a un elemento que esta inmediatamente después de otro y tienen el mismo padre. Se usa el símbolo +. Por ejemplo:

Esta regla selecciona los p que estan despues de un h1 y ambos tiene el mismo padre

```
h1 + p{  
  color:blue;  
}
```

```
<div >  
  <h1></h1>  
  <p>Este es el parrafo seleccionado</p>  
  <div>  
    <p></p>  
  </div>  
</div>
```

Selector general de hermano

Selecciona un elemento (1) que comparte el padre con otro elemento (2) y a su vez (1) esta precedido por (2).

```
h1 ~ p{  
  color:blue;  
}
```

```
<div >  
  <h1></h1>  
  <p>Este parrafo es seleccionado</p>  
  <div>  
    <p></p>  
  </div>  
  <p>Es parrafo es seleccionado</p>  
</div>
```

Selectores de ID

Cuando le poníamos un ID a un elemento html, este id debía ser único dentro del documento, por lo tanto el id es un nombre único que identifica al elemento y es buena idea usarlo dentro de css para seleccionarlo.

El selector de ID permite seleccionar un elemento a través de su ID, para ello el selector tiene la forma:

```
• #id_del_elemento{  
  • propiedad1: valor  
}
```

Es decir se usa el simbolo #+id para seleccionar a un elemento por su id, por ejemplo:

```
<p id="intro">  
  
</p>
```

Para seleccionar el p con el id = intro y darle el color rojo usamos el siguiente selector css:

```
• #intro{  
  color:red;  
}
```

Como vemos no es necesario poner p#intro ya que el id es único en todo el documento.

Selector de clase

Recordemos que los elementos html pueden tener el atributo class, el cual permite especificar una o mas clases que comparten los elementos html, si uno o mas elementos html tenían la misma clase significaba que estos compartían algún aspecto en común o formaban parte de un grupo. Por ende difiere del id el cual solo un elemento lo puede tener.

Se puede seleccionar elementos que tengan la misma clase usando el selector de clase, para esto se usa el punto (.) y luego el nombre de la clase:

```
.nombre_clase{  
  propiedad:valor;  
}
```

Por ejemplo, si tenemos varios div que comparten la clase publicidad y queremos que los elementos con la clase publicidad tengan font-size de 20 px y color verde podemos hacer lo siguiente:

```
<div class="publicidad">  
  <p>div 1 con clase publicidad</p>  
</div>  
<div class="publicidad">  
  <p>div 2 con clase publicidad</p>  
</div>  
<div class="publicidad">  
  <p>div 3 con clase publicidad</p>  
</div>
```

El selector CSS resultante sería:

```
.publicidad{  
  font-size: 20px;  
  color:green;  
}
```

Como varios elementos pueden compartir la misma clase se puede combinar el selector de clase con otros selectores para especificar que solo queremos aplicar una regla de estilo a ciertos elementos que tengan una clase y no a todos los elementos que la tengan. El mas sencillo es especificar el nombre del elemento html seguido de la clase.

Por ejemplo queremos que solo los div que contiene la clase publicidad tengan color verde y font-size 20 px, entonces:

```
<div class="publicidad">
  <p>div 1 con clase publicidad</p>
</div>
<article class="publicidad">
  <p>article 1 con clase publicidad</p>
</article>
<div class="publicidad">
  <p>div 2 con clase publicidad</p>
</div>
```

El selector CSS resultante sería:

```
/*Selecciona los div con la clase publicidad*/


div.publicidad{
  font-size: 20px;
  color:green;
}


```

Selector universal

En CSS hay un selector universal que selecciona cualquier elemento, es el símbolo `*`.

Por ejemplo:

```
/*Hace que todos los elementos tengan color rojo*/
*{
  color:red;
}
/*Hace que todos los elementos dentro del elemento
  con id intro tengan color azul
*/
#intro *{
  color:blue;
```


Propiedad color - Cambiamos el color del texto

Se puede cambiar el color del texto usando la propiedad color.

Valores	<ul style="list-style-type: none">• Valor de color numérico o nombre• inherit
Valor por defecto	Depende del navegador y el elemento html
Aplica para	Todos los elementos
Se hereda?	si

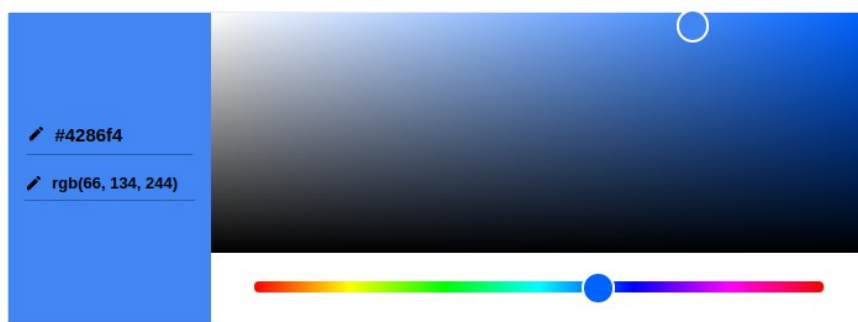
Los valores de colores puede ser un valor predefinido (nombre de color) o un valor numérico que especifica un color RGB (red-green-blue los colores primarios que forman cualquier color)

```
• h1{  
  color:gray;  
}  
• h1{  
  color:#666666;  
}  
• h1{  
  color:#666;  
}  
• h1{  
  color: rgb(102,102,102);  
}
```

Más adelante vamos a ver en detalle los colores, pero por ahora es importante sabe que hay colores por defecto y colores que se pueden especificar numéricamente. En total son 143 colores predefinidos, dejo un link para que lo vean:

https://www.w3schools.com/cssref/css_colors.asp

Si en google buscamos por color picker nos muestra un herramienta para buscar colores.



Ejercicio 8-6:

A la hoja de estilo anterior le vamos a sumar nuevas reglas aplicando los nuevos selectors y usando colores.

1. A los elementos strong que tengan como padre a un h4 darle color maroon.
2. Al elemento que tenga el id info darle el color teal.
3. A los parrafos que estén dentro del elemento con el id info hacer que se muestren en italic.
4. A los elementos que tengan la clase price hacer lo siguiente:
 1. que tengan como fuente a Georgia, y como familia de fuentes serif.
 2. Que aparezcan con italic.
 3. Que tengan color gray.
5. A los elementos con la clase label hacer que:
 1. aparezcan en negrita.
 2. La inclinación sea normal.
6. Los parrafos que tengan la clase warning hacer que:
 1. tengan color rojo.
 2. El tamaño sea x-small