

LMSGI

7. Añadiendo estilo con css

Llegó la magia

¿Recuerdas al Mago de Oz?
Bien, esta es la parte del libro
que todo pasará de B/N a color

- ¿Sabes todo lo que necesitas para crear páginas así?
- Tan sólo...
- ... aprender CSS.
- Empezamos...



La casa ideal

Le daremos diseño a este lugar...

```
dormitorio {  
  cortinas: crudo;  
  alfombra: lana roja;  
}
```

... y el baño necesita una gran ayuda!

```
baño {  
  azulejo: crudo;  
  cortinas: rosa;  
}
```



Sintaxis del CSS

- Cada **sentencia CSS** consiste en:
 - Un **lugar**: como el dormitorio
 - La **propiedad** de ese lugar: cortinas, alfombra
 - Un **estilo** a aplicar a la propiedad: color azul o rojo
- HTML tiene elementos y CSS:
 - Un **elemento**: <p>, por ejemplo
 - La **propiedad** del elemento: color de fondo
 - El **estilo** a aplicar: rojo

Usando CSS con html

Lo primero que hay que poner es el elemento, pero en CSS no ponemos <> alrededor

p {

}

Se colocan todos los estilos para el elemento <p> dentro de los { y }

Después, se indica la propiedad del elemento a la que le daremos estilo, en este caso, el color de fondo

background-color: red;

Fíjate que aparecen dos puntos (:) entre la propiedad y su estilo

Seguidamente, se indica el estilo para esa propiedad

Y finalmente, ponemos un punto y coma

A todo este conjunto entero le llamamos REGLA

Agregar más estilo

- Podemos añadir ***tantas propiedades*** y valores en cada regla, ***como*** queramos o ***necesitemos***.
- Si queremos ***añadir un borde*** alrededor, podríamos añadirlo a la regla de esta forma:

```
p {
```

```
background-color: red;
```

```
border: 1px solid gray;
```

```
}
```

Añadimos otra propiedad
y su correspondiente valor

El elemento <p> tendrá
un borde...

... que tiene un píxel de espesor, es sólido y gris

No hay preguntas tontas

- ¿Cada elemento `<p>` tiene el mismo estilo o pueden dos párrafos tener colores diferentes?
- ¿Cómo sabemos cuáles son las propiedades de un elemento?
 - <http://es.html.net/tutorials/css/>
 - <http://www.w3schools.com/css/>
- ¿Recordamos las ventajas del lenguaje?

Incluir el CSS dentro de HTML

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Head First Lounge</title>

    <style>
    </style>

  </head>
  <body>
    <h1>Welcome to the Head First Lounge</h1>
    <p>
      
    </p>
```

← Ésta no es la única forma de dar estilos a una página web. Más tarde, veremos otra forma.

← utilizamos el elemento <style>

← ... dentro del elemento <head> de la página web

← Le asignamos "text/css" a la propiedad type

← Y dentro se colocan las reglas CSS

Añadiendo estilo

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Head First Lounge</title>
    <style>
```

Esta es la regla que va a especificar el color de la fuente de los párrafos.

Únicamente seleccionamos el elemento <p> para aplicarle este estilo

```
    </style>
  </head>
```

```
p {
  color: maroon;
}
```

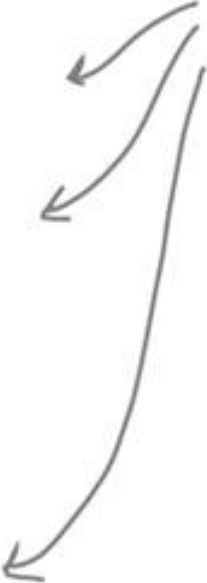
La propiedad para cambiar el color de la fuente se llama "color" (quizá podrías pensar que debería llamarse "font-color" o "text-color", pero no es así).

Establecemos el texto a un bonito color marrón que va a juego con los sofás de la sala.

Añadiendo estilo

```
<body>
  <h1>Welcome to the Head First Lounge</h1>
  <p>
    
  </p>
  <p>
    Join us any evening for refreshing
    <a href="beverages/elixir.html">elixirs</a>,
    conversation and maybe a game or two of
    <em>Dance Dance Revolution</em>.
    Wireless access is always provided;
    BYOWS (Bring your own web server).
  </p>
  <h2>Directions</h2>
  <p>
    You'll find us right in the center of downtown
    Webville. If you need help finding us, check out our
    <a href="about/directions.html">detailed directions</a>.
    Come join us!
  </p>
</body>
</html>
```

El selector `p`
selecciona todos
los párrafos en el
HTML



Añadiendo estilo



Estilos para los encabezados

```
h1 {  
  font-family: sans-serif;  
  color:      gray;  
}
```

Esta regla cambia el tipo de letra
y el color de la fuente para el
encabezado <h1>

```
h2 {  
  font-family: sans-serif;  
  color:      gray;  
}
```

Y esta regla hace exactamente
lo mismo con el encabezado <h2>

```
p {  
  color: maroon;  
}
```

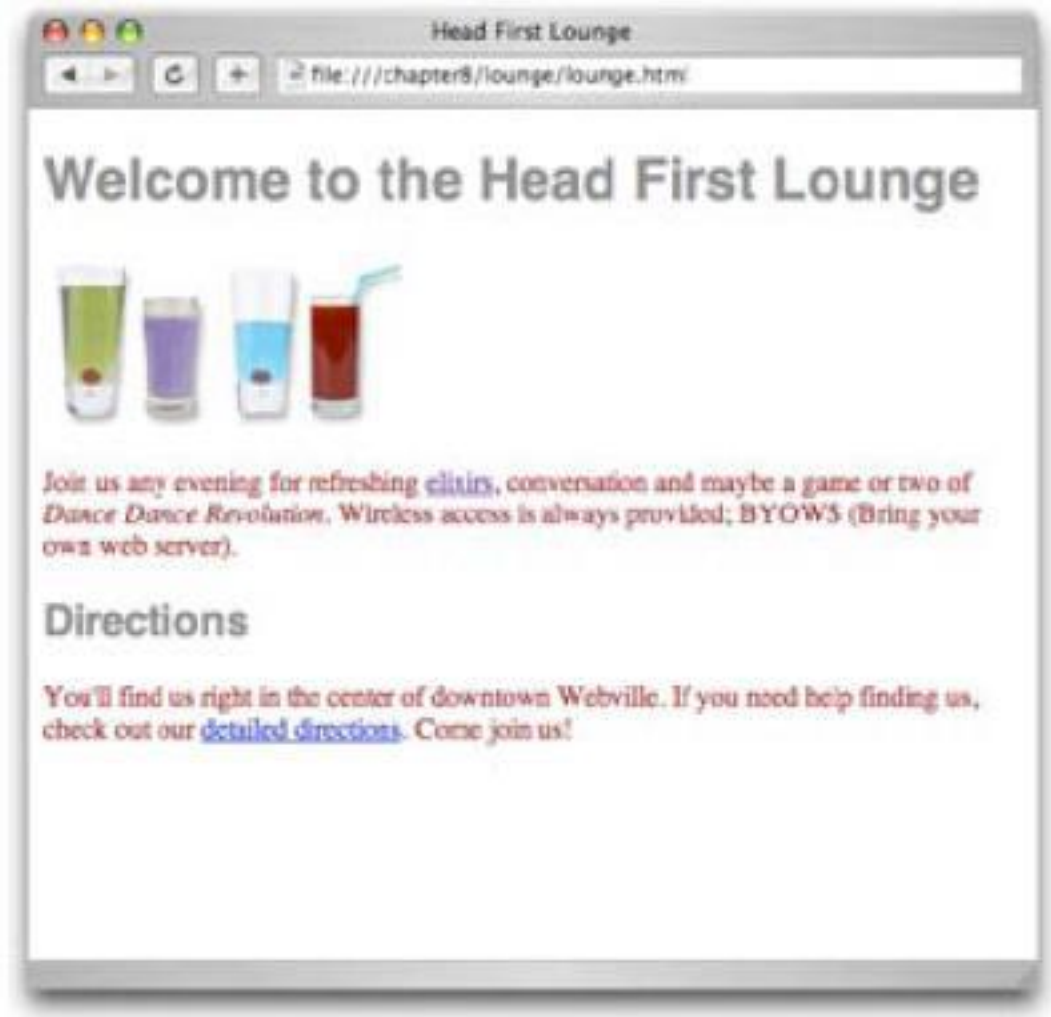
```
h1, h2 {  
  font-family: sans-serif;  
  color:      gray;  
}
```

Para escribir una regla que afecte
a más de un elemento, ponemos
los elementos separados por comas
como h1,h2

```
p {  
  color: maroon;  
}
```

Lo probamos

Aquí tenemos los
cambios para los dos
encabezados



Una línea bajo el encabezado

border-bottom: 1px solid black;

Esta propiedad controla cómo aparece el borde inferior del elemento

Le daremos un píxel de espesor y será una línea sólida negra

```
h1, h2 {  
  font-family: sans-serif;  
  color: gray;  
  border-bottom: 1px solid black;  
}
```

Aquí estamos añadiendo la propiedad del borde inferior a los elementos h1 y h2

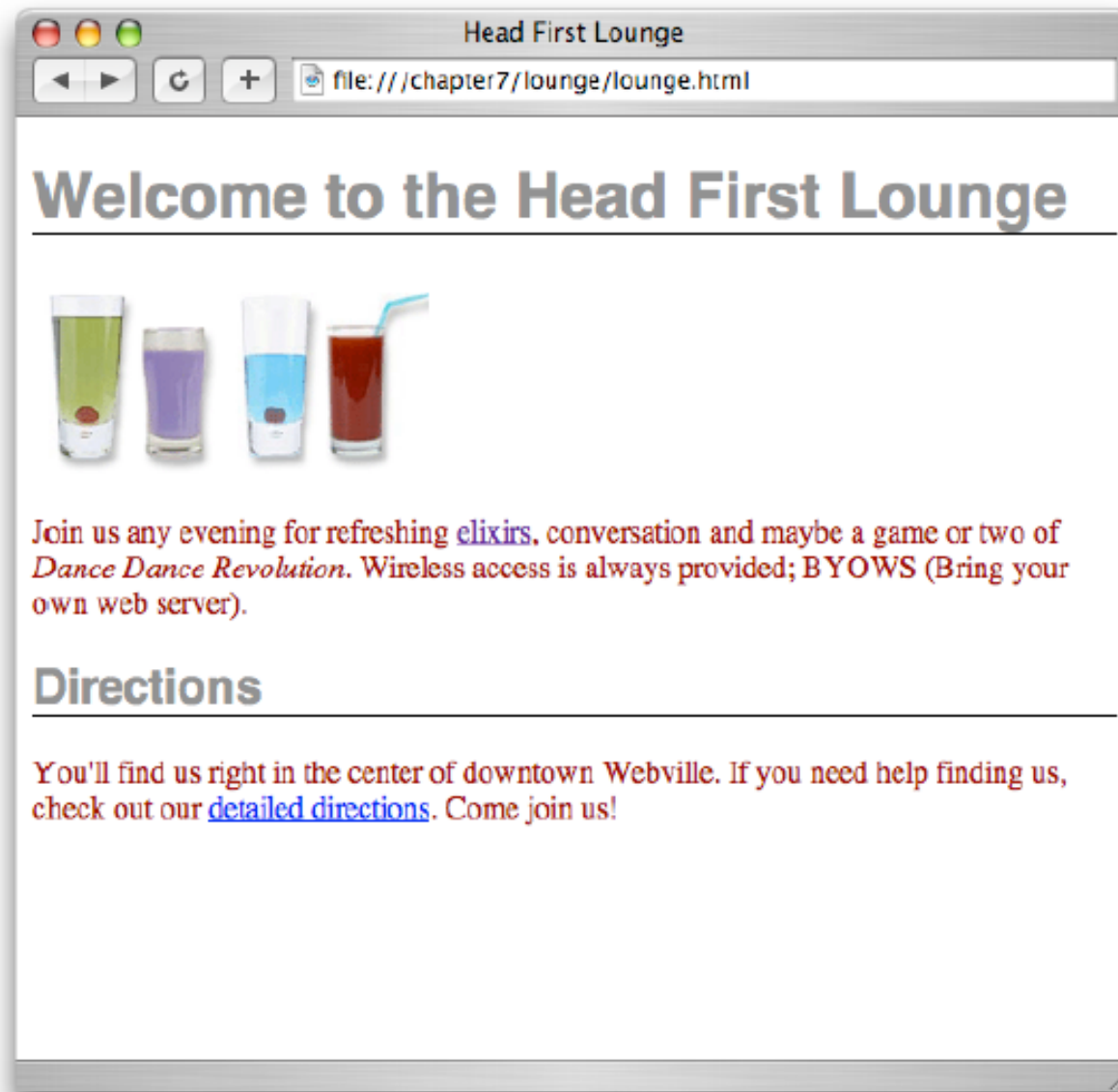
```
p {  
  color: maroon;  
}
```

Si hacemos esto ...

... obtendremos bordes en los 2 encabezados, y no es lo que queremos.



Una línea bajo el encabezado



Una nueva regla, sólo para <h1>

```
h1, h2 {  
  font-family: sans-serif;  
  color: gray;  
}
```

La primera regla se queda igual, será una regla combinada para cambiar la fuente de <h1> y <h2>

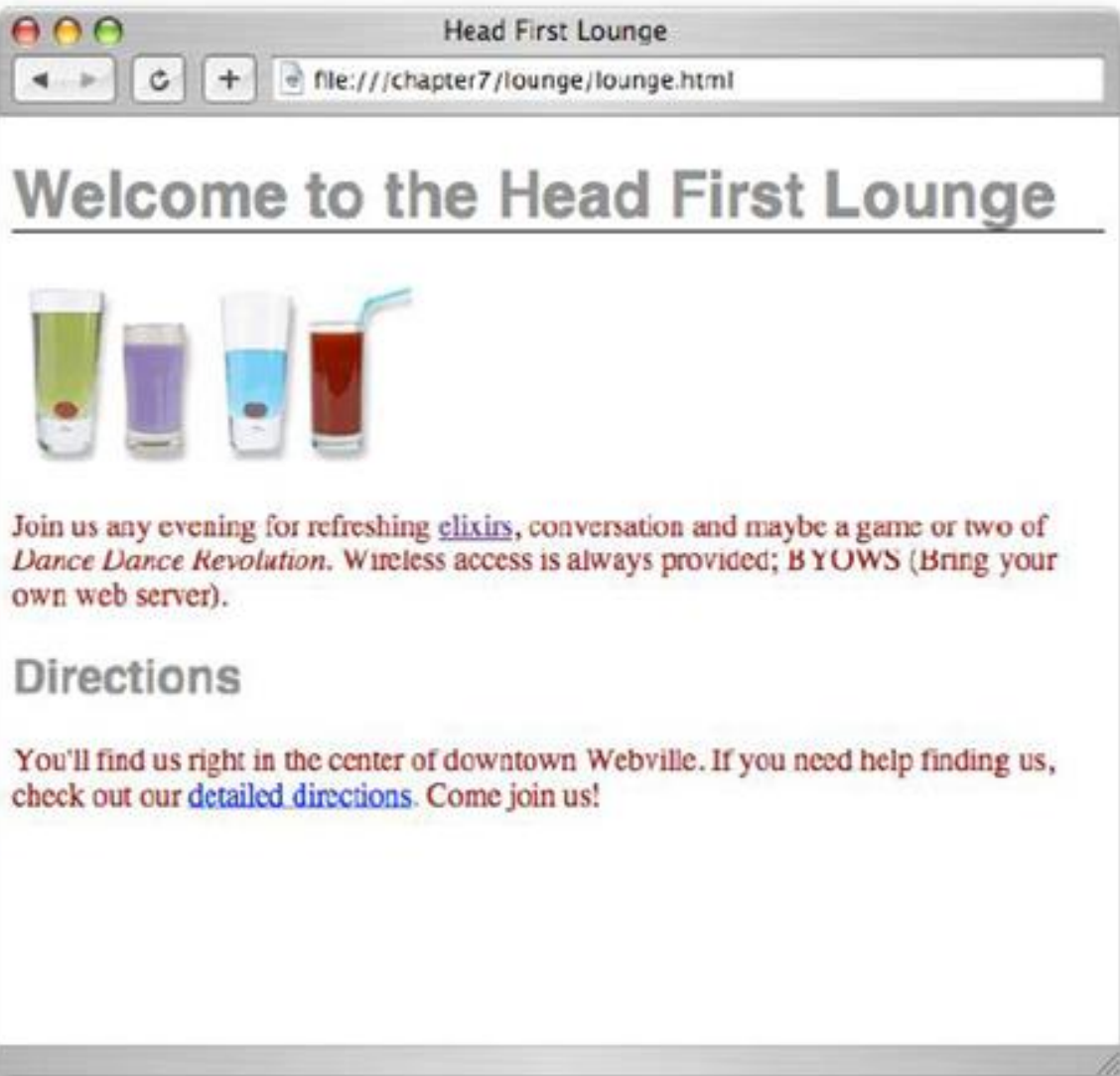
```
h1 {  
  border-bottom: 1px solid black;  
}
```

Pero añadimos una segunda regla para añadir otra propiedad solamente a <h1>

```
p {  
  color: maroon;  
}
```


Una nueva regla, sólo para <h1>

Aquí está el borde inferior en negro



Y aquí sin borde, justo lo que queríamos



No hay preguntas tontas

- ¿Cómo funciona exactamente cuando tenemos más de una regla para un elemento?
- ¿Qué ventajas tiene este enfoque? ¿No sería mejor organizar cada elemento por separado, y así sabemos exactamente los estilos que tenemos?
- ¿Por qué usamos el borde inferior para subrayar texto? ¿No existe un estilo de subrayado para el texto?

Selectores css

A esto le llamamos
SELECTOR

```
h1 {  
    color: gray;  
}
```

El estilo se aplica a los elementos
descritos por el selector, en este caso
los elementos <h1>

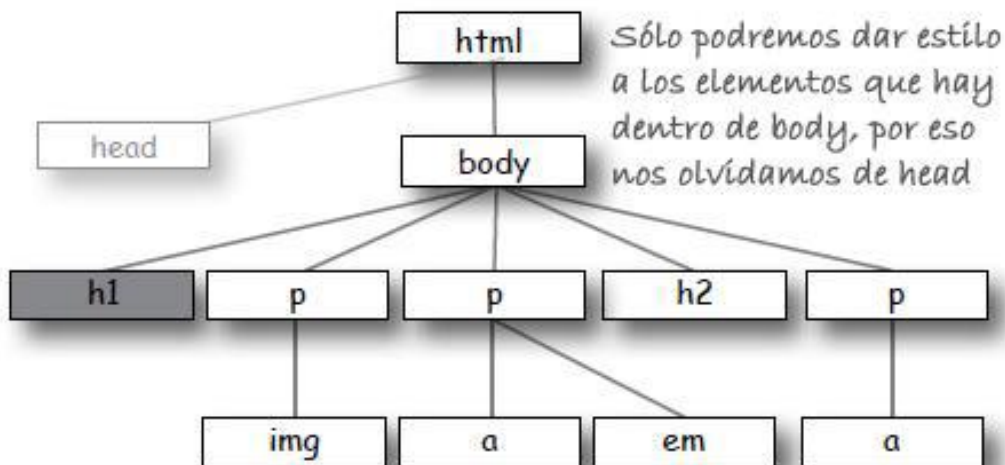
Otro selector.
El estilo se aplica a los elementos <h1> y <h2>.

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

Observamos los selectores

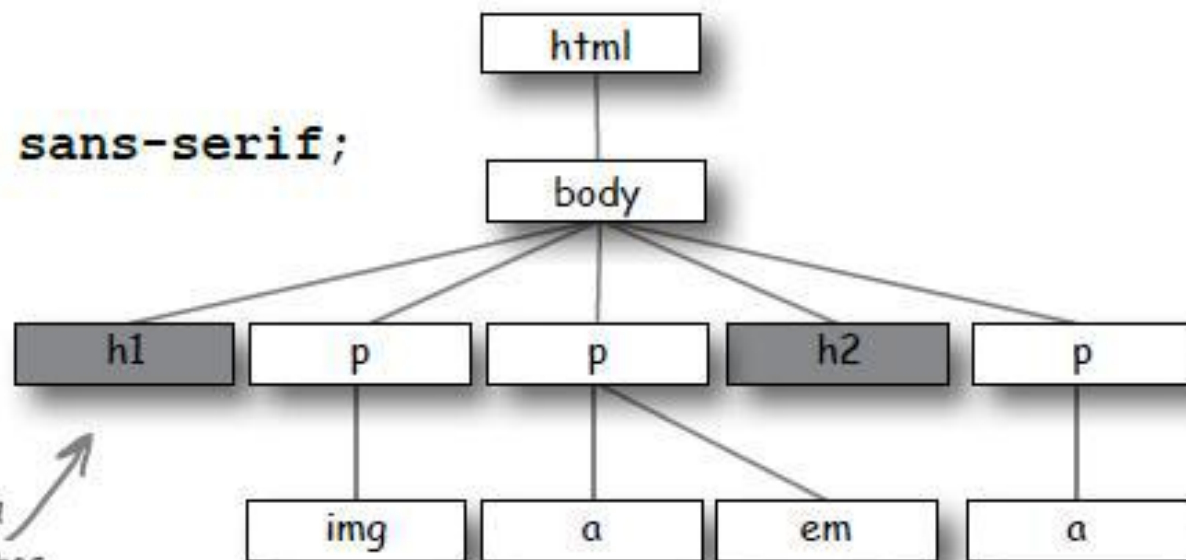
```
h1 {  
  font-family: sans-serif;  
}
```

Este selector aplica estilo a todos los elementos `<h1>` de la página



Observamos los selectores

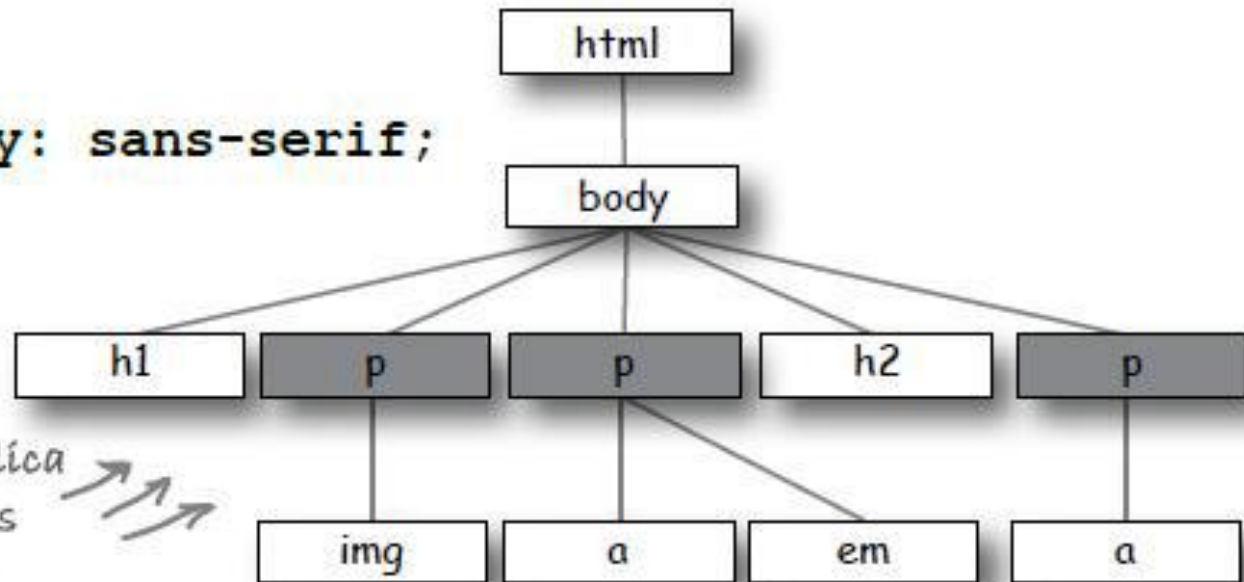
```
h1, h2 {  
  font-family: sans-serif;  
}
```



Este selector aplica
estilo a los elementos
<h1> y <h2>

Observamos los selectores

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

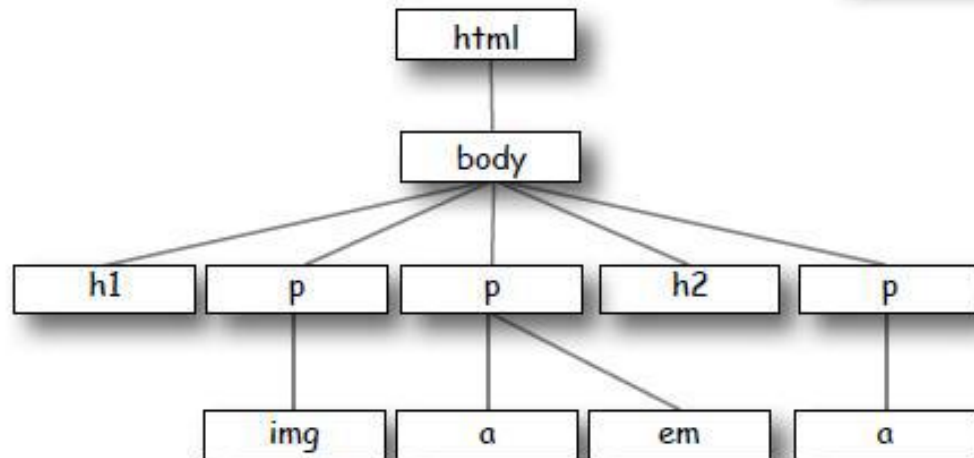


Este selector aplica
estilo a todos los
elementos <p>

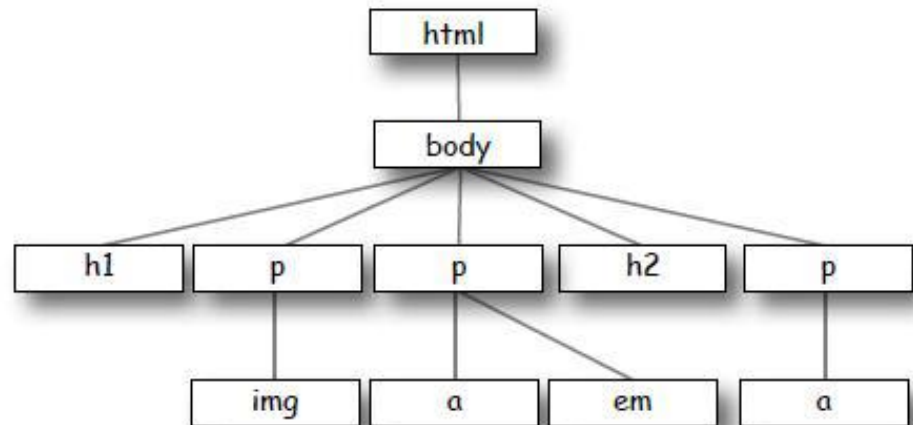


Ejercicio: Selecciona los elementos afectados

```
p, h2 {  
  font-family: sans-serif;  
}
```

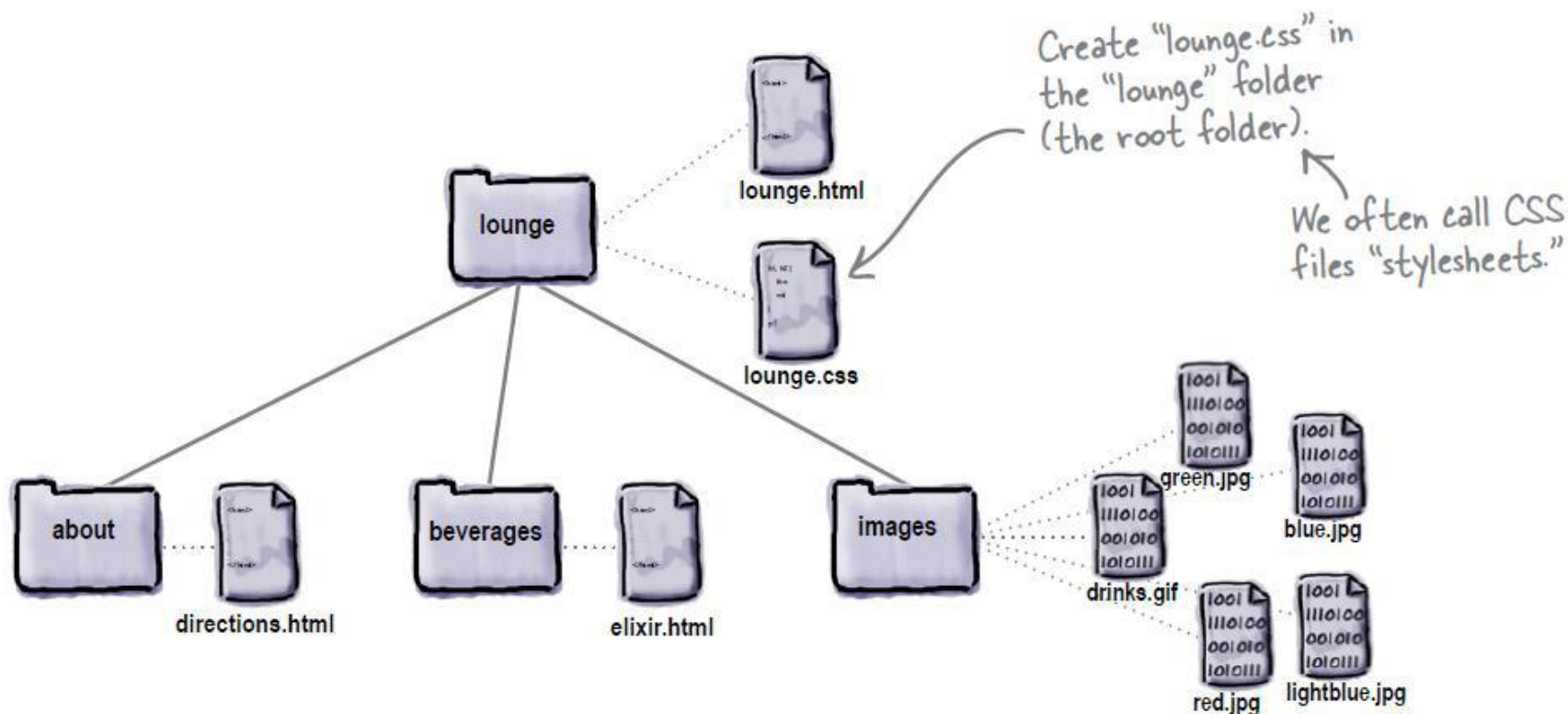


```
p, em {  
  font-family: sans-serif;  
}
```



Creando un fichero de estilos


- Vamos a crear un fichero llamado **lounge.css** para guardar todas las reglas de estilo para todas las páginas del BAR.



Creando un fichero de estilos

- Ahora copia todas las reglas al fichero nuevo, sin copiar el elemento `<style>`
- El nuevo fichero sólo contendrá CSS, no HTML
- Borra las reglas de donde estaban (`<head>`).

```
h1, h2 {  
    font-family: sans-serif;  
    color: gray;  
}  
  
h1 {  
    border-bottom: 1px solid black;  
}  
  
p {  
    color: maroon;  
}
```



Tu fichero bar.css debería contener esto. Recuerda no incluir las etiquetas `<style>`

Enlazar el fichero HTML con el CSS

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Head First Lounge</title>
    <link type="text/css" rel="stylesheet" href="lounge.css">
    <style>
    </style>
  </head>
  <body>
    <h1>Welcome to the Head First Lounge</h1>
    <p>
      
    </p>
    .
    .
    .
  </p>
</body>
</html>
```

Este es el código que enlaza con el fichero externo CSS

No necesitamos el elemento <style> nunca más. Lo borramos.

El resto del XHTML es igual

El elemento `<link>`

Usa el elemento `link` para enlazar a la información externa

El tipo de información es `text/css`, es decir, una hoja de estilo en cascada

Y la hoja de estilo se encuentra referenciada en `href`. Esto es un enlace relativo, pero podría ser una URL

```
<link type="text/css" rel="stylesheet" href="lounge.css">
```

El atributo `rel` especifica la relación entre el fichero XHTML y la hoja de estilo. Y estamos enlazando con una hoja de estilo.

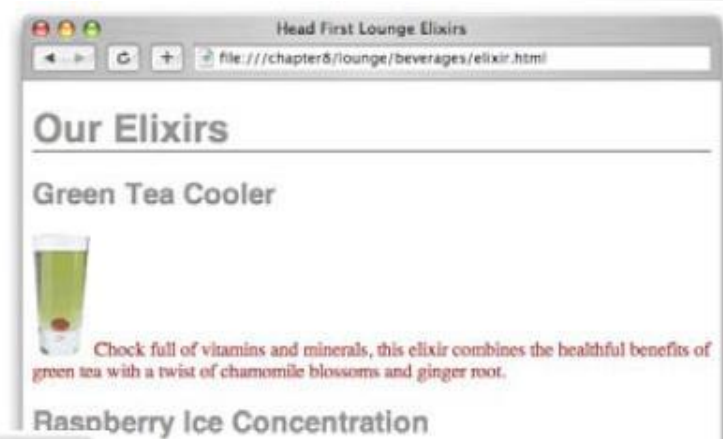
`<link>` es un elemento vacío

Ejercicio

- Enlaza todos los documentos html del proyecto lounge con el fichero lounge.css
- Pon atención a introducir correctamente las rutas relativas.

El poder del CSS

- Con un solo fichero de estilos, hemos dado estilo a todas las páginas.



ombining raspberry juice with lemon grass, citrus peel and rosehips, this icy make your mind feel clear and crisp.

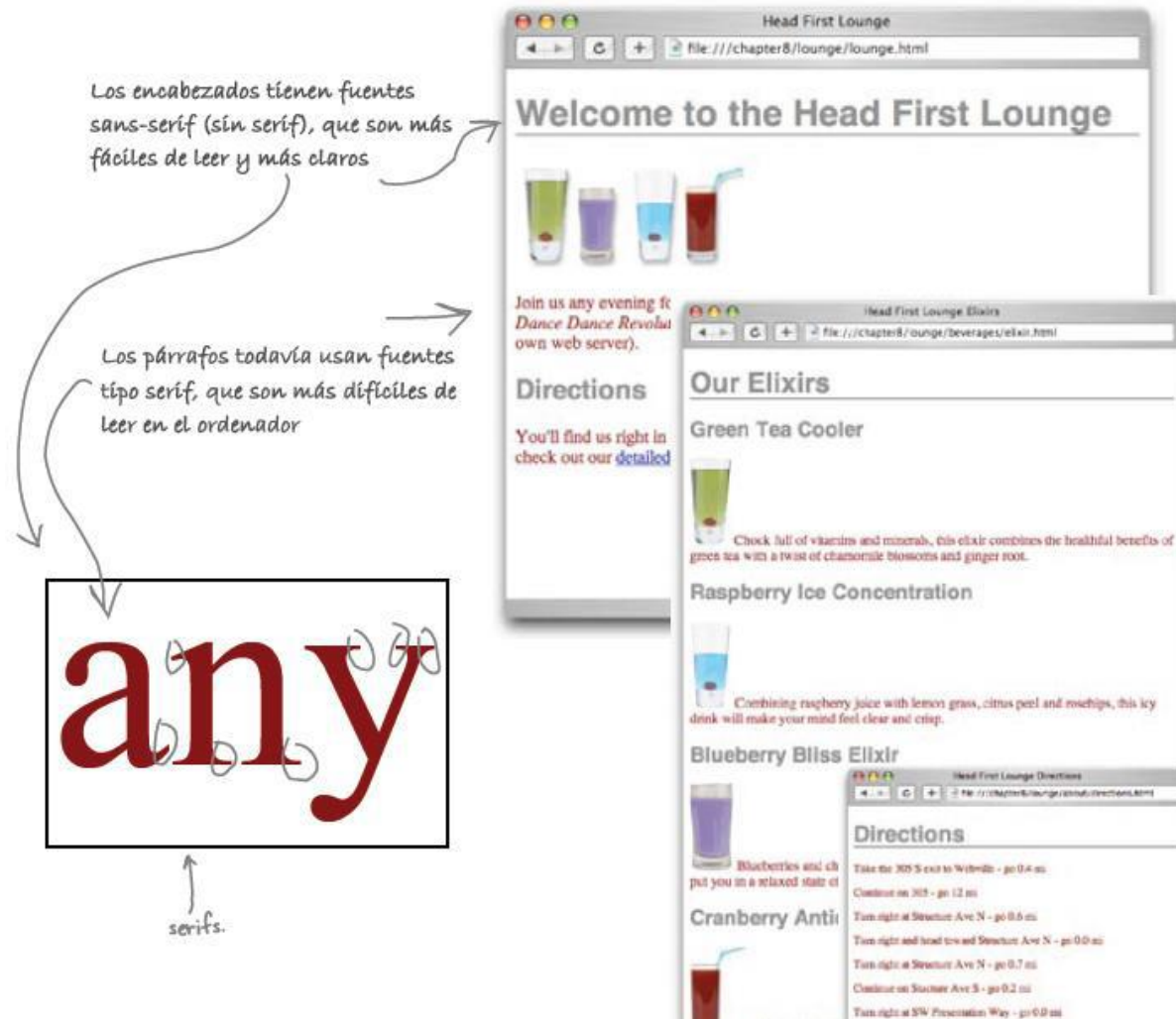
erry Bliss Elixir

lueberrie
a relaxe
erry

Wake up



Ejercicio: *Cambia todas las fuentes a sans-serif*



¿Será ésta la mejor solución?

- ¿Será lo mejor dar estilo a la fuente en cada elemento?
- Y si añadimos elementos nuevos, ¿cogerán este estilo o habrá que definirlo?
- ¿Habrá alguna forma de especificar la fuente para toda la página?

Hablemos de la HERENCIA

- ¿Te has fijado que los elementos internos al párrafo también han cambiado a sans-serif?

Cuando añadimos la propiedad de la fuente sans-serif al selector p, cambia el tipo de letra en todos los elementos p, pero también cambia el estilo de los enlaces y del texto enfatizado.

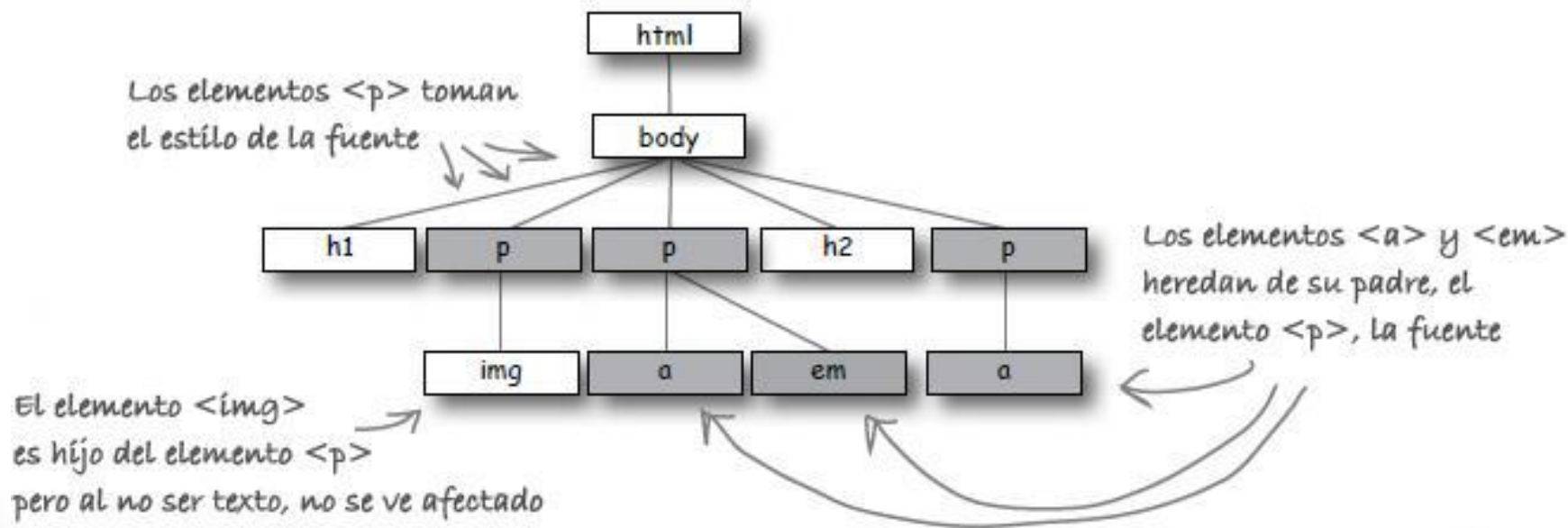


Hablemos de la HERENCIA

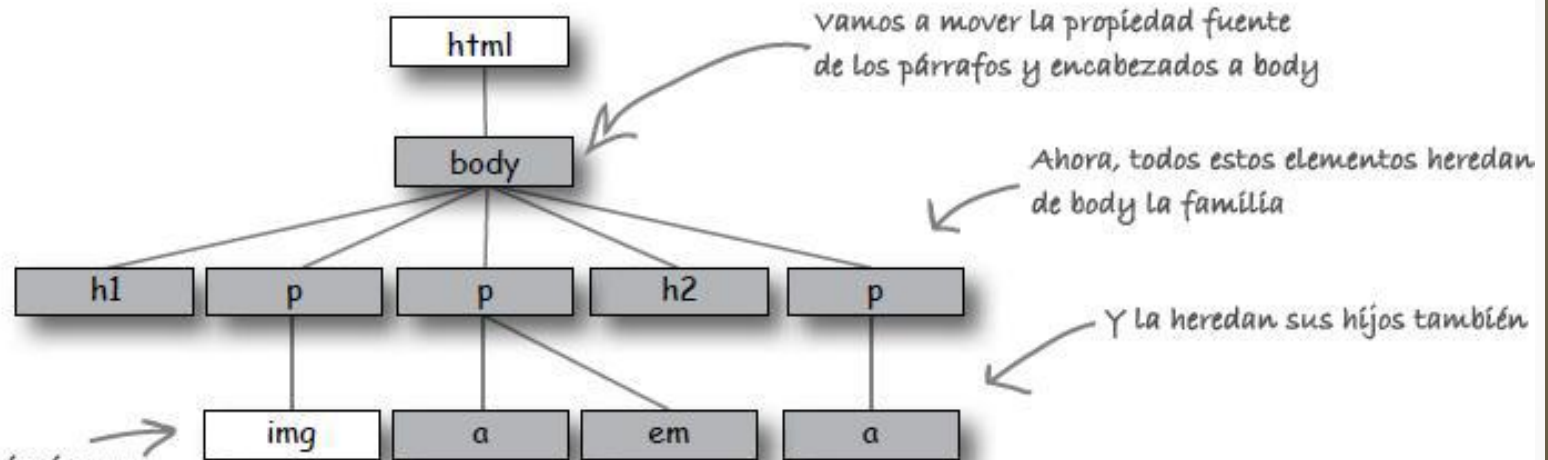
- Los elementos que están dentro del elemento <p>, heredan el estilo de la fuente de <p>
- En este caso: <a> y heredan de <p>, ya que <p> es el padre.
- Pero no todos los estilos se heredan:
 - La familia de la fuente sí se hereda para los elementos de texto
 - Los elementos que no son de texto, no heredan características de texto

¿Cómo trabaja la HERENCIA?

Si damos estilo al elemento `<p>`, aquí están todos los elementos a los que podría afectar



¿Y si movemos la propiedad hacia arriba?



Recuerda, las imágenes no tienen texto

Oh, esto es poderoso! Simplemente cambiando la propiedad de la fuente en la regla body, podríamos cambiar el tipo de fuente para un sitio entero

¿Lo probamos?

- Añade una regla con un nuevo selector: **body**
- Añade la propiedad **font-family**
- Asígnale el valor **sans-serif**
- Borra las propiedades de <h1>,<h2> y <p>

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

Primero, añade una regla que tenga como selector al elemento body. Luego, añade la propiedad font-family con el valor sans-serif.

```
h1, h2 {  
    font-family: sans-serif;  
    color: gray;  
}
```

```
h1 {  
    border-bottom: 1px solid black;  
}
```

```
p {  
    font-family: sans-serif;  
    color: maroon;  
}
```

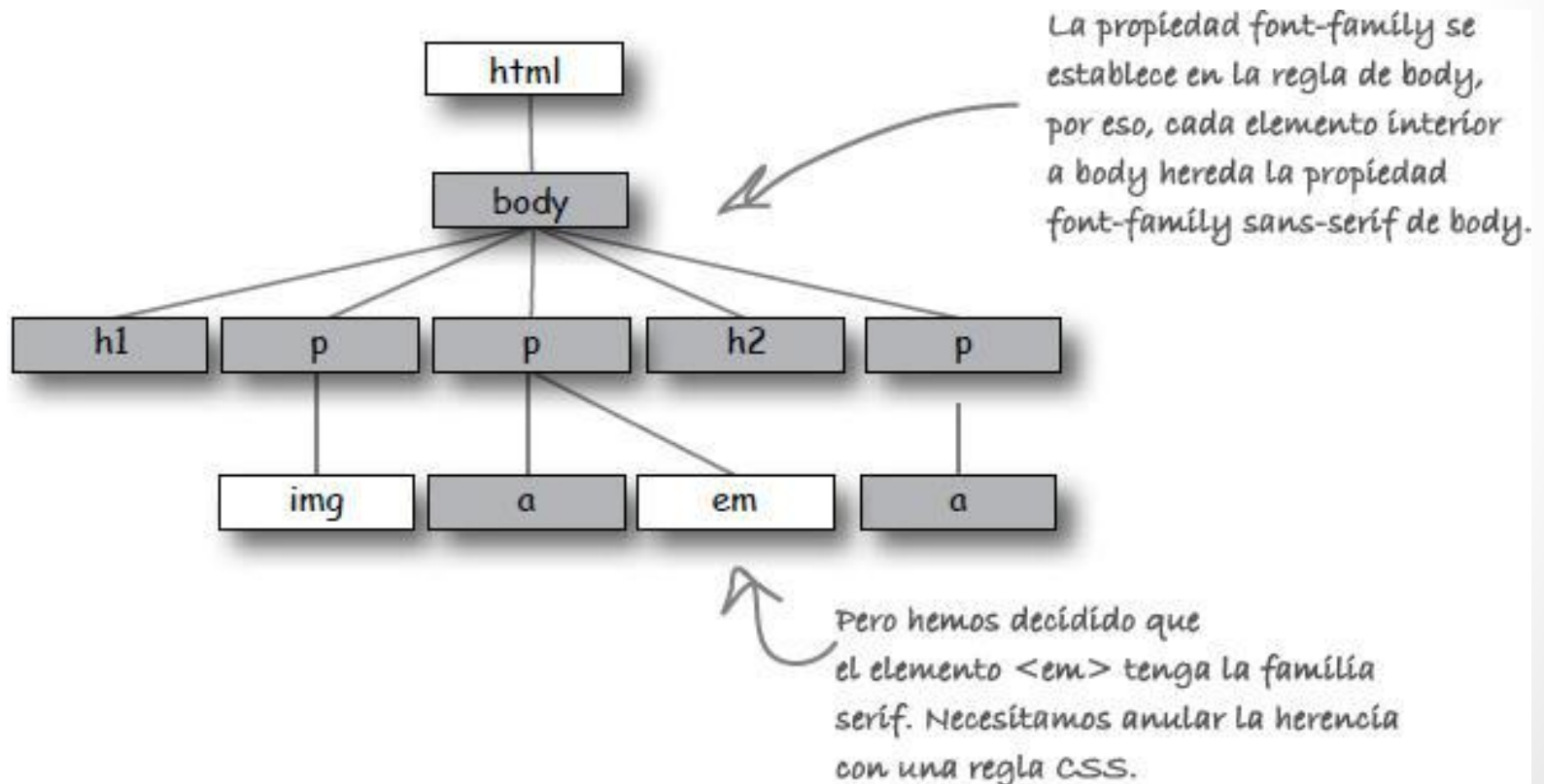
Después, eliminamos la propiedad font-family de los selectores <h1>, <h2> y <p>

¿Qué ha pasado?

- Nos damos cuenta de que hemos obtenido el mismo resultado.
- Pero, ¿era exactamente lo que esperábamos?
- Al pasar las propiedades al elemento body, la herencia lo aplica a todos sus descendientes.
- ¿Qué pasa cuando queremos que algún elemento sea diferente de esa fuente? ¿Tendremos que eliminar la propiedad del body, y asignar las propiedades por separado otra vez?

Cambiar el estilo de cualquier elemento


- Queremos cambiar la fuente de los elementos `` a una fuente serif:



Sobrescribir la herencia

```
body {  
    font-family: sans-serif;  
}  
  
h1, h2 {  
    color: gray;  
}  
  
h1 {  
    border-bottom: 1px solid black;  
}  
  
p {  
    color: maroon;  
}  
  
em {  
    font-family: serif;  
}
```

Para sobrescribir la propiedad font-family heredada de body, añade una nueva regla que seleccione los em con el valor de la propiedad font-family a serif



No hay preguntas tontas

- ¿Cómo sabe el navegador qué regla tiene que aplicar al elemento `` cuando hemos anulado la herencia?
- ¿Cómo sabemos qué propiedades CSS son heredadas y cuáles no?
- ¿Puedo siempre anular una propiedad que sea heredada si no la quiero?
- Todo esto es complicado. ¿Hay alguna forma de añadir comentarios para recordarnos la función de cada regla?

Aplicando estilos individualmente



Estoy pensando que quedaría más atractivo que el texto inferior de cada elixir fuera del color del elixir. ¿Se puede hacer?

Texto verde

Texto azul

Texto violeta

Texto rojo, aunque
no hace falta
cambiarlo...

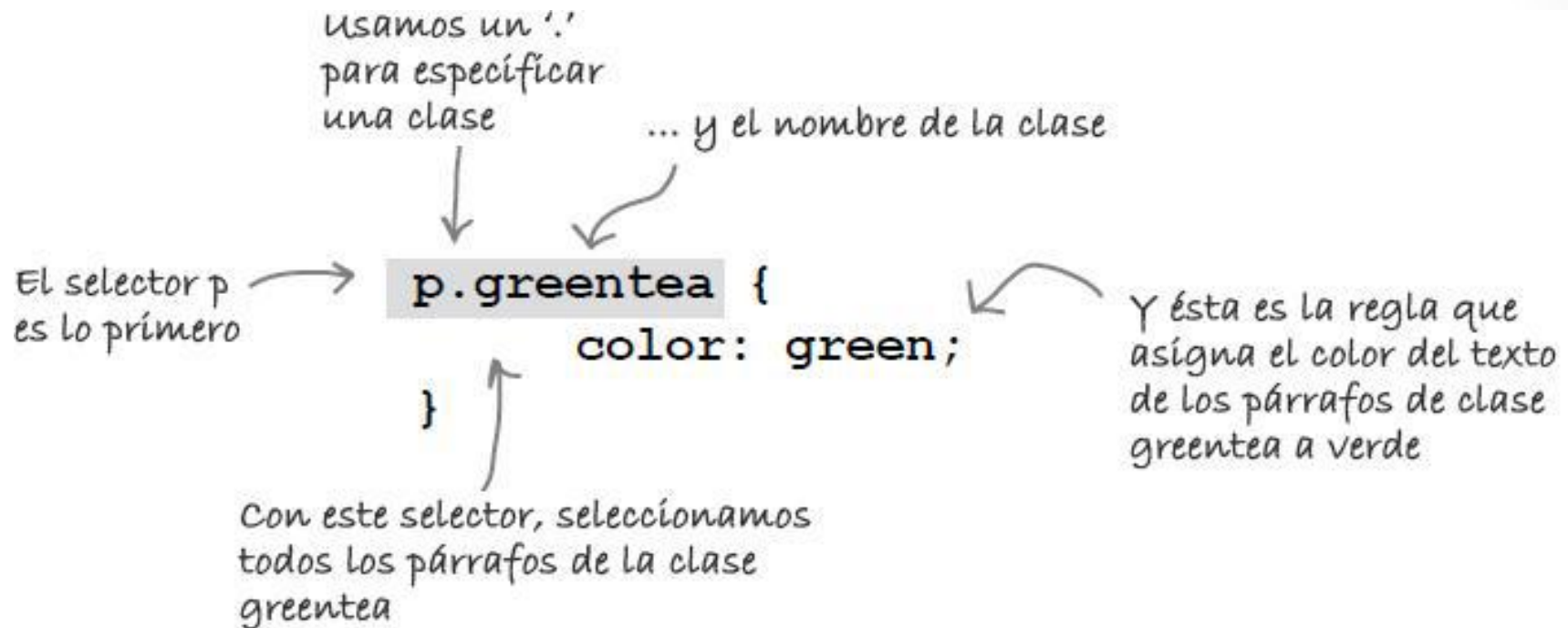


Añadiendo un elemento a una CLASE

```
<html>
  <head>
    <meta charset="utf-8">
    <title>Head First Lounge Elixirs</title>
    <link type="text/css" rel="stylesheet" href="../lounge.css">
  </head>
  <body>
    <h1>Our Elixirs</h1>
    <h2>Green Tea Cooler</h2>
    <p class="greentea">
      
      Chock full of vitamins and minerals, this elixir
      combines the healthful benefits of green tea with
      a twist of chamomile blossoms and ginger root.
    </p>
    <h2>Raspberry Ice Concentration</h2>
    <p>
      
      Combining raspberry juice with lemon grass,
      citrus peel and rosehips, this icy drink
      will make your mind feel clear and crisp.
    </p>
```

Para añadir un elemento a una clase, añádele el atributo "class" con el nombre de la clase

Crear un *SELECTOR* para la *CLASE*



*Crear un **SELECTOR** para la **CLASE***

```
body {  
    font-family: sans-serif;  
}  
  
h1, h2 {  
    color: gray;  
}  
  
h1 {  
    border-bottom: 1px solid black;  
}  
  
p {  
    color: maroon;  
}  
  
p.greentea {  
    color: green;  
}
```

Ejercicio

- Añade una nueva clase para cada elixir y aplícasela a los párrafos correspondientes.

Definiendo otro selector

- ¿Y si ahora quisiéramos asignar el mismo estilo a otro elemento como <blockquote>?

```
blockquote.greentea, p.greentea {  
    color: green;  
}
```

Añadimos otro selector para
añadir el mismo estilo a <blockquote>

- Para definir un blockquote lo haremos así:
 - <blockquote class="greentea">

Definiendo otro selector

- ¿Y si ahora quisiéramos añadir el mismo estilo a los elementos `<h1>`, `<h2>` y `<h3>`? ¿Escribiríamos un selector ENORME?

```
.greentea {
```

```
    color: green;
```



Si no ponemos los nombres a los elementos y ponemos un punto (.), la regla se aplicará a todos los miembros de la clase (definidos en el XHTML como greentea)

- Seguimos definiendo los elementos así:
 - `<elemento class="greentea">`

¿Podría un elemento pertenecer a varias clases?

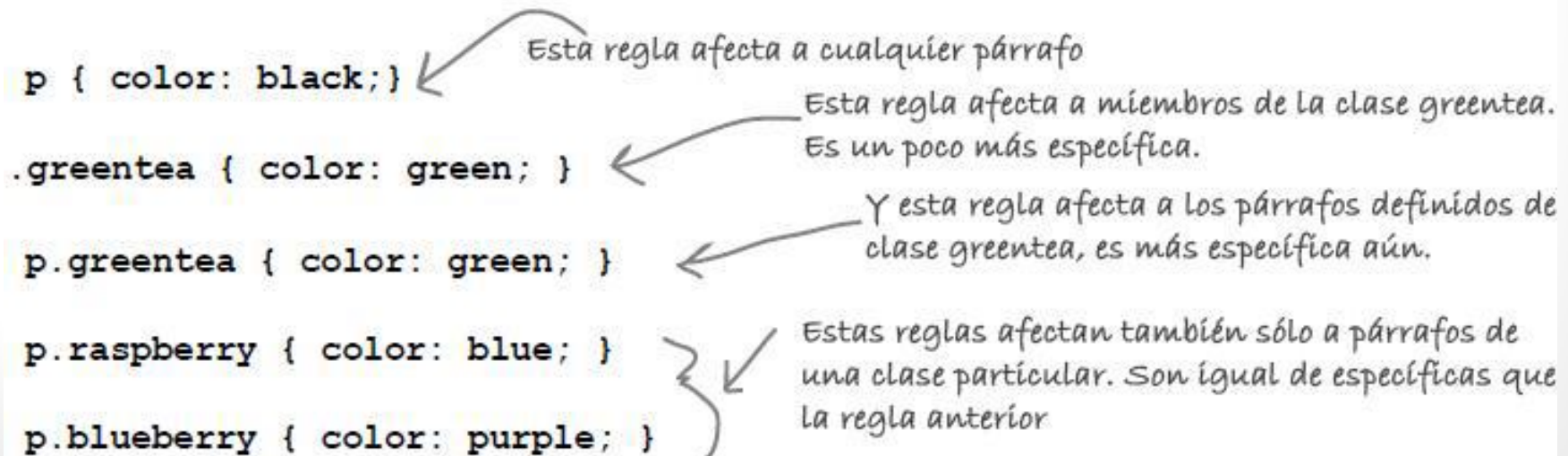
- Sí que podría, tan sólo tendríamos que definirlo en todas las clases:
 - `<p class="greentea bluetea redtea">`
- Todas las clases separadas por espacios, sin orden
- ¿Qué estilo quedaría si todas modifican el color?
- Tendría más sentido si cada clase modificara unas propiedades diferentes:
 - `<h1 class="productos especial">`
- donde *productos* define la *font_size* y el *weight* y *especial* define el *color* del texto

Reglas de oro -Aplicación de Estilos

1. Se quiere observar qué valor tiene una propiedad para un elemento.
2. Observar si ...¿existe algún selector para modificar la propiedad del elemento?
 - **Si:** el valor asignado será su valor
 - **No:** Confiamos en la Herencia
3. **Herencia:** miramos los padres y los padres de los padres, ¿tienen esa propiedad?
4. Si no lo encontramos, tendrá el valor por defecto definido en el navegador
5. Y si hay múltiples selectores, ¿qué ocurre?

Múltiples selectores -conflicto

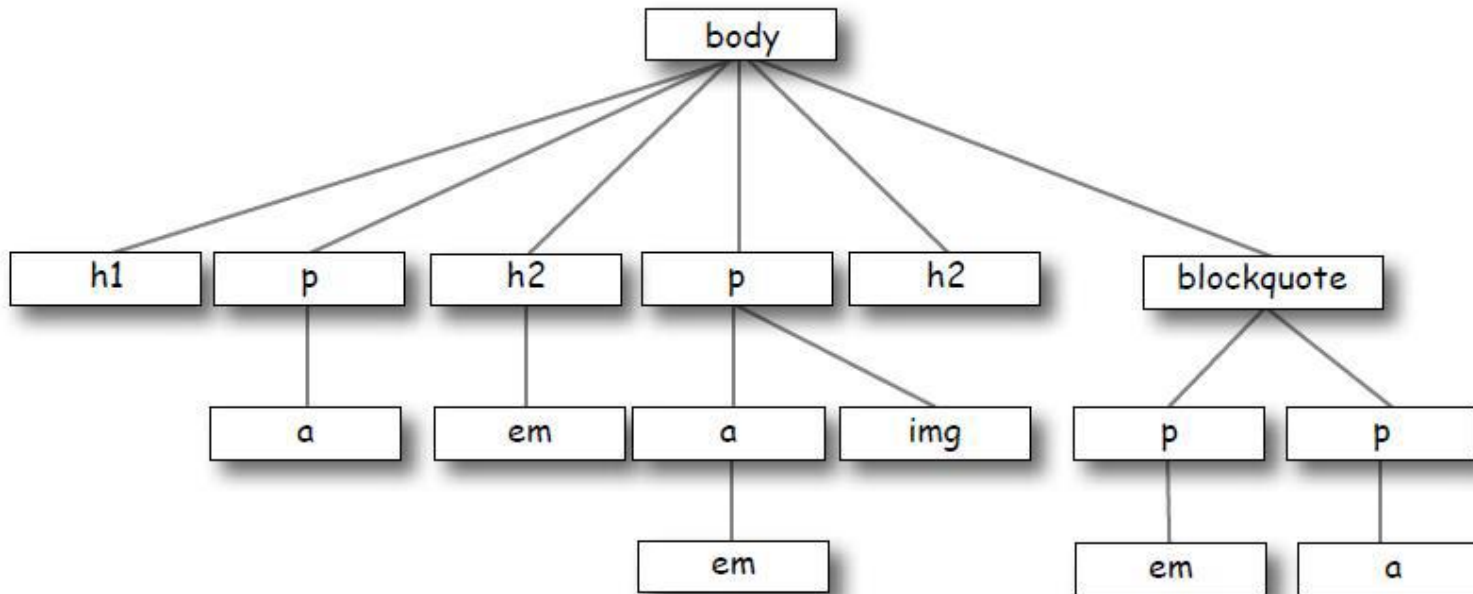
- Si hay múltiples selectores modificando la misma propiedad, ¿cuál gana? ¿quién desempata?
 - Gana el más específico
- Pero, ¿qué significa ser más específico?
 - Veremos un ejemplo para entenderlo:



Múltiples selectores

- En el fichero index.html cambia la definición del párrafo de clase greentea, para incluir todas las clases del resto de párrafos:
 - `<p class ="greentea raspberry blueberry">`
- Salva y actualiza. ¿Qué color tiene ahora el párrafo definido así?
- Ahora, reordena las clases de tu HTML, poniendo greentea al final:
 - `<p class ="rasberry blueberry greentea">`
- Salva y actualiza. ¿Qué color tiene ahora ese párrafo?
- Ahora, abre tu fichero CSS y mueve la regla **p.greentea** al final del fichero.
- Salva y actualiza. ¿Qué color tiene ahora ese párrafo?
- Finalmente, mueve la segunda regla (**rasperry**) al final del fichero.
- Salva y actualiza. ¿Qué color tiene ahora ese párrafo?
- Después de haber acabado, redefine el párrafo verde para que pertenezca a una sola regla de nuevo:
 - `<p class ="greentea">`
- Salva y actualiza. ¿Qué color tiene ahora ese párrafo?

¿Cómo se aplica?



```
body {  
    color: green;  
}  
  
p {  
    color: black;  
}
```

←
Este es el CSS. úsalo para
determinar qué elementos
se adaptan a cada color.

Encuentra los errores

```
<style>

body {
    background-color: white

h1, {
    gray;
    font-family: sans-serif;
}

h2, p {
    color:

<em> {
    font-style: italic;
}

</style>
```

Cómo validar archivos CSS

- <http://jigsaw.w3.org/css-validator/>
- Con este validador podremos ver si nuestro código CSS es correcto o tiene errores.

Deutsch English Español Français 한국어 Italiano Nederlands 日本語 Polski Português Русский فارسی Svenska Български Українська Čeština Romanian 简体中文

W3C CSS Validation Service
Verifica Hojas de Estilo en Cascada (CSS) y documentos (X)HTML con hojas de estilo

mediante URI mediante Carga de Archivo mediante Entrada directa


Validar mediante URI


Introduce la URI de un documento (HTML con CSS o sólo CSS) que desees validar :

Dirección :

► Más opciones

Check

 The W3C CSS validator is developed with assistance from the Mozilla Foundation, and supported by community donations. [Donate](#) and help us build better tools for a better web.

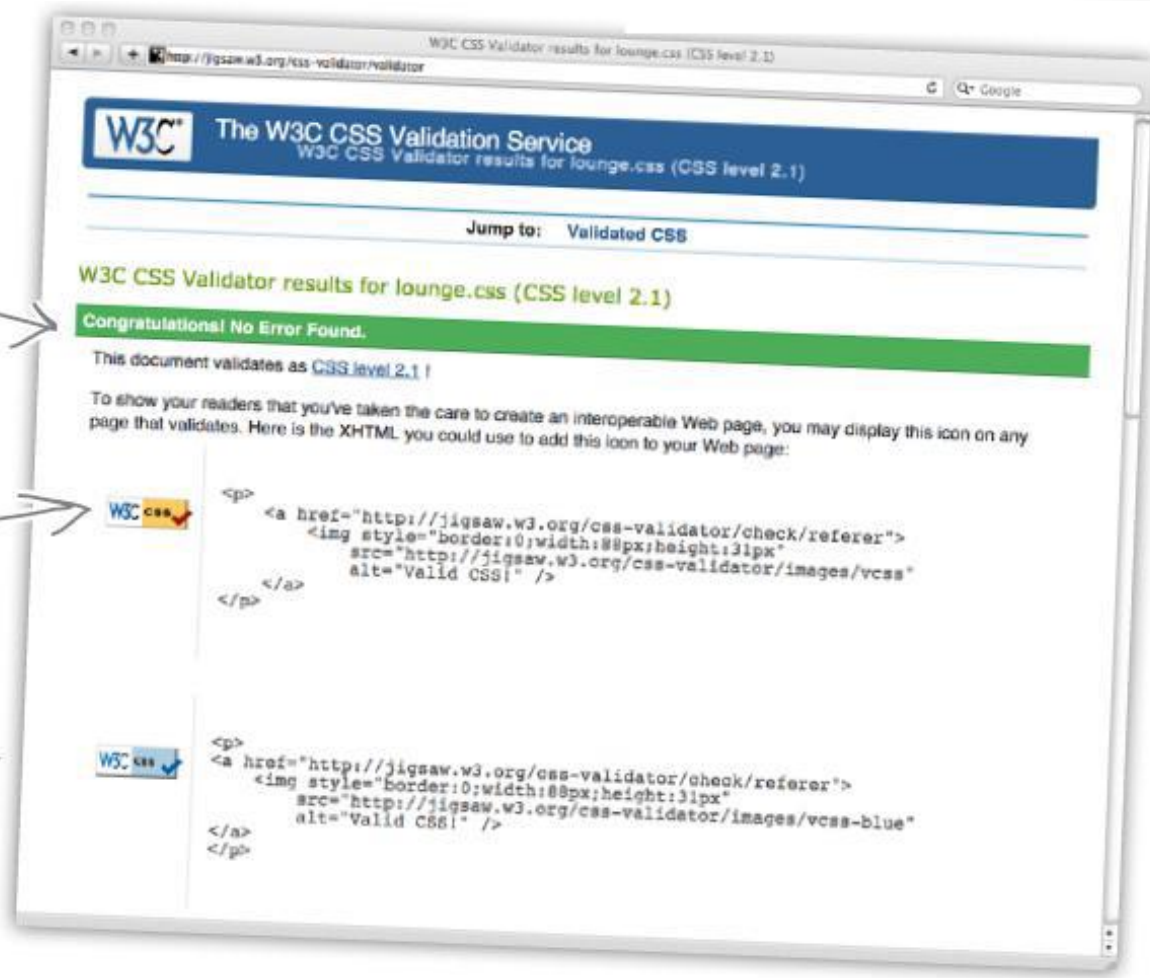
 773

Nota: Si desees validar tu hoja de estilo CSS incrustada en un documento (X)HTML, deberías antes [comprobar que el \(X\)HTML utilizado es válido](#).

Validador css

Yay! Our CSS validates as CSS 2.1 (the validator hasn't upgraded to CSS 3 yet, but if it has by the time you read this, it should still validate).

Here are some icons you can put on your web page if you want to show off that your CSS validates. (You can get similar icons for validated HTML, too.)



Sopa de propiedades

Use color to set the font color of text elements.

color

This property controls the weight of text. Use it to make text bold.

font-weight

This is how you tell an element how to position its left side.

left

This property sets the space between lines in a text element.

line-height

font-size

Makes text bigger or smaller.

margin

If you need space between the edge of an element and its content, use margin.

border

This property puts a border around an element. You can have a solid border, a ridged border, a dotted border...

background-color

This property controls the background color of an element.

letter-spacing

This lets you set the spacing between letters. Like this.

text-align

Use this property to align your text to the left, center, or right.

top

Controls the position of the top of the element.

Use this property for italic or oblique text.

font-style

This property lets you change how list items look in a list.

list-style

Use this property to put an image behind an element.

background-image

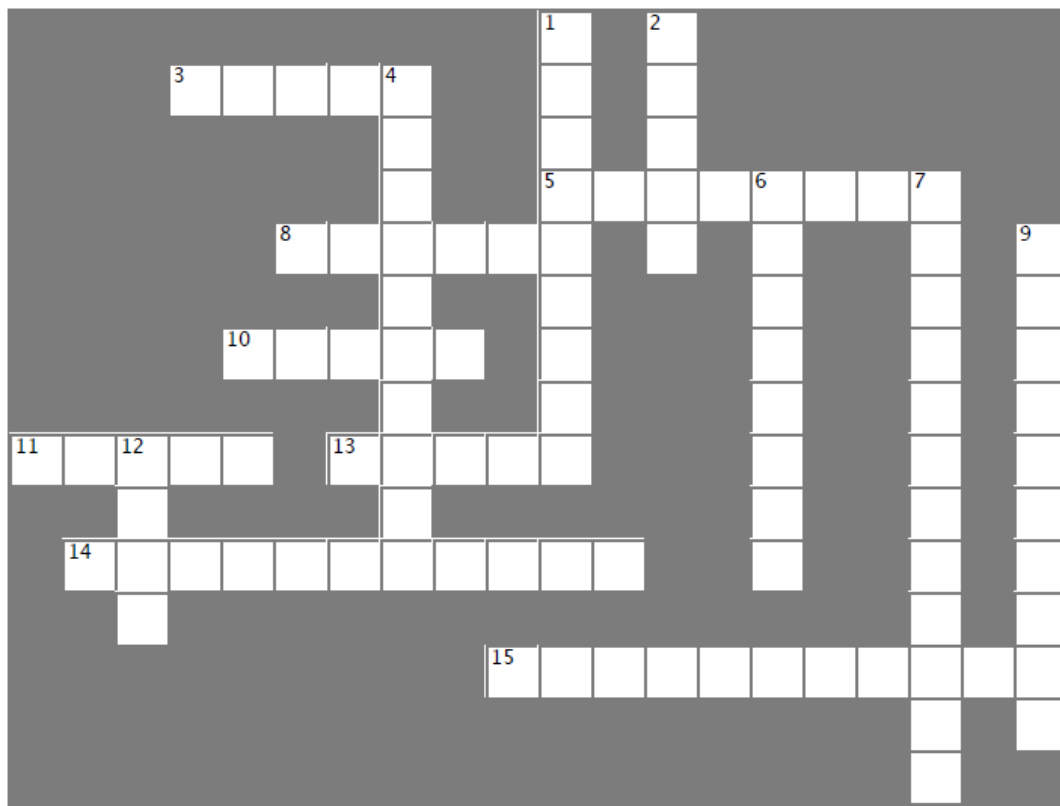
Aquí tenemos algunas de las propiedades que se puede dar estilo con CSS.

Across

3. Styles are defined in these.
5. Selects an element.
8. Each rule defines a set of properties and _____.
10. Defines a group of elements.
11. Property that represents font color.
13. Ornamental part of some fonts.
14. How elements get properties from their parents.
15. Property for font type.

Down

1. Fonts without serifs.
2. You can place your CSS inside these tags in an HTML file.
4. An external style file is called this.
6. With inheritance, a property set on one element is also passed down to its _____.
7. Won this time because they used external stylesheets.
9. They really wanted some style.
12. Use this element to include an external stylesheet.



Puntos importantes

- CSS utiliza sentencias simples, llamadas **reglas**.
- Cada regla da estilo a una selección de elementos HTML.
- Una típica regla consiste en un selector junto con una o más propiedades y valores.
- El selector especifica a qué elementos se aplica la regla.
- Cada sentencia propiedad=valor acaba en ;
- Todas las propiedades y valores de una regla van entre {}.
- Podemos seleccionar cualquier elemento usando su nombre como selector.
- Separando nombres de elementos entre comas, podemos seleccionar múltiples elementos a la vez

Puntos importantes

- Una de las formas más sencillas de incluir estilo en HTML es con la etiqueta `<style>`.
- Siempre deberíamos usar un enlace a una hoja de estilo externa.
- El elemento `<link>` es usado para incluir una hoja de estilo externa.
- Muchas propiedades son heredadas. Por ejemplo, `body`.
- Siempre podemos sobrescribir propiedades que son heredadas creando una regla más específica para el elemento que queremos cambiar.
- Usamos el atributo ***class*** para añadir elementos a una clase.

Puntos importantes

- Usamos un punto (.) entre el nombre del elemento y la clase para seleccionar un elemento específico en esa clase.
- Usamos **.nombreClase** para seleccionar todos los elementos pertenecientes a esa clase.
- Un elemento puede pertenecer a más de una clase, colocando múltiples nombres de clase en el atributo clase con espacios entre los nombres.
- Podemos validar nuestro CSS usando el W3C Validator, en:
 - <http://jigsaw.w3.org/css-validator/>.