

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 charter 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.

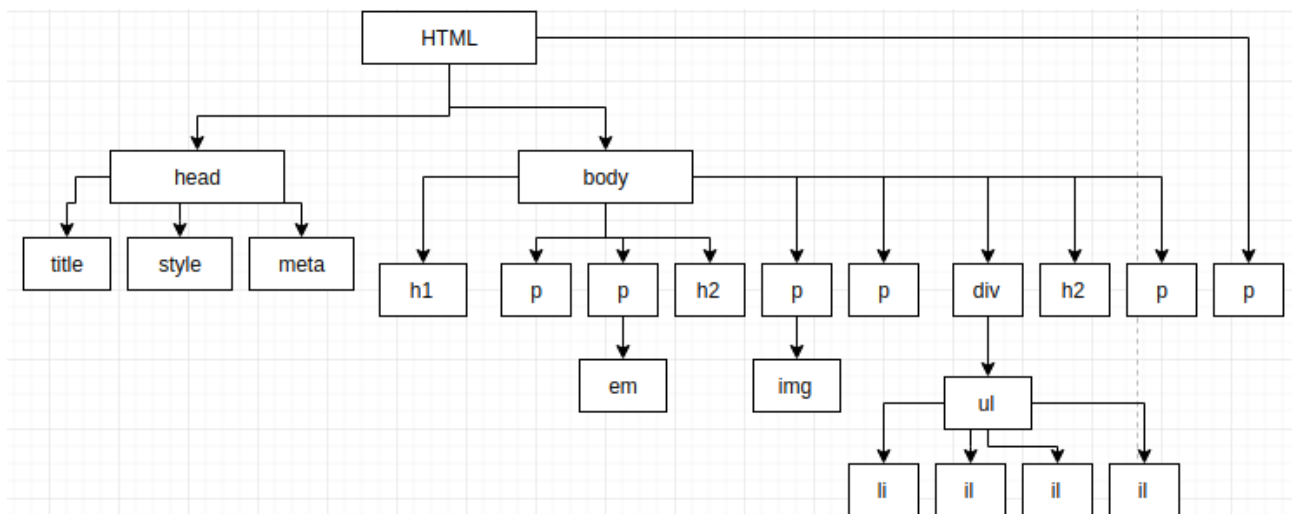
Hay fiesta señores



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`.

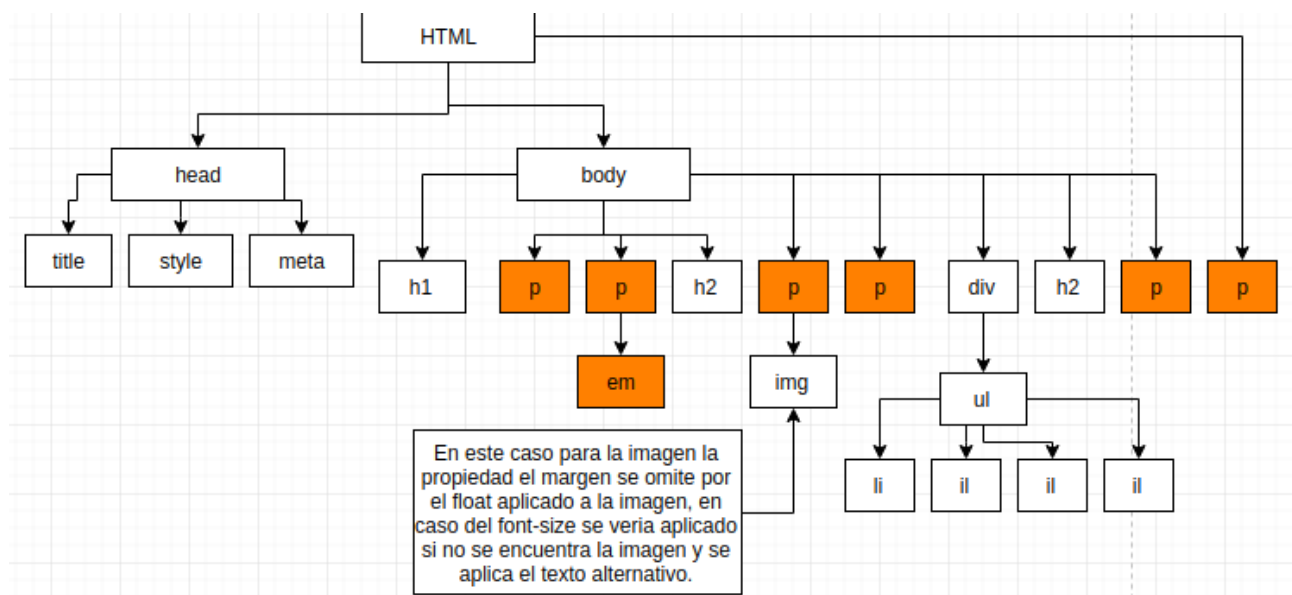
Todos los elementos que están mas arriba que un elemento en la jerarquía son sus **ancestros (ancestros)**. Dos o mas elementos con el mismo padre son **hermanos (siblings)**.

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

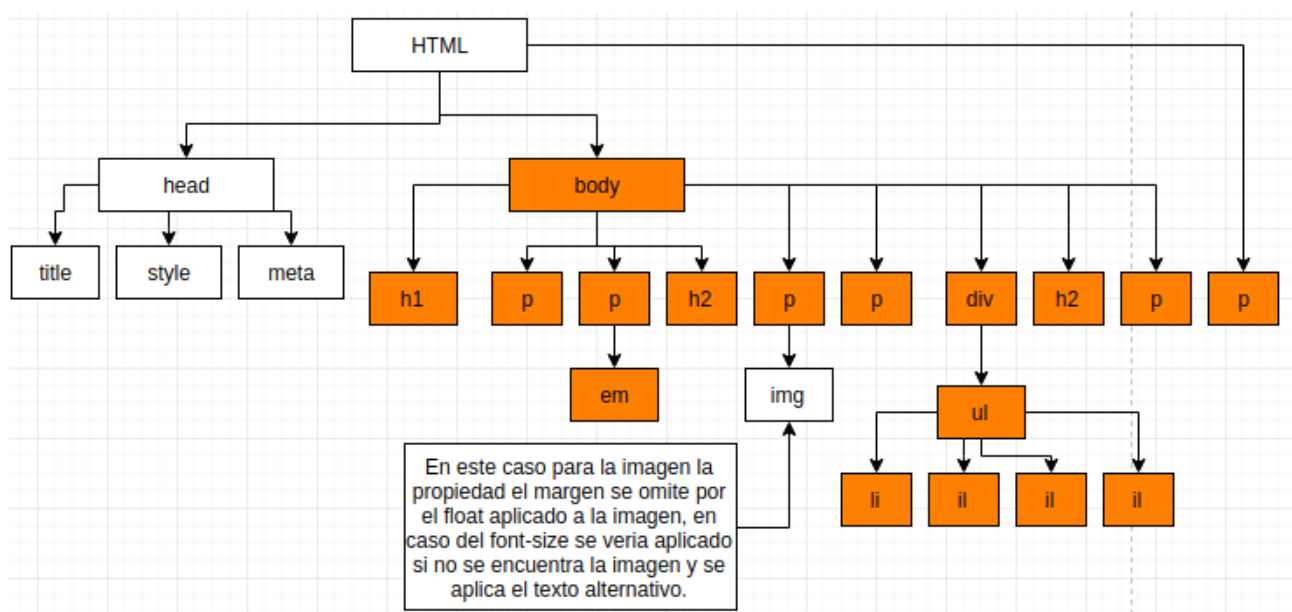
Como se transmiten los valores de propiedades a través de la herencia?

```
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;
}
```

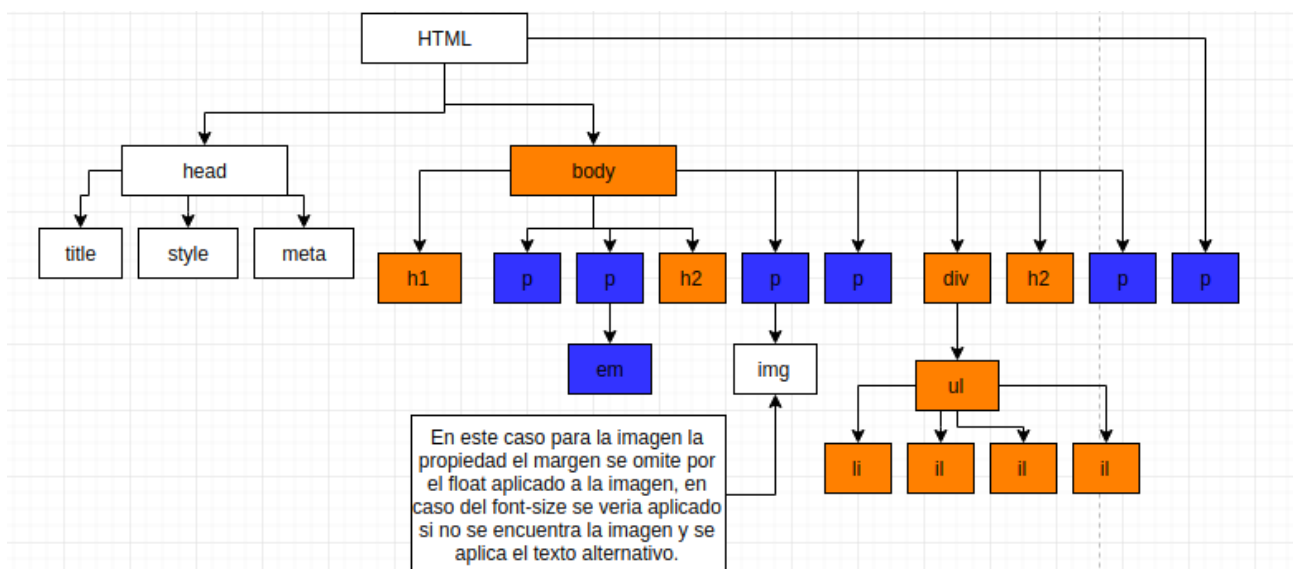
nos ahorramos de tener que escribir mas selectores para indicar que todos los elementos dentro de la página tengan que tener la fuente sans-serif.

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.

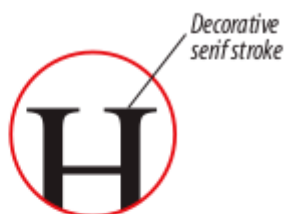
```
@font-face {
  font-family: 'Font_name';
  src: url('myfont-webfont.eot?#iefix') format('embedded-opentype'),
  url('myfont-webfont.woff') format('woff'),
  url('myfont-webfont.ttf') format('truetype'),
  url('myfont-webfont.svg#svgFontName') format('svg');
}
```

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.

RECORDATORIO:

En el caso de los valores que se puede usar em o porcentaje, el valor que se herda es el calculado, por ejemplo, si seteamos font-size en 2em el cual resulta en 14px, los 14px se heredan por los descendientes y no los 2em.

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;  
}
```

En este caso si un div contiene a un section y el section contiene un parrafo, el parrafo va a ser seleccionado. Como vemos solo se tiene que cumplir que en algún momento un div lo contenga, pero puede estar contenido dentro de otros elementos.

```
<div>  
  <p>Este parrafo se seleccion</p>  
</div>  
<div >  
  <section>  
    <p>Este parrafo se selecciona</p>  
  </section>  
</div>
```

Otro ejemplo, 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

Detalle de los selectores

Como vimos antes algunos selectores son mas detallados que otros y por ende tienen un peso mayor al momento de determinar que reglas de estilo se terminan aplicando en el caso de conflictos con varias reglas que quieren modificar la misma propiedad.

Ahora que aprendimos algunos selectores mas, es buen momento para saber como CSS les asigna peso según el detalle que tienen, la siguiente lista va de mas específico a menos específico:

1. Selectores de ID (son mas específicos y le ganan en peso a las reglas de 2, 3, 4).
2. Selectores de clase (son mas específicos y le ganan en peso a las reglas de 3, 4).
3. Selectores contextuales (Descendientes, hijos, hermanos adyacentes entre otros que vimos, son mas específicos y le ganan en peso a las reglas de 4).
4. Selectores de elementos individuales.

Veamos algunos ejemplos:

En este caso como las dos reglas afectan la misma propiedad, la segunda que es un selector contextual (selector de un descendiente) tiene mas peso que el selector de elemento individual p. Por ende la segunda regla es aplicada.

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

En e siguiente ejemplo la primera es sobre escrita por la segunda regla, pero como la tercera regla es un selector de clase le gana a las otras dos.

```
p{  
  color:red;  
}  
div p{  
  color: blue;  
}  
p.intro{  
  color:green;  
}
```

Propiedad line-height

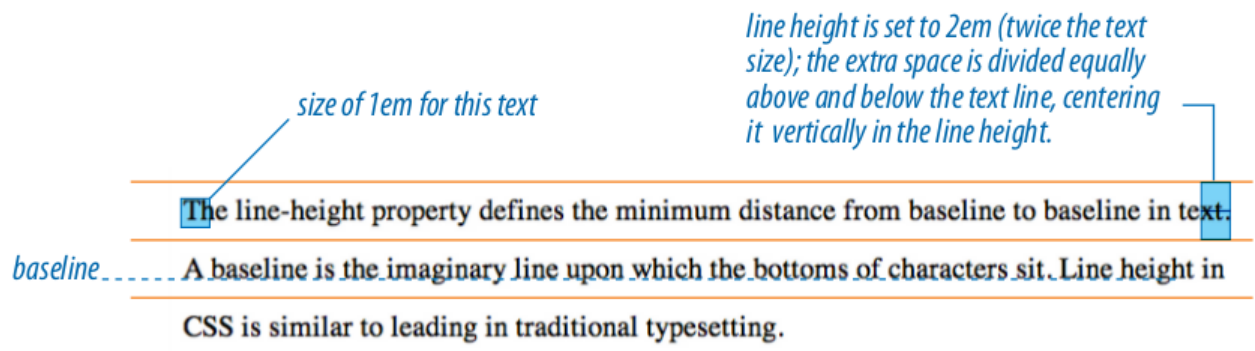
Esta propiedad define la distancia mínima desde una línea base de texto a otra. La línea base es la línea "imaginaria" sobre la cual el texto se sitúa (lo podríamos ver como un renglón). Este espacio es dividido equitativamente arriba y abajo del texto.

Valores	<ul style="list-style-type: none">• Número.• Unidad de medida (px, em, cm ...).• Porcentaje.• normal.• inherit
Valor por defecto	normal (heredado)
Aplica para	Todos los elementos html
Se hereda?	SI

En el caso que se especifique solo un número, este es como un factor de escala, se multiplica el font-size actual por el número.

Con EM y porcentaje ocurre lo mismo que antes, en este caso se basan en el font-size heredado del elemento.

```
p{
  line-height: 2;
}
p{
  line-height: 2em;
}
p{
  line-height: 200%;
}
```



```
line-height: 2em;
```

Propiedad text-indent

Esta propiedad permite darle indentación o sangría a la primera línea de texto.

Valores	<ul style="list-style-type: none">• Unidad de medida.• Porcentaje.• inherit
Valor por defecto	0
Aplica para	Elementos de bloque, celdas en tablas, bloques en línea.
Se hereda?	SI

En el caso de los porcentajes o em, se calcula en base al width del elemento padre.

```
p {  
  text-indent: 2em;  
}  
  
p.segundo {  
  text-indent: 25%;  
}  
  
p.tercero {  
  text-indent: 25px;  
}
```

The house we stayed in was called Pozzarello and it was built around the year 1200 as the home of the gardner who tended the grounds of the adjacent castle. The thick walls kept us nice and cool inside, despite the blistering mid-day heat. This is the view from our bedroom window. It looks a lot like the scene on the way to Sienna, and the scene on the way to the grocery store. We were surrounded by beautiful countryside for most of our travels. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum

The house we stayed in was called Pozzarello and it was built around the year 1200 as the home of the gardner who tended the grounds of the adjacent castle. The thick walls kept us nice and cool inside, despite the blistering mid-day heat. This is the view from our bedroom window. It looks a lot like the scene on the way to Sienna, and the scene on the way to the grocery store. We were surrounded by beautiful countryside for most of our travels. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum

The house we stayed in was called Pozzarello and it was built around the year 1200 as the home of the gardner who tended the grounds of the adjacent castle. The thick walls kept us nice and cool inside, despite the blistering mid-day heat. This is the view from our bedroom window. It looks a lot like the scene on the way to Sienna, and the scene on the way to the grocery store. We were surrounded by beautiful countryside for most of our travels. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum

Propiedad text-align

Esta propiedad permite alinear el texto como cualquier programa de edición de documentos.

Valores	<ul style="list-style-type: none">• left• right• center• justify• inherit
Valor por defecto	<ul style="list-style-type: none">• left para lenguajes que se leen de izquierda a derecha• right para lenguajes que se leen de derecha a izquierda.
Aplica para:	Elementos de bloque, celdas en tablas, bloques en línea.
Se hereda	si

```
p {  
  text-align: left; /*Alinea el texto al margen izquierdo*/  
  text-align:right; /*Alinea el texto al margen derecho*/  
  text-align: center; /*Centra el texto en el bloque de texto*/  
  text-align: justify; /*Alinea el texto en ambos márgenes izquierda y derecha*/  
}
```

The house we stayed in was called Pozzarello and it was built around the year 1200 as the home of the gardner who tended the grounds of the adjacent castle. The thick walls kept us nice and cool inside, despite the blistering mid-day heat. This is the view from our bedroom window. It looks a lot like the scene on the way to Siennam.

The house we stayed in was called Pozzarello and it was built around the year 1200 as the home of the gardner who tended the grounds of the adjacent castle. The thick walls kept us nice and cool inside, despite the blistering mid-day heat. This is the view from our bedroom window. It looks a lot like the scene on the way to Siennam.

The house we stayed in was called Pozzarello and it was built around the year 1200 as the home of the gardner who tended the grounds of the adjacent castle. The thick walls kept us nice and cool inside, despite the blistering mid-day heat. This is the view from our bedroom window. It looks a lot like the scene on the way to Siennam.

The house we stayed in was called Pozzarello and it was built around the year 1200 as the home of the gardner who tended the grounds of the adjacent castle. The thick walls kept us nice and cool inside, despite the blistering mid-day heat. This is the view from our bedroom window. It looks a lot like the scene on the way to Siennam.

También esta la propiedad **vertical-align**, la cual alinea verticalmente al texto.

Valores	baseline, sub, super, top, text-top, middle, textbottom, bottom, porcentaje, unidad de medida, inherit.
Valor inicial	baseline
Se hereda	no

Propiedad text-decoration

Esta propiedad permite agregar subrayado, sobre rayado, tachado al texto.

Valores:	<ul style="list-style-type: none">• none• underline• overline• line-through
Valor por defecto	none
Aplica para	Todos los elementos
Se hereda	no

```
p {  
  text-decoration: none;  
  text-decoration: underline;  
  text-decoration: overline;  
  text-decoration: line-through;  
}
```

The house we stayed in was called Pozzarello and it was built around the year 1200

The house we stayed in was called Pozzarello and it was built around the year 1200

The house we stayed in was called Pozzarello and it was built around the year 1200

~~The house we stayed in was called Pozzarello and it was built around the year 1200~~

Por ejemplo si a un link le queremos sacar el subrayado:

```
a{  
  text-decoration: none;  
}
```

Propiedad text-transform

Esta propiedad permite capitalizar, transformar a mayúsculas o minúsculas.

Valores	<ul style="list-style-type: none">• none.• capitalize• lowercase• uppercase• inherit
Valor por defecto	none
Aplica para	Todos los elementos
Se hereda	si

```
p{
  text-transform: none; /*El texto se muestra por defecto*/
}
p{
  text-transform: capitalize; /*Pone la primera letra de cada palabra en mayúscula*/
}
p{
  text-transform: lowercase; /*Pone todo el texto en minúsculas*/
}
p{
  text-transform: uppercase; /*Pone todo el texto en mayúsculas*/
}
```

The house we stayed in was called Pozzarello and it was built around the year 1200

The House We Stayed In Was Called Pozzarello And It Was Built Around The Year 1200

the house we stayed in was called pozzarello and it was built around the year 1200

THE HOUSE WE STAYED IN WAS CALLED POZZARELLO AND IT WAS BUILT AROUND
THE YEAR 1200

Propiedad text-shadow

Esta propiedad permite agregar sombreado detrás del texto, pero adelante del fondo y borde si es que se tiene. El sombreado es soportado en todos los navegadores excepto en IE10.

Valores	<ul style="list-style-type: none">• 4 valores:<ul style="list-style-type: none">◦ offset horizontal.◦ Offset vertical.◦ blur radius◦ color• none
Valor por defecto	none
Aplica para	Todos los elementos
Se hereda	si

El valor para configurar el sombreado son 3 valores de unidades de medidas y un color, los valores se separan por coma. La declaración mínima solo requiere el offset horizontal, vertical y color.

```
h1{
  color:darkgreen;
  text-shadow: .2em .2em silver;
}
h2{
  color:darkgreen;
  text-shadow: .3em .3em silver;
}
```

```
<h1>Bienvenidos al show</h1>
<h2>Bienvenidos al show</h2>
```

Bienvenidos al show

Bienvenidos al show

- El primer valor es el offset horizontal, el cual posiciona al sombreado a la derecha del texto con un valor positivo y a la izquierda con un valor negativo.
- El segundo valor es el offset vertical, el cual posiciona al sombreado abajo de texto con un valor positivo y arriba del texto con un valor negativo.
- El color especifica el color del sombreado.

El valor que falta especificar es el blur radius el cual desenfoca o hace el sombreado borroso, este se especifica como tercer parametro de la propiedad.

- Un valor de cero (0), el cual es el por defecto si no se especifica ya que no es obligatorio, el sombreado aparece normal.
- Valores mayores a cero hacen que el sombreado se vuelva mas suave y difuminado.
- Usualmente se prueba hasta que se encuentra el valor ideal.

```
h1{
  /*Con blur radius*/
  color:darkgreen;
  text-shadow: .2em .2em .2em silver;
}
h2{
  /*sin blur radius*/
  color:darkgreen;
  text-shadow: .3em .3em silver;
}
```

**Bienvenidos al
show**

Bienvenidos al show

Se pueden aplicar varias sombras a un elemento cada una separada por coma, en este caso, se aplica la primera en la lista, y luego las demas son agregadas en orden detras de la primera. También se puede hacer que el texto tenga efecto brillo al solo agregar una sombra detras del texto con difuminado y color.

```
h1{
  /*múltiples sombras*/
  color:black;
  text-shadow: -.7em -.5em .2em silver,
               .2em .2em .1em gray;
}
h2{
  /*texto con efecto brillo*/
  color:black;
  text-shadow: 0 0 .7em purple;
}
```

Bienvenidos al show

Bienvenidos al show

Modificando las listas

Propiedad list-style-type

Esta propiedad permite modificar el marcador que aparece en las listas ya sean ordenadas o no.

Valores	None, disc, circle, square, decimal, decimal-leading-zero, lower-alpha, upper-alpha, lower-latin, upper-latin, lower-roman, upper-roman, lower-greek, inherit
Valor por defecto	Disc en ul y decimal en ol
Aplica para	ul, li, ol, o elementos con la propiedad display con el valor list-item
Se hereda	Si
Utilidad	El valor none remueve el marcador. Este se usa en las barras de navegación.

Keyword	System
decimal	1, 2, 3, 4, 5...
decimal-leading-zero	01, 02, 03, 04, 05...
lower-alpha	a, b, c, d, e...
upper-alpha	A, B, C, D, E...
lower-latin	a, b, c, d, e... (same as lower-alpha)
upper-latin	A, B, C, D, E... (same as upper-alpha)
lower-roman	i, ii, iii, iv, v...
upper-roman	I, II, III, IV, V...
lower-greek	α , β , γ , δ , ϵ ...

Propiedad list-style-position

Esta propiedad indica si los marcadores están dentro o fuera del area de contendo para los elementos de la lista.

Valores	inside, outside, inherit
Valor por defecto	outside
Aplica para	ul, li, ol, o elementos con la propiedad display con el valor list-item
Se hereda	si

```
li {background-color: #F99;}  
ul#outside {list-style-position: outside;}  
ul#inside {list-style-position: inside;}
```

```
<ul id="outside">  
  <li>Elemento</li>  
  <li>Elemento</li>  
  <li>Elemento</li>  
  <li>Elemento</li>  
</ul>  
<ul id="inside">  
  <li>Elemento</li>  
  <li>Elemento</li>  
  <li>Elemento</li>  
  <li>Elemento</li>  
</ul>
```

-
- Elemento
 - Elemento
 - Elemento
 - Elemento

- Elemento
- Elemento
- Elemento
- Elemento

Propiedad list-style-image

Esta propiedad permite cambiar el marcador por una imagen.

Valores	url, none, inherit
Valor por defecto	none
Aplica para	ul, li, ol, o elementos con la propiedad display con el valor list-item
Se hereda	si
A tener en cuenta	Especificar de igual manera la propiedad list-style-type, en caso que la propiedad no sea soportada por el navegador

- Elemento
- Elemento
- Elemento
- Elemento

```
ul{  
  list-style-image: url(cuadrado-rosa.gif);  
  list-style-type: disc;  
  list-style-position: outside;  
}
```

```
<ul>  
  <li>Elemento</li>  
  <li>Elemento</li>  
  <li>Elemento</li>  
  <li>Elemento</li>  
</ul>
```


Ejercicio 8-7

Modificar la hoja de estilo con la que venimos trabajando con lo siguiente:

1. Al body agregarle un espacio entre líneas base de 1.75em y que el texto este centrado.
2. Eliminar la regla de estilo que le da color teal al elemento con id info.
3. Hacer que los h1 tengan color purple.
4. Hacer que los parrafos descendientes de el elemento con el id info tengan color gray.
5. Hacer que los h2 aparezcan en mayusculas y en color purple.
6. Hacer que los parrafos que aparecen despues de los h2 (selector h2 + p) estén en italic.
7. Hacer que los h4 tengan el color sienna.
8. Agregarle un sombreado a los h1 con offset horizontal de .1em , vertical de .1em , blur radius de .2em y color lightslategray.

Colores y backgrounds

Existen dos formas de especificar el valor de un color en CSS:

- Colores predefinidos.
- Valor numérico que describe un color.

Colores predefinidos

En cuanto a los colores predefinidos es la opción que venimos usando hasta el momento. Solo tenemos que especificar el nombre del color.

```
h1,h2{  
  color: purple;  
}
```

Actualmente CSS3 soporta 143 colores predefinidos, el siguientes es un link con todos los nombres:

https://www.w3schools.com/colors/colors_names.asp

Valores numéricos

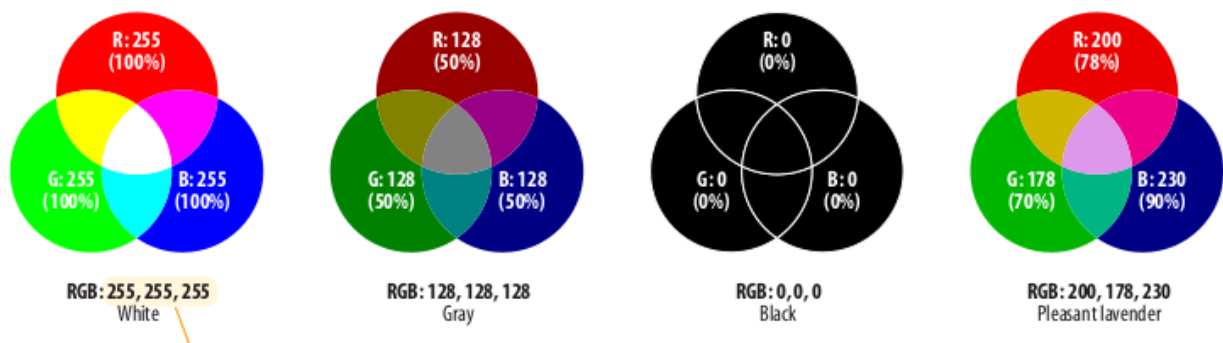
Para especificar colores que no aparecen entre los predefinidos (los cuales estan limitados) se usan valores numéricos a través del sistema de colores RGB (red-green-blue), mediante el cual podemos especificar millones de colores diferentes.

Que es RGB?

Las computadoras crean los colores combinando 3 colores: rojo, verde y azul (red, green y blue). Esto se conoce como el modelo de color RGB. CSS permite especificar cuanto de cada color queremos combinar para formar otro color.

En CSS cada uno de los 3 colores se especifica con una cantidad que va de 0 a 255, donde 0 es nada y 255 es todo.

The RGB color model



En CSS se especifica usando el valor `rgb(cantidad rojo, cantidad verde, cantidad azul)`, por ejemplo:

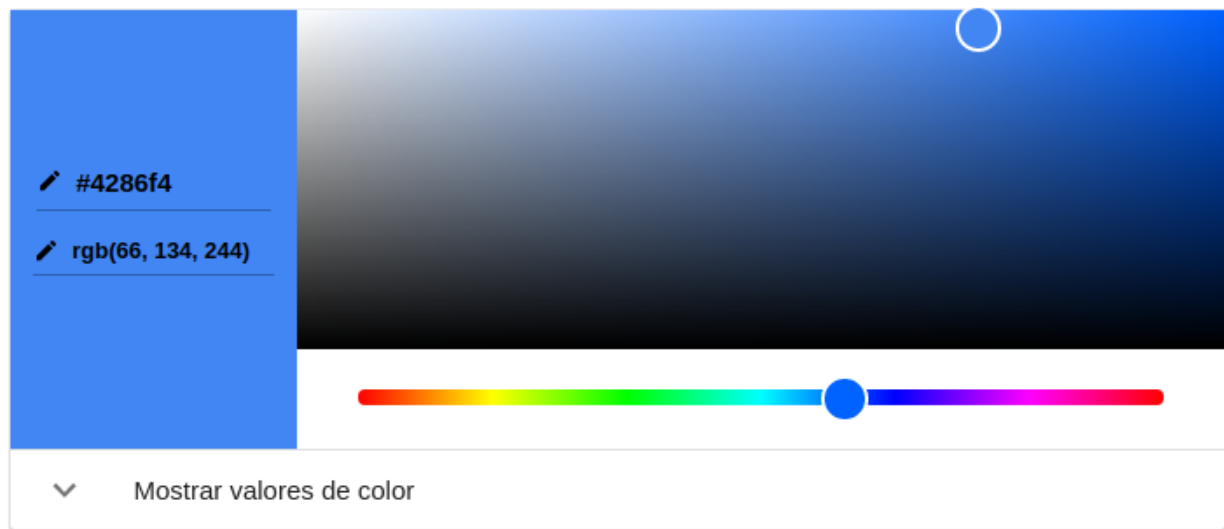
```
p{  
  color: rgb(120, 178, 100);  
  color: rgb(0, 0, 0);  
}
```

`rgb(0,0,0)` indica el negro

`rgb(255,255,255)` es el blanco

Seleccionar un color

La herramienta usada para ayudar a elegir colores de esta forma es el color picker. Si buscamos en google hay varias opciones e inclusive google por si mismo ofrece una opción.



Comentarios

Opciones para escribir en CSS

Vimos la opción de usar rgb(rojo, verde, azul) con valores numéricos que van de 0 a 255 para cada uno.

```
p{
  color: rgb(120, 178, 100);
  color: rgb(0,0,0);
}
```

También se puede usar **porcentajes** de cada color

```
p{
  color: rgb(20%, 10%, 40%);
}
```

O bien la tercera opción usando el sistema de numeración **hexadecimal**, para ello un valor de color tiene la siguiente forma:



Cada letra puede tener los valores desde 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F (están ordenados de menor a mayor). El color picker también ofrece este numero para facilitar su elección. Donde 0 es el numero mas chico y F el mas grande. Veamos algunos ejemplos:

```
p{
  color: #FFFFFF;
  color: #000000;
  color: #A6B6E6;
}
```

Esta forma requiere como mínimo 3 números, al igual que antes todos 0 es negro y todos F es blanco y en ese rango se mueven los colores. También es indistinto si las letras están en mayúsculas o minúsculas pero es ideal que se usen unas u otras.

RGBa color

La función RGBa color de css permite especificar un color y a su vez su transparencia u opacidad. Tiene la forma:

rgba(rojo, verde, azul, opacidad);

donde el valor de la opacidad va de 0 a 1:

- 0 totalmente transparente.
- 1 totalmente opaco (color verdadero).

```
p {
  color: rgba(255, 0, 0, 0.1);
  color: rgba(255, 0, 0, 0.2);
  color: rgba(255, 0, 0, 0.3);
  color: rgba(255, 0, 0, 0.4);
  color: rgba(255, 0, 0, 0.5);
  color: rgba(255, 0, 0, 0.6);
  color: rgba(255, 0, 0, 0.7);
  color: rgba(255, 0, 0, 0.8);
  color: rgba(255, 0, 0, 0.9);
  color: rgba(255, 0, 0, 1);
}
```

```
p {
  color: rgba(120, 0, 120, 0.1);
  color: rgba(120, 0, 120, 0.2);
  color: rgba(120, 0, 120, 0.3);
  color: rgba(120, 0, 120, 0.4);
  color: rgba(120, 0, 120, 0.5);
  color: rgba(120, 0, 120, 0.6);
  color: rgba(120, 0, 120, 0.7);
  color: rgba(120, 0, 120, 0.8);
  color: rgba(120, 0, 120, 0.9);
  color: rgba(120, 0, 120, 1);
}
```

Ahora veamos algunas propiedades relacionadas a colores.

Propiedad Foreground (primer plano) color

Esta propiedad permite especificar el color del texto y del borde de un elemento html.

Valores	<ul style="list-style-type: none">• Un valor de color• inherit
Valor por defecto	Depende del navegador
Aplica para	Todos los elementos
Se hereda	si

Por ejemplo tenemos dos parrafos uno con borde y otro sin, le queremos dar el color #1C61CE tanto al texto como al borde, entonces:

```
p{
  color:#1C61CE;
}
.borde{
  border: 4px dashed;
}
```

```
<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
</p>
<p class="borde">
  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
</p>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

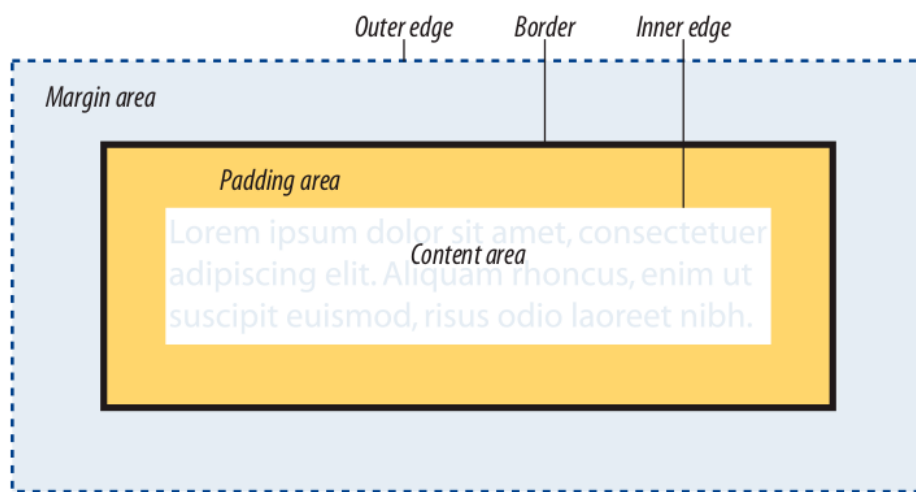
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Propiedad background-color

Esta propiedad aplica un fondo de color a cualquier elemento.

Valores	<ul style="list-style-type: none">• Un valor de color• transparent• inherit
Valor por defecto	transparent
Aplica para	Todos los elementos
Se hereda	si

Esta propiedad rellena con un color hasta el borde de un elemento html teniendo en cuenta le modelo en caja (próxima unidad).



```
p {  
  background-color: #1C61CE;  
  /*MODELO EN CAJA PROCIMA UNIDAD*/  
  border: 4px dashed;  
  padding: 20px;  
  margin: 20px;  
  outline: 5px dotted red;  
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Propiedad opacity

Esta propiedad cumple la misma función que el valor de opacidad en la función rgba. Permite especificar con un número de 0 a 1 cuan opaco es un color.

Valores	<ul style="list-style-type: none">Número de 0 a 1
Valor por defecto	1
Aplica para	Todos los elementos
Se hereda	no
A tener en cuenta	<ul style="list-style-type: none">Afecta a la propiedad background-color y color.Si solo queremos afectar una de las dos se debe usar la función rgba.

```
.primero,.segundo, .tercero {  
  color: blue;  
  background: lightblue;  
  opacity: .25;  
}  
  
.primero {  
  opacity: .25;  
}  
  
.segundo {  
  opacity: .5;  
}  
  
.tercero {  
  opacity: 1;  
}
```

```
<h1 class="primero">Hola a todos!</h1>  
<h1 class="segundo">Hola a todos!</h1>  
<h1 class="tercero">Hola a todos!</h1>
```

Hola a todos!

Hola a todos!

Hola a todos!

```
.primero{
  /*OPACIDAD APLICADA SOLO A LA PROPIEDAD COLOR*/
  color: rgba(0,0,255,0.4);
  background-color: lightblue;
}
.segundo{
  /*OPACIDAD APLICADA A AMBAS PROPIEDADES*/
  color: blue;
  background-color: lightblue;
  opacity: 0.4
}
```

Hola a todos!

Hola a todos!

Pseudo selectores de clases

El ejemplo mas claro para introducir esto son los links, sin agregar estilos, por defecto un link es de un color antes de hacerle click y una vez que le hacemos click, el navegador mantiene un seguimiento de esto, las próximas veces que veamos al link va a tener otro color.

```
<body>
  <h1><a href="https://es.wikipedia.org/">Ir a wikipedia</a></h1>
</body>
```

Antes de hacer click por primera vez al link:

Ir a wikipedia

Despues de hacerle click:

Ir a wikipedia

Esto se debe a que el navegador mantiene un seguimiento de los links que han sido clickeados, pero no solo controla los links, sino que mantiene seguimiento de otros eventos o estados como:

- El cursor del mouse esta sobre un elemento.
- Si hicimos foco sobre un elemento.
- Si un elemento es el primero de su tipo.
- Si un elemento es el ultimo hijo de su padre.
- Entre otros.

En CSS se puede aplicar estilos a estos estados o eventos que mantiene el navegador usando **selectores Pseudo-class**.

Los pseudo selectores se indican con dos puntos (:) y van siempre despues de un elemento. Por ejemplo: un link que fue visitado le quiero dar color rojo

```
a:visited{
  color:red;
}
```

Pseudo class selectores de Links

Estos selectores seleccionan links dependiendo si fueron clickeados o no:

:link aplica estilos a links no visitados o clickeados.

:visited aplica estilos a links que han sido clickeados o visitados.

Por defecto los navegadores usan las siguientes reglas:

```
/*Configuracion de los navegadores*/  
a:link{  
    text-decoration: underline;  
    color:blue;  
}  
a:visited{  
    color: purple;  
}
```

Por ejemplo si queremos hacer los links no visitados rojos y sin subrayado, los visitados en color verde.

```
a:link{  
    text-decoration: none;  
    color:red;  
}  
a:visited{  
    color: green;  
}
```

Ir a wikipedia
Ir a wikipedia

Pseudo-class selectores de acciones de usuario

Estos selectores seleccionan elementos que resultan de diferentes acciones de usuario como:

:focus se aplica cuando un elemento es seleccionado y listo para recibir datos (formularios)

:hover se aplica cuando el puntero del mouse esta sobre el elemento.

:active se aplica cuando el elemento (como un link o boton) esta en proceso de ser clickeado o darle un touch en pantallas tactiles

selector :focus

Cuando en un formulario hacemos click en un campo para escribir, el navegador le hace foco a ese campo. Cuando un elemento es destacado y listo para recibir datos de entrada se dice que tiene foco. El selector :focus permite aplicar estilos a elementos cuando tiene foco.

Por ejemplo hacer que el fondo de un campo de formulario sea amarillo cuando tiene foco.

```
<form class="" action="index.html" method="post">
  <label for="nombre">Nombre:</label>
  <input type="text" name="nombre" value="">
</form>
```

```
input:focus{
  background-color: yellow;
}
```

Nombre:

Nombre:

Selector :hover

Este selector selecciona un elemento mientras el usuario tiene el cursor sobre el mismo. Se puede aplicar casi a todos los elementos html, la mayoría de las veces es usado para cambiar la apariencia de los links cuando el usuario se posiciona sobre ellos para indicar que una acción es posible.

Por ejemplo hacer que un link tenga fondo azul y letra negra cuando se le apunta.

```
a:hover{
  color: black;
  background-color: lightblue;
}
```



Es bueno saber que este selector no está disponible en dispositivos con pantallas táctiles.

Selector :active

Este selector aplica a elementos mientras están en proceso de ser activados. En el caso de un link este selector es aplicado mientras se le mantiene click presionado al link o mientras se presiona con el dedo en una pantalla táctil. Por lo general los estilos aplicados pueden ocurrir muy rápido.

Por ejemplo hacer que los links tengan color rojo mientras están siendo presionados y fondo negro.

```
a:active{
  color: red;
  background-color: black;
}
```



Recomendaciones:

- Por lo general se especifican los estilos para todos los estados de un link, a modo de efecto visual me parece que permite una manera de dar feedback al usuario de que algo pasa con el link en diversos estados. Y los usuarios esperan este feedback.
- El focus se suele especificar ya que si el usuario llega al link apretando tab el focus entra en funcionamiento, por lo general se le aplica el mismo estilo que con hover, pero no es requerido.
- Para evitar que algún estado sobrescriba a otro el orden recomendado es:

```
a{}
a:link{}
a:visited{}
a:focus{}
a:hover{}
a:active{}
```

```

a {
  text-decoration: none;
}
a:link {
  color: maroon;
}
a:visited {
  color: gray;
}
a:focus {
  color: maroon;
  background-color: #ffd9d9;
}
a:hover {
  color: maroon;
  background-color: #ffd9d9;
}
a:active {
  color: red;
  background-color: #ffd9d9;
}

```

Samples of my work:

- Pen and Ink Illustrations
- Paintings
- Collage

[a:link](#)

Links are maroon and not underlined.

Samples of my work:

- Pen and Ink Illustrations
- Paintings
- Collage

[a:focus](#)

[a:hover](#)

While the mouse is over the link or when the link has focus, the pink background color appears.

Samples of my work:

- Pen and Ink Illustrations
- Paintings
- Collage

[a:active](#)

As the mouse button is being pressed, the link turns bright red.

Samples of my work:

- Pen and Ink Illustrations
- Paintings
- Collage

[a:visited](#)

After that page has been visited, the link is gray.

Contenido generado con los selectores :before y :after

Estos selectores permiten insertar contenido antes y despues de un elemento sin tener que agregar ese contenido en el documento html. Por ejemplo para insertar iconos a ciertos elementos con ciertas clases.

Por ejemplo, un parrafo que describie un cuento, entonces antes de empezar queremos agregar Érase una vez y al final queremos agregar El fin.

El pseudo selector **:before** permite agregar contenido antes de un elemento.

El pseudo selector **:after** permite agregar contenido despues de un elemento.

Casi todas las propiedades css son validas dentro de este selector

```
p:before{
  content: "Éras una vez,";
  font-weight: bold;
  color:purple;
}
p:after{
  content: "El fin.";
  font-weight: bold;
  color:purple;
}
```

```
<body>
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
    eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad
    minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea
    commodo consequat. Duis aute irure dolor in reprehenderit in voluptate
    velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat
    cupidatat non proident, sunt in culpa qui officia deserunt mollit anim
    id est laborum.</p>
</body>
```

Éras una vez, Lorem ipsum dolor sit amet, consectetur
Ut enim ad minim veniam, quis nostrud exercitation
reprehenderit in voluptate velit esse cillum dolore eu
officia deserunt mollit anim id est laborum.**El fin.**

Existen muchos selectores, hasta el momento vimos los mas importantes y de mayor uso, a continuación, adjunto el link donde los pueden encontrar a todos:

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

Es difícil poder acordarse de memoria de todos los selectores, lo ideal es manejar los de mayor uso, si en algún momento un selector es muy largo en cuanto a su definición, complejo o rebuscado, es buena opción ver en el listado y tratar de encontrar alguno otro selector que permita hacerlo mas sencillo.

Ejercicio 9-1

Para todos los ejercicios N°9 se va a trabajar con el archivo menu.html, menustyles.css y la carpeta imágenes.

1. Dar color #993399 a los h1.
2. Dar color #CC6600 a los h2.
3. Hacer que el fondo de la página tenga color #D2DC9D.
4. Dar al header un color blanco con transparencia del 0.5
5. Hacer que los links tengan el mismo color lila que los h1.
6. Hacer que los links visitados tengan color #937393
7. Cuando el mouse este sobre los links, hacer que el texto tenga color #C700F2 y fondo #FFFFFF. Hacer lo mismo cuando el link tiene foco.
8. Cuando se mantiene presionado el click sobre el link hacer que tenga color #FF00FF y fondo blanco.

Background imagen

Anteriormente vimos como añadir imagenes al contenido del documento html usando el tag ``, pero si queremos añadir imagenes en sentido decorativo o para aportar al estilo de la página mas que a la estructura estas imagenes son añadidas como backgrounds usando CSS.

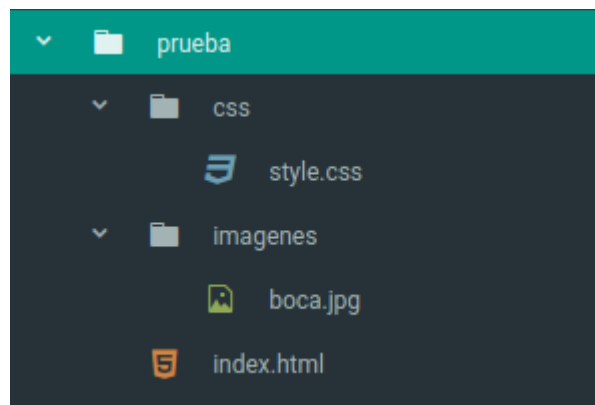
Existe una colección de propiedades usadas para colocar imagenes como fondos.

Propiedad background-image

Esta propiedad coloca una imagen como fondo de cualquier elemento.

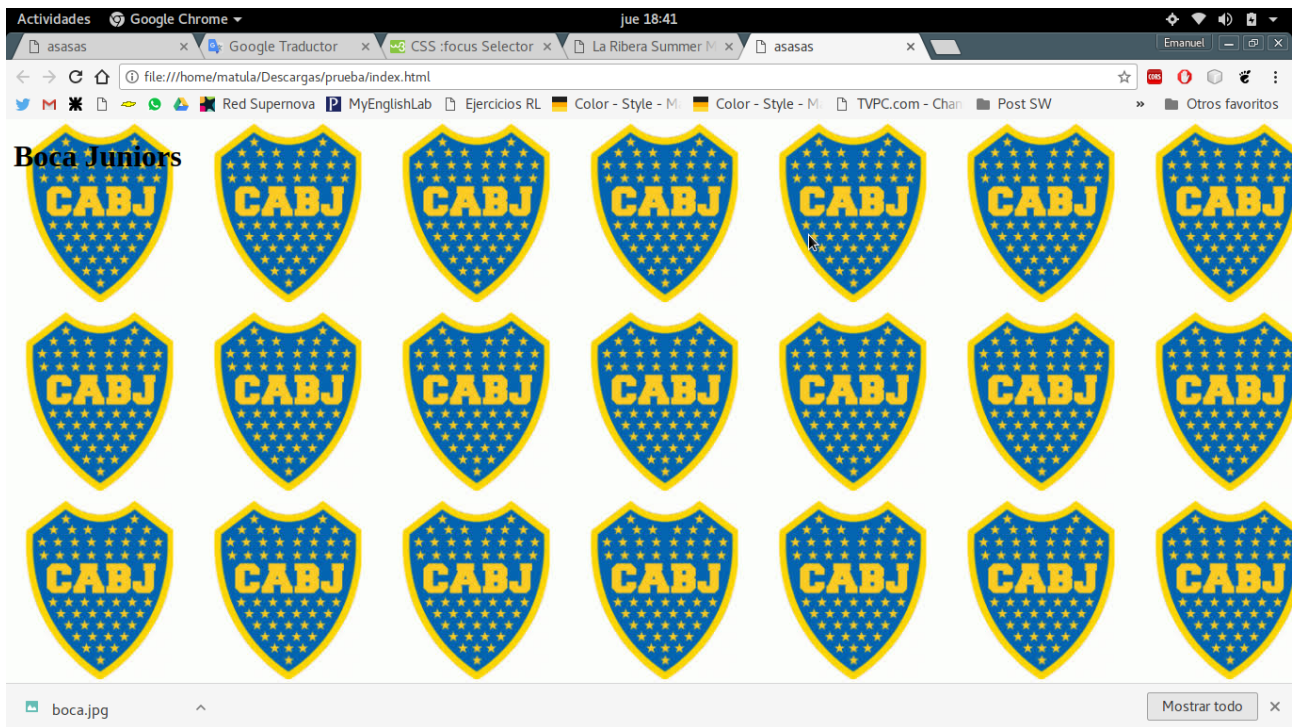
Valores	<ul style="list-style-type: none">• URL con la ubicación de la imagen.• none• inherit
Valor por defecto	none
Aplica para	Todos los elementos
Se hereda	no

Ejemplo: Quiero colocar de fondo de la página una imagen del escudo de boca de 200px por 200px.



```
body{  
  background-image: url(../imagenes/boca.jpg);  
}
```

```
<!DOCTYPE html>  
<html>  
  
  <head>  
    <meta charset="utf-8">  
    <title>asasas</title>  
    <link rel="stylesheet" href="css/style.css">  
  </head>  
  <body>  
    <h1>Boca Juniors</h1>  
  </body>  
</html>
```

Consideraciones:

- La imagen se ubica en la esquina superior izquierda y se dibuja horizontal y verticalmente hasta que ocupa su tamaño original, si no ocupa todo el espacio esta se repite hasta llenar todo el espacio disponible (esto se puede cambiar).
- Esta propiedad actúa igual que el background-color completando el fondo de los mismo elementos del modelo en caja.
- Si también especificamos el background color, la imagen se dibuja por encima de este, en caso de que se falle al cargar la imagen el color es visualizado, es buena práctica especificar un color parecido en caso de que la imagen no cargue.

Ejercicio 9-2

Continuando con la hoja de estilo del ejercicio 9-1 hacer que el body tenga como fondo la imagen bullseye.png ubicada en la carpeta images. No borrar la propiedad del color de fondo. ¿Cuál es el resultado? (la imagen es transparente)

Propiedad background-repeat

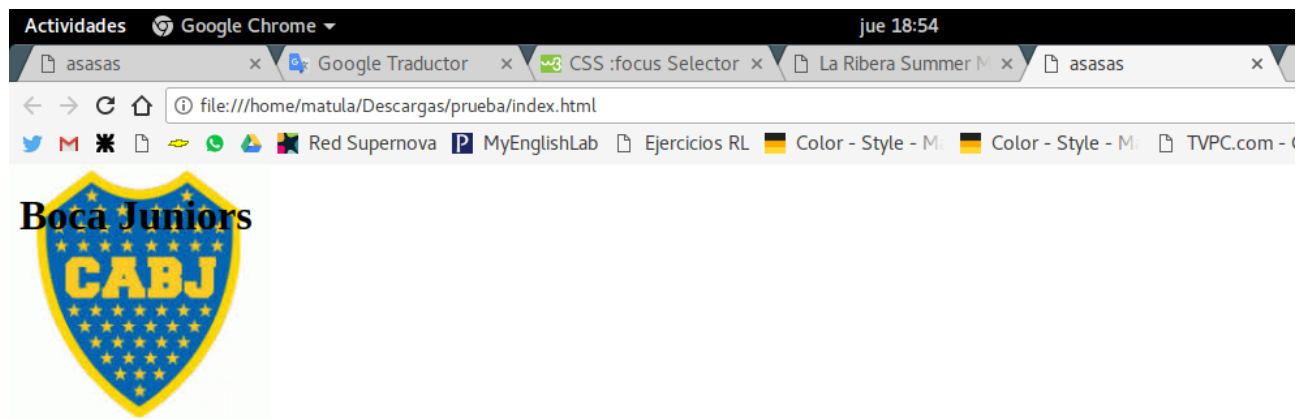
Como vimos en el ejercicio y ejemplo anterior, las imagenes que no tiene un tamaño igual al total del elemento en el cual se colocan como fondo se repiten hasta ocupar el tamaño total, particularmente de izquierda hacia la derecha y luego de arriba para abajo.

Este comportamiento se puede modificar con la propiedad background-repeat.

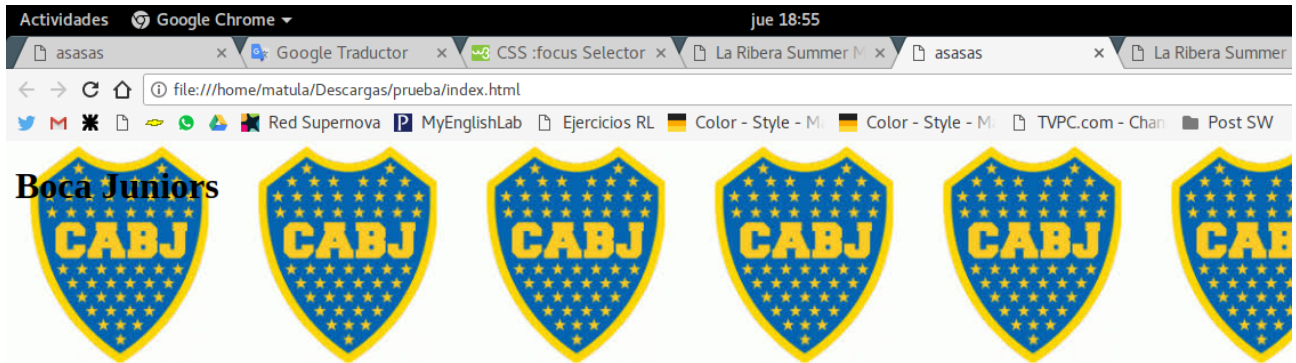
Valores	<ul style="list-style-type: none">• repeat.• repeat-x• repeat-y• no-repeat.• inherit
Valor por defecto	repeat
Aplica para	Todos los elementos
Se hereda	no

Por ejemplo si queremos que el escudo de boca solo aparezca una vez:

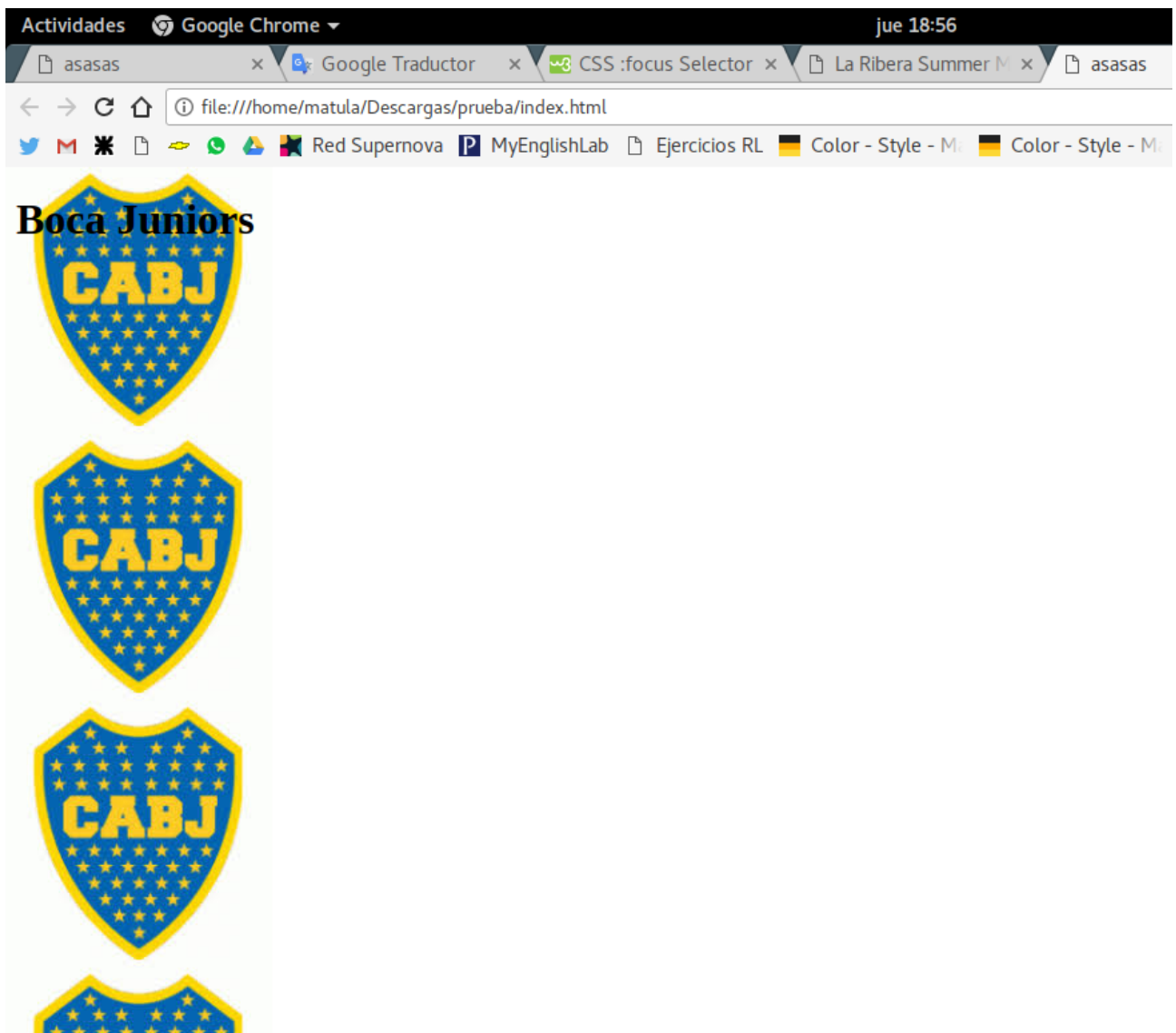
```
body{  
    background-image: url(../imagenes/boca.jpg);  
    background-repeat: no-repeat;  
}
```



```
body{
  background-image: url(../imagenes/boca.jpg);
  background-repeat: repeat-x;
}
```



```
body{
  background-image: url(../imagenes/boca.jpg);
  background-repeat: repeat-y;
}
```



Ejercicio 9-3

En el header agregar como fondo la imagen purpledotted.png de la carpeta images con el patrón de repetición repeat-x.

Propiedad background-position

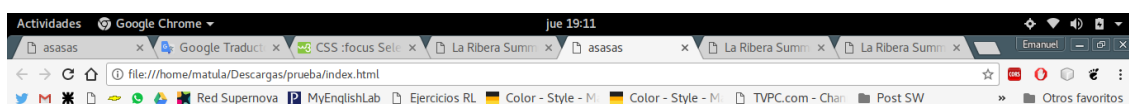
Esta propiedad especifica la posición de la imagen inicial en el fondo, es decir, la primera imagen que se dibuja y a partir de la cual hacia la derecha y para abajo pueden dibujarse otras.

Valores	<ul style="list-style-type: none">• pixeles• Porcentaje.• left• center• right• top• bottom• inherit
Valor por defecto	0% 0% (left top)
Aplica para	Todos los elementos
Se hereda	no
A tener en cuenta	Esta propiedad toma dos valores separados por un espacio en blanco, el primero indica la posicion horizontal de inicio y el segundo la vertical.

Palabras claves:

- left, center, top, right, bottom sitúan la imagen en el borde del padding del elemento.
- Si solo se especifica el valor horizontal, el vertical es tomado como center. Por ejemplo background-position:right; es lo mismo que background-position:right center;

```
body {  
    background-image: url(../imagenes/boca.jpg);  
    background-repeat: no-repeat;  
    background-position: center center;  
}
```



Boca Juniors

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

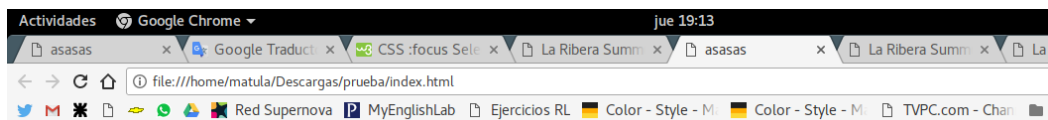
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Píxeles:

- Especifican la posición mediante la distancia desde la esquina superior izquierda, usando píxeles.

```
body {  
  background-image: url(../imagenes/boca.jpg);  
  background-repeat: no-repeat;  
  background-position: 300px 200px;  
}
```



Boca Juniors

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrum exercitationem ullam veniam, quis nostrum ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrum exercitationem ullam veniam, quis nostrum ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrum exercitationem ullam veniam, quis nostrum ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

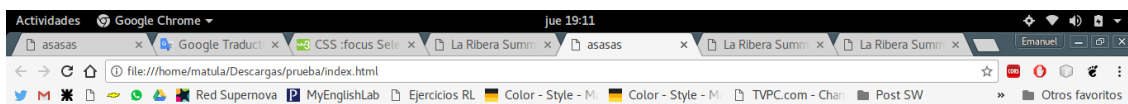
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrum exercitationem ullam veniam, quis nostrum ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.



Porcentaje:

- Es similar a las unidades de medida pixeles, de acuerdo a los porcentajes especificados es lo que la imagen se desplaza con respecto a la esquina superior izquierda.
- Si se omite el segundo valor este toma por defecto 50% (centrado).

```
body {  
    background-image: url(../imagenes/boca.jpg);  
    background-repeat: no-repeat;  
    background-position: 50% 50%;  
}
```



Boca Juniors

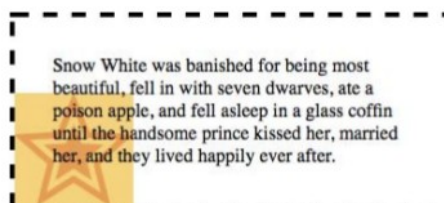
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

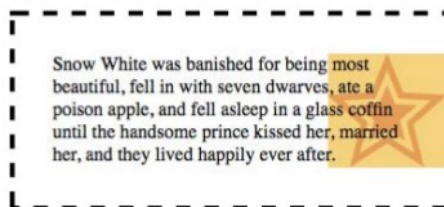
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

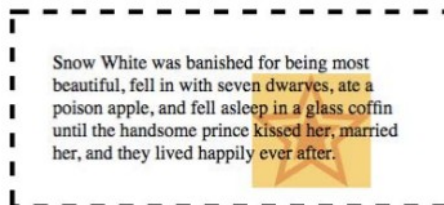
Ejemplos



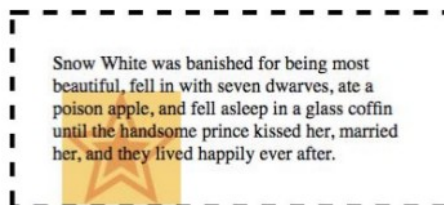
background-position: left bottom;



background-position: right center;



background-position: 200px 50px;



background-position: 15% 100%;

Ejercicio 9-4

1. Hacer que la imagen usada en el header `purpledot.png` se ubique centrada horizontalmente y en el top verticalmente.
2. Borrar los valores de las propiedades `background-imagen` y `background-position` del `body`.
3. Hacer que el `body` tenga como imagen de fondo `blackgoose.png`, que la imagen no se repita y este centrada horizontalmente y verticalmente.

Propiedad background-attachment

Si en el ejercicio 9-4 scroleamos la página vemos que la imagen desaparece en un momento, es decir, scrollea junto con la página. La propiedad background-attachment permite especificar como la imagen va a permanecer con respecto al contenido de la página.

Valores	<ul style="list-style-type: none">• scroll• fixed.• local• inherit
Valor por defecto	scroll
Aplica para	Todos los elementos
Se hereda	no

Con el valor **scroll** la imagen scrollea junto con la página:

```
background-attachment: scroll;
```

Con el valor **fixed** la imagen se queda en una posición fija pero teniendo como límite el final del elemento que la contiene:

```
background-attachment: fixed;
```

Ver ejemplos carpeta ejemplo12

Ejercicio 9-5

1. Hacer que la imagen de fondo del body tenga background-attachment con valor fixed.
2. Agregar, probar funcionamiento y luego borrar: hacer que la imagen del header también tenga fixed como attachment.