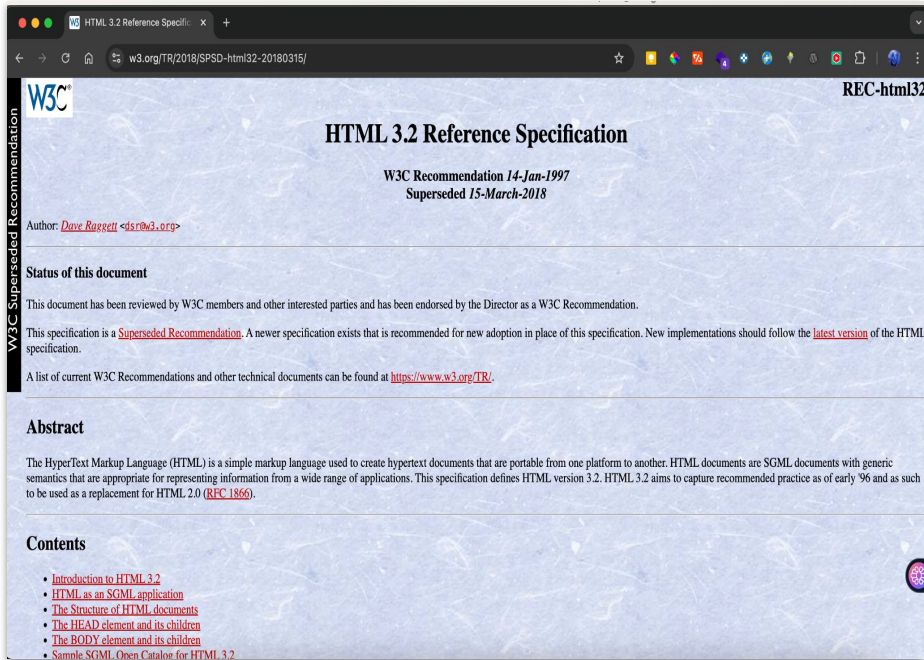


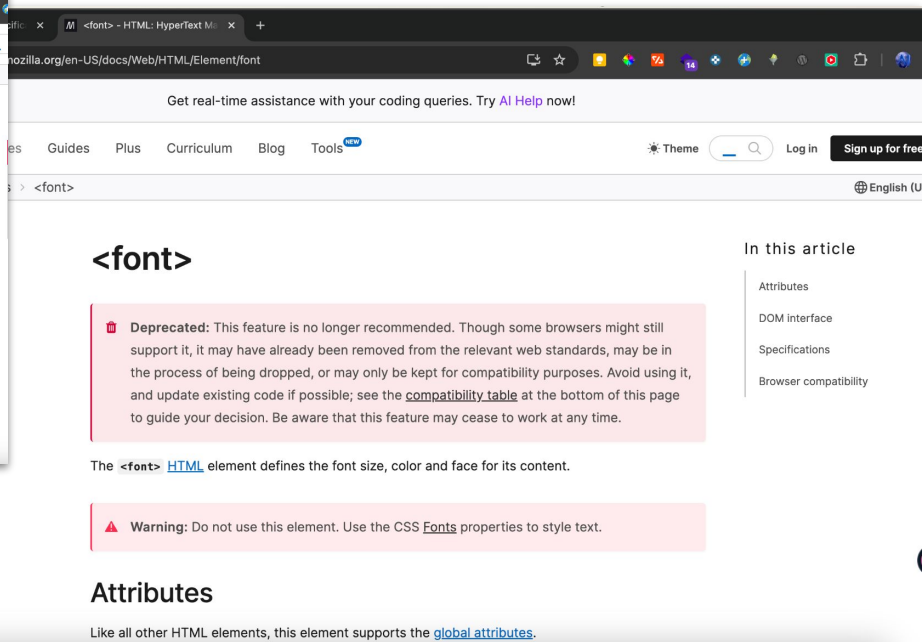
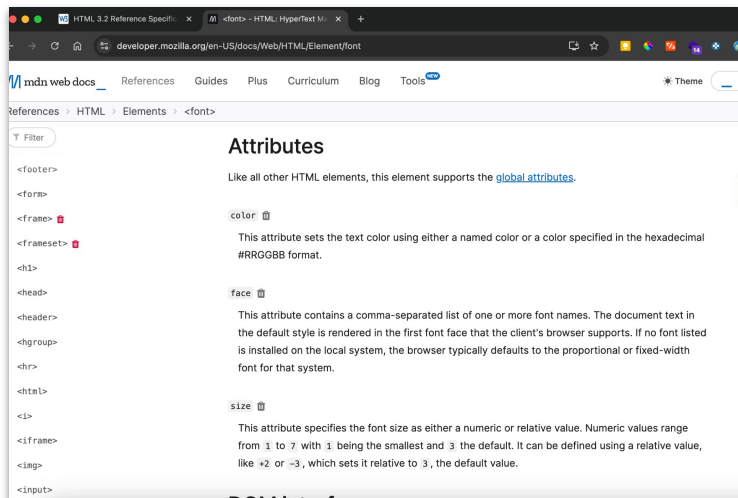
Introducción a CSS

HTML 3.2



Se incluye la etiqueta

Atributos de la etiqueta



Hoja de Estilo en Cascada v.1

Håkon Wium Lie



Håkon Wium Lie, March 2009

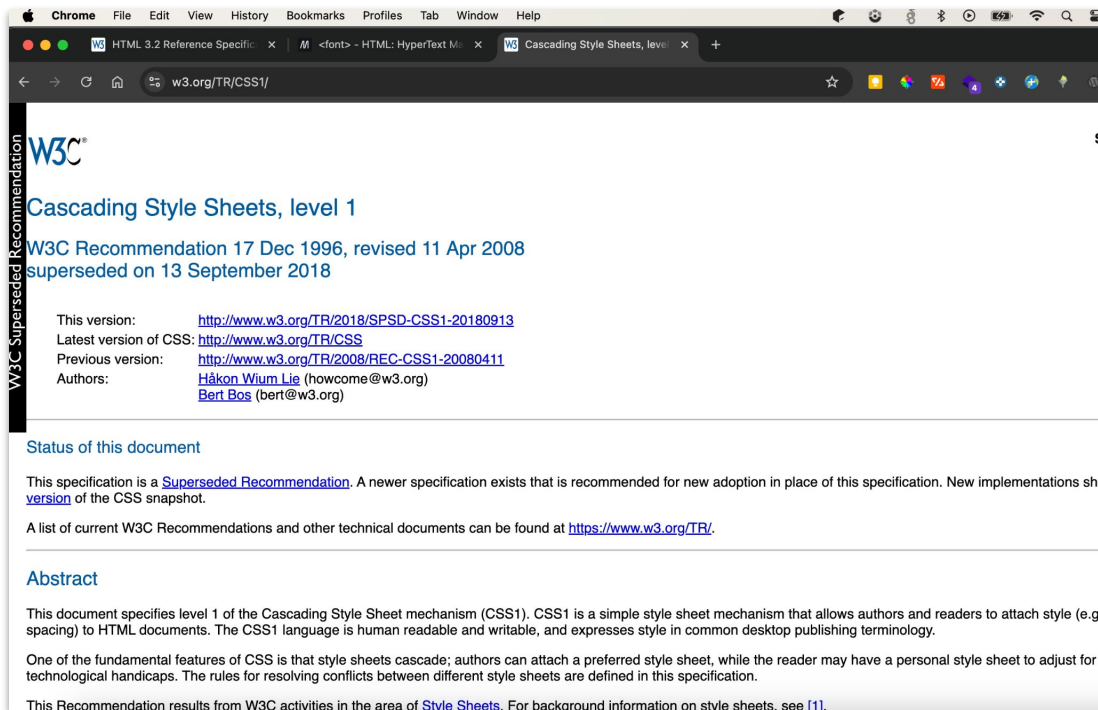
Born July 26, 1965 (age 59)
Halden, Norway

Nationality Norwegian

Employer YesLogic

Known for Cascading Style Sheets

Website [Personal homepage of Håkon W. Lie](#)

A screenshot of a web browser displaying the W3C Cascading Style Sheets, level 1 page. The browser's address bar shows "w3.org/TR/CSS1/". The page has a vertical sidebar on the left with the text "W3C Superseded Recommendation". The main content area features the W3C logo, the title "Cascading Style Sheets, level 1", and the text "W3C Recommendation 17 Dec 1996, revised 11 Apr 2008 superseded on 13 September 2018". Below this, there are links for "This version:", "Latest version of CSS:", "Previous version:", and "Authors:". The "Status of this document" section states that this is a "Superseded Recommendation" and provides a link to the current W3C Recommendations. The "Abstract" section describes the CSS1 language and its purpose.

Chrome File Edit View History Bookmarks Profiles Tab Window Help

HTML 3.2 Reference Specific... x M - HTML: HyperText Me... x W3 Cascading Style Sheets, level... x +

w3.org/TR/CSS1/

W3C[®]

Cascading Style Sheets, level 1

W3C Recommendation 17 Dec 1996, revised 11 Apr 2008
superseded on 13 September 2018

This version: <http://www.w3.org/TR/2018/SPSD-CSS1-20180913>
Latest version of CSS: <http://www.w3.org/TR/CSS>
Previous version: <http://www.w3.org/TR/2008/REC-CSS1-20080411>
Authors: [Håkon Wium Lie](#) (howcome@w3.org)
[Bert Bos](#) (bert@w3.org)

Status of this document

This specification is a [Superseded Recommendation](#). A newer specification exists that is recommended for new adoption in place of this specification. New implementations should use the [version](#) of the CSS snapshot.

A list of current W3C Recommendations and other technical documents can be found at <https://www.w3.org/TR/>.

Abstract

This document specifies level 1 of the Cascading Style Sheet mechanism (CSS1). CSS1 is a simple style sheet mechanism that allows authors and readers to attach style (e.g. spacing) to HTML documents. The CSS1 language is human readable and writable, and expresses style in common desktop publishing terminology.

One of the fundamental features of CSS is that style sheets cascade; authors can attach a preferred style sheet, while the reader may have a personal style sheet to adjust for technological handicaps. The rules for resolving conflicts between different style sheets are defined in this specification.

This Recommendation results from W3C activities in the area of [Style Sheets](#). For background information on style sheets, see [\[1\]](#).

Mientras trabajaba con Tim Berners-Lee y Robert Cailliau en el CERN en 1994, propuso el concepto de hojas de estilo en cascada (CSS). Como escaparate y banco de pruebas, integró CSS en el navegador web Arena, que se convirtió en la primera implementación de CSS.

Después de unirse al W3C en 1995, trabajó en las especificaciones CSS, incluidas CSS1, CSS2, y RFC 2318 (marzo de 1998). La mayoría de estas especificaciones se desarrollaron con Bert Bos, considerado cocreador de CSS. Durante la siguiente década, CSS se estableció como uno de los estándares web fundamentales, con un profundo impacto en la tipografía, la estética y la accesibilidad en la web.

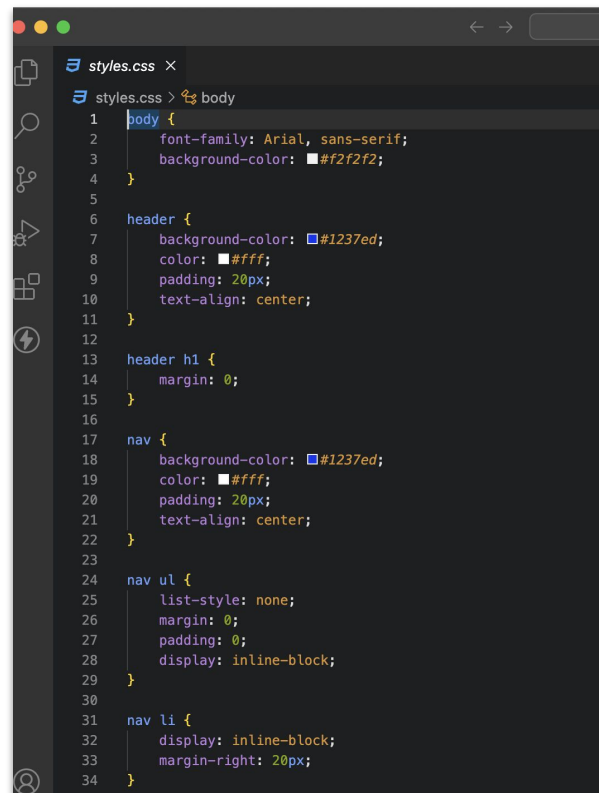
Qué es CSS?

Cascading Style Sheets

Hojas de Estilo en Cascada



CSS, que significa "Cascading Style Sheets" (Hojas de Estilo en Cascada), es un lenguaje utilizado para describir la presentación de un documento escrito en HTML o XML.

A screenshot of a code editor window titled 'styles.css'. The editor shows CSS code for a web page. The code is as follows:

```
1 body {  
2   font-family: Arial, sans-serif;  
3   background-color: #f2f2f2;  
4 }  
5  
6 header {  
7   background-color: #1237ed;  
8   color: #fff;  
9   padding: 20px;  
10  text-align: center;  
11 }  
12  
13 header h1 {  
14   margin: 0;  
15 }  
16  
17 nav {  
18   background-color: #1237ed;  
19   color: #fff;  
20   padding: 20px;  
21   text-align: center;  
22 }  
23  
24 nav ul {  
25   list-style: none;  
26   margin: 0;  
27   padding: 0;  
28   display: inline-block;  
29 }  
30  
31 nav li {  
32   display: inline-block;  
33   margin-right: 20px;  
34 }
```

CSS es fundamental porque separa el contenido de la presentación. Esto significa que puedes cambiar el aspecto de un sitio web sin tener que modificar su contenido. Además, permite que un mismo documento HTML pueda tener diferentes estilos dependiendo del dispositivo o la pantalla desde la que se acceda (como una computadora, una tableta o un teléfono móvil).



CSS Puro

CSS puro es simplemente escribir las reglas de estilo directamente en un archivo `.css`. Aquí es donde defines los estilos sin ayuda de herramientas adicionales.

```
body {  
    font-family: Arial, sans-serif;  
    background-color: #f2f2f2;  
}  
  
header {  
    background-color: #1237ed;  
    color: #fff;  
    padding: 20px;  
    text-align: center;  
}
```

Las hojas de estilo, específicamente las hojas de estilo en cascada (CSS, por sus siglas en inglés), son archivos que contienen reglas para definir la apariencia visual de los elementos HTML.

Principales funciones y características

1. Separación del Contenido y la Presentación
2. Reutilización
3. Compatibilidad con Múltiples Dispositivos
4. Jerarquía y Cascada
5. Flexibilidad y Control Preciso

Los estilos en CSS se describen mediante reglas que consisten en selectores y declaraciones.

- **Selectores:** Identifican los elementos a estilizar. Pueden ser simples (como etiquetas HTML, clases, y IDs) o complejos (como combinaciones de estos con pseudoclases y pseudoelementos).

Ejemplo: `p { color: red; }` aplica el color rojo a todos los párrafos (`<p>`)

- **Declaraciones:** Consisten en una propiedad y un valor, separados por dos puntos y finalizados con un punto y coma.

Ejemplo: `color: blue;` establece el color del texto a azul.

Los estilos se pueden aplicar de tres maneras principales

- **Estilos en Línea:** Se añaden directamente a los elementos HTML mediante el atributo style.

Ejemplo: `<p style="color: blue;">Texto azul</p>`

Los estilos se pueden aplicar de tres maneras principales

- **Estilos Internos:** Se definen dentro de una etiqueta `<style>` en el `<head>` del documento HTML.

```
<head> <style> p { color: blue; } </style> </head>
```

Los estilos se pueden aplicar de tres maneras principales

- **Estilos Externos:** Se definen en archivos CSS externos que se enlazan al documento HTML mediante la etiqueta <link>.

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

Los estilos se pueden aplicar de tres maneras principales

- **Estilos Externos:** Se definen en archivos CSS externos que se enlazan al documento HTML mediante la etiqueta <link>.

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


El lenguaje de marcas HTML se complementa con CSS para crear páginas web bien diseñadas. Mientras que HTML proporciona la estructura y el contenido, CSS se encarga de la presentación visual.

- **Clases y IDs:** Permiten aplicar estilos específicos a elementos particulares.

Ejemplo:

Html

```
<p class="highlight">Texto resaltado</p>
```

Css

```
.highlight { background-color: yellow; }
```

El lenguaje de marcas HTML se complementa con CSS para crear páginas web bien diseñadas. Mientras que HTML proporciona la estructura y el contenido, CSS se encarga de la presentación visual.

- **Clases y IDs:** Permiten aplicar estilos específicos a elementos particulares.

Ejemplo:

Html

```
<p class="highlight">Texto resaltado</p>
```

Css

```
.highlight { background-color: yellow; }
```

- **Clases y IDs:** Permiten aplicar estilos específicos a elementos particulares.

Ejemplo:

Html

```
<p>  
<a id="contact" href="https://miweb.com/contact.html">Contacto</a>  
</p>
```

Css

```
#contact { color: red; }
```

- **Pseudoclases y Pseudoelementos:** Permiten estilizar estados específicos de los elementos o partes específicas de los mismos.

Ejemplo:

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

- **Pseudoclases y Pseudoelementos:** Permiten estilizar estados específicos de los elementos o partes específicas de los mismos.

Ejemplo:

```
.required::after {content: "*";color: red;}
```

Las pseudoclases son palabras clave que se añaden a los selectores en CSS para especificar un estado especial de los elementos. Se utilizan para aplicar estilos cuando los elementos cumplen con ciertas condiciones, como cuando se pasa el ratón por encima de ellos o cuando han sido visitados.

Ejemplos:

:hover: Aplica un estilo cuando el usuario pasa el cursor sobre un elemento.

Cambiar el color de un enlace cuando se pasa el ratón por encima.

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

Ejemplos:

`:visited`: Aplica un estilo a los enlaces que ya han sido visitados.

Cambiar el color de un enlace después de que ha sido clicado.

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

`:nth-child(n)`: Aplica un estilo a los elementos que son el enésimo hijo de su padre.

Cambiar el fondo de las filas impares de una tabla.

```
tr:nth-child(odd) {background-color: #f2f2f2;}
```

Los pseudoelementos se utilizan para aplicar estilos a partes específicas de un elemento. Permiten estilizar subpartes de elementos, como la primera letra o la primera línea de un párrafo, o insertar contenido antes o después de un elemento.

Ejemplos:

`::before`: Inserta contenido antes del contenido real de un elemento.

Ejemplo: Añadir un icono antes de un enlace.

```
a::before {content: "🔗";}
```


`::after`: Inserta contenido después del contenido real de un elemento.

Ejemplo: Añadir un asterisco después de un campo obligatorio en un formulario.

```
.required::after {content: "*";color: red;}
```

`::first-line`: Aplica un estilo a la primera línea de un párrafo.

Ejemplo: Hacer la primera línea de un párrafo en negrita.

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

`::first-letter`: Aplica un estilo a la primera letra de un párrafo.

Ejemplo: Hacer la primera letra de un párrafo grande y de color.

```
p::first-letter {font-size: 3em; color: red;}
```

Cascada, especificidad y herencia

Fundamentalmente, CSS consiste en declarar reglas: bajo diversas condiciones, queremos que ciertas cosas sucedan.

Cuando dos o más reglas apuntan al mismo elemento en su página, las reglas pueden proporcionar declaraciones contradictorias.

Cascada, especificidad y herencia

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Marcado para el encabezado de la página.</title>
8  </head>
9
10 <body>
11     <header class="page-header">
12         <h1 id="page-title" class="title">Cafetería El Buen Café</h1>
13         <nav>
14             <ul id="main-nav" class="nav">
15                 <li><a href="#">Inicio</a></li>
16                 <li><a href="#">Café</a></li>
17                 <li><a href="#">Cervezas</a></li>
18                 <li><a href="#" class="featured">Especiales</a></li>
19             </ul>
20         </nav>
21     </header>
22 </body>
23
24 </html>
```

Cascada, especificidad y herencia

```
1  h1{  
2      font-family: serif;  
3  }  
4  
5  #page-title{  
6      font-family: sans-serif;  
7  }  
8  
9  .title {  
10     font-family: monospace;  
11 }  
12
```

Cascada, especificidad y herencia

Analicemos la cascada: cuando las declaraciones entran en conflicto, la cascada considera tres aspectos para resolver la diferencia.

1. Origen de la hoja de estilo
2. Especificidad del selector
3. Orden de origen

Cascada, especificidad y herencia

Las hojas de estilo que nosotros añadimos no son las únicas que el navegador aplica. Existen diferentes tipos, orígenes, de hojas de estilo.

- Las que añadimos son llamadas **author styles** (estilos de autor). **Tienen prioridad sobre los estilos del navegador.**
- Las que el navegador añade por defecto se llaman **user agent** (estos estilos varían un poco entre navegadores).

Cascada, especificidad y herencia

```
1  /* Estilo para el encabezado h1 */
2  h1 {
3      color: #2f4f4f; /* Color del texto: gris verdoso oscuro */
4      margin-bottom: 10px; /* Espacio de 10px debajo del encabezado */
5  }
6
7  /* Estilo para el contenedor de navegación con ID "main-nav" */
8  #main-nav {
9      margin-top: 10px; /* Espacio de 10px encima del contenedor de navegación */
10     list-style: none; /* Elimina los marcadores de lista por defecto */
11     padding-left: 0; /* Elimina el padding a la izquierda del contenedor */
12 }
13
14 /* Estilo para cada elemento de lista dentro del contenedor de navegación */
15 #main-nav li {
16     display: inline-block; /* Muestra los elementos de la lista en línea */
17 }
18
19 /* Estilo para los enlaces dentro del contenedor de navegación */
20 #main-nav a {
21     color: #ffffff; /* Color del texto del enlace: blanco */
22     background-color: #13a4a4; /* Color de fondo del enlace: turquesa */
23     padding: 5px; /* Añade un padding de 5px alrededor del contenido del enlace */
24     border-radius: 2px; /* Redondea ligeramente las esquinas del fondo del enlace */
25     text-decoration: none; /* Elimina el subrayado predeterminado de los enlaces */
26 }
```

Reduce los márgenes

Quita los estilos de las listas del user agent

Hace que los elementos de lista se posicionen uno al lado del otro en vez de apilarse

Proporciona una apariencia similar a un botón para los enlaces de navegación

Cascada, especificidad y herencia

Existe una excepción a la regla de estilo de origen: declaraciones que son marcadas como **important**.

```
color : #ff0000 important ;
```

1. Author Important
2. Author
3. User agent



Alta prioridad

Cascada, especificidad y herencia

Si el conflicto no se puede resolver basándose en el origen, el navegador pasará a buscar la especificidad del selector.

1. Estilos de línea

Estas son, en efecto, declaraciones de alcance, que anulan cualquier declaración aplicada desde su hoja de estilo o etiquetas de estilo.

Cascada, especificidad y herencia

index.html > html

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>Marcado para el encabezado de la página.</title>
8    <link rel="stylesheet" href="/styles.css">
9  </head>
10
11 <body>
12   <header class="page-header">
13     <h1 id="page-title" class="title">Cafetería El Buen Café</h1>
14     <nav>
15       <ul id="main-nav" class="nav">
16         <li><a href="#">Inicio</a></li>
17         <li><a href="#">Café</a></li>
18         <li><a href="#">Cervezas</a></li>
19         <li><a href="#" class="featured" id="red" style="background-color: orange;">
20           Especiales</a></li>
21       </ul>
22     </nav>
23   </header>
24 </body>
25
26
27 </html>
```

Estilo de línea

Para poder sobrescribir esta regla se debe utilizar important!

Cascada, especificidad y herencia

```
14  /* Estilo para cada elemento de lista dentro del contenedor de navegación */
15  #main-nav li {
16      display: inline-block; /* Muestra los elementos de la lista en línea */
17  }
18
19  /* Estilo para los enlaces dentro del contenedor de navegación */
20  #main-nav a {
21      color: □#ffffff; /* Color del texto del enlace: blanco */
22      background-color: █#13a4a4; /* Color de fondo del enlace: turquesa */
23      padding: 5px; /* Añade un padding de 5px alrededor del contenido del enlace */
24      border-radius: 2px; /* Redondea ligeramente las esquinas del fondo del enlace */
25      text-decoration: none; /* Elimina el subrayado predeterminado de los enlaces */
26  }
27  .featured {
28      background-color: █purple;
29  }
30
```

← Mayor especificidad

Cascada, especificidad y herencia

Las reglas exactas de la especificidad son:

- Si un selector tiene más Id's este gana (es el más específico).
- Si el resultado es un empate, gana el selector con más clases.
- Si el resultado es un empate, gana el selector con más nombres de etiquetas.

Cascada, especificidad y herencia

```
39
40  html body header h1 {
41      color: ■blue;
42  }
43
44  body header.page-header h1 {
45      color: ■orange;
46  }
47
48  .page-header .title {
49      color: ■green;
50  }
51
52  #page-title {
53      color: ■red;
54  }
55  |
```

1

2

3

4

-

+

Cascada, especificidad y herencia

Selector	IDs	Classes	Tags	Notation
html body header h1	0	0	4	0,0,4
body header.page-header h1	0	1	3	0,1,3
.page-header .title	0	2	0	0,2,0
#page-title	1	0	0	1,0,0

Cascada, especificidad y herencia

```
#main-nav a {  
  color: white;  
  background-color: #13a4a4;  
  padding: 5px;  
  border-radius: 2px;  
  text-decoration: none;  
}  
  
.featured {  
  background-color: orange !important;  
}
```

← Hace que la declaración sea importante; ahora es un origen de mayor prioridad

Cascada, especificidad y herencia

```
#main-nav a {  
  color: white;  
  background-color: #13a4a4;  
  padding: 5px;  
  border-radius: 2px;  
  text-decoration: none;  
}
```

La especificidad sigue siendo (1,0,1)

```
#main-nav .featured {  
  background-color: orange;  
}
```

Aumenta la especificidad a (1,1,0)

La anotación !important ya no es necesaria.

Cascada, especificidad y herencia

```
.nav {  
  margin-top: 10px;  
  list-style: none;  
  padding-left: 0;  
}
```

Cambia
"#main-nav" a
".nav" en toda
la hoja

```
.nav li {  
  display: inline-block;  
}
```

```
.nav a {  
  color: white;  
  background-color: #13a4a4;  
  padding: 5px;  
  border-radius: 2px;  
  text-decoration: none;  
}
```

Reduce la
primera
especificidad
(0,1,1)

```
.nav .featured {  
  background-color: orange;  
}
```


Aumenta la
segunda
especificidad
(0,2,0)

Cascada, especificidad y herencia

Si el origen y la especificidad son la misma, entonces la declaración que aparece de última en la hoja de estilos, o aparece en la hoja de estilos incluida tarde en la página.

```
.nav a {  
  color: white;  
  background-color: #13a4a4;  
  padding: 5px;  
  border-radius: 2px;  
  text-decoration: none;  
}
```

```
a.featured {  
  background-color: orange;  
}
```



Hace que las
especificidades
sean iguales

Cascada, especificidad y herencia

Este es el código demuestra el comportamiento.

```
index.html X styles.css
index.html > html > body > header.page-header > nav
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Marcado para el encabezado de la página.</title>
8   <link rel="stylesheet" href="./styles.css">
9 </head>
10
11 <body>
12   <header class="page-header">
13     <h1 id="page-title" class="title">Cafetería El Buen Café</h1>
14     <nav>
15       <ul id="main-nav" class="nav">
16         <li><a href="#">Inicio</a></li>
17         <li><a href="#">Café</a></li>
18         <li><a href="#">Cervezas</a></li>
19         <li><a href="#" class="featured" id="red"><!-- style="background-color: orange;" -->
20           Especiales</a></li>
21       </ul>
22     </nav>
23   </header>
24   <main class="main">
25     <p>Asegurate de verificar nuestra <a href="#" class="featured">disponibilidad</a></p>
```

El enlace
destacado fuera de
la navegación
obtendrá el estilo



Trabajando con unidades relativas

**El poder de la unidades
relativas**

El poder de la unidades relativas

Las unidades relativas son una de las herramientas que proporciona CSS para trabajar en este nivel de abstracción. En lugar de establecer un tamaño de fuente de 14px, puedes configurarlo para que escale proporcionalmente al tamaño de la ventana. O puedes establecer el tamaño de todo lo que hay en la página en relación con el tamaño de fuente base y luego cambiar el tamaño de toda la página con una sola línea de código.

El poder de la unidades relativas

A medida que la tecnología mejoró y los fabricantes introdujeron monitores de mayor resolución y surgieron los teléfonos inteligentes, hubo que abandonar las columnas de una cantidad conocida de píxeles y comenzar a pensar en un diseño responsivo.

Responsivo: en CSS, esto se refiere a estilos que responden de manera diferente, según el tamaño de la ventana del navegador.

El poder de la unidades relativas

En CSS, 1em significa el tamaño de fuente del elemento actual; su valor varía dependiendo del elemento al que lo estés aplicando.

En el siguiente ejemplo el relleno tiene un valor específico de 1em. Esto se multiplica por el tamaño de la fuente, lo que produce un relleno renderizado de 16px.

El poder de la unidades relativas

```
</nav>
<main class="main">
  <p>Asegurate de verificar nuestra <a href="" class="featured">disponibilidad</a></p>
  <p class="padded">Hemos creado asociaciones con pequenas granjas de todo el mundo para seleccionar los
    granos en el pico de la temporada. Luego tostamos cuidadosmente en pequeños lotes para maximizar su
    potencial.</p>
</main>
</header>
</body>
</html>
```

```
a.featured {
  background-color: orange;
}

.padded {
  font-size: 16px;
  padding: 1em;
}
```

El poder de la unidades relativas

Cafetería El Buen Café

[Inicio](#) [Café](#) [Cervezas](#) [Especiales](#)

p.padded 869 x 69

[disponibilidad](#)

Hemos creado asociaciones con pequeñas granjas de todo el mundo para seleccionar los granos en el pico de la temporada. Luego tostamos cuidadosamente en pequeños lotes para maximizar su potencial.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    </head>
  <body>
    <header class="page-header">
      <h1 id="page-title" class="title">Cafetería El Buen Café</h1>
      <nav>
        </nav>
      <main>
        <p>
          </p>
        <p class="padded">
          Hemos creado asociaciones con pequeñas granjas de todo el mundo para
          seleccionar los granos en el pico de la temporada. Luego tostamos
          cuidadosamente en pequeños lotes para maximizar su potencial.
        </p>
      </main>
    </header>
  </body>
</html>
```

html body header.page-header main **p.padded**

Styles Computed Layout Event Listeners DOM Breakpoints Properties >>

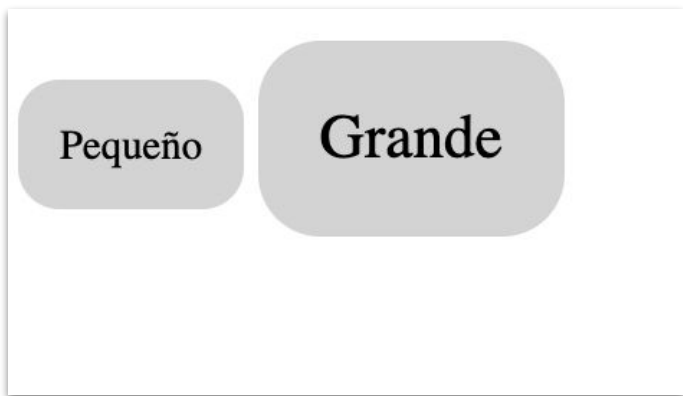
margin 16
border 16
padding 16
837x37

Filter Show all Group

- display block
- font-size 16px
- height 37px
- margin-block-end 16px
- margin-block-start 16px
- margin-inline-end 0px
- margin-inline-start 0px
- padding-bottom 16px

El poder de la unidades relativas

Usar ems puede ser conveniente al configurar propiedades como relleno, alto, ancho o radio de borde porque escalará uniformemente con el elemento si hereda diferentes tamaños de fuente o si el usuario cambia la configuración de la fuente.



Elemento con tamaño relativo en el padding y en el border radius

El poder de la unidades relativas

Aplicando ems a diferentes elementos.

```
potencial.</p>

<span class="box box-small">Pequeño</span>
<span class="box box-large">Grande</span>
<br><br><br><br><br>

</main>

</header>
</body>

</html>
```

```
.box {
  padding: 1em;
  border-radius: 1em;
  background-color: lightgray;
}

.box-small {
  font-size: 12px;
}

.box-large {
  font-size: 18px;
}
```

El poder de la unidades relativas

Usando ems para definir el tamaño de la fuente.

Nosotros amamos el Café.

Nosotros amamos el Café.

Dos tamaños de fuente diferente usando ems

El poder de la unidades relativas

Usando ems para definir el tamaño de la fuente.

The diagram illustrates the flow from code to the rendered output. On the left, the HTML code is shown: `<h1>Nosotros amamos el Café.</h1>`, `<p class="slogan">Nosotros amamos el Café.</p>`, `</main>`, `</header>`, `</body>`, and `</html>`. In the middle, the CSS code is shown: `.slogan { font-size: 1.2em; }`. On the right, the rendered output is shown: a large heading "Nosotros amamos el Café." and a paragraph "Nosotros amamos el Café." The heading is styled with a large font size, and the paragraph is styled with a slightly smaller font size, demonstrating the effect of the CSS rule.

El poder de la unidades relativas

Usando ems para definir el tamaño de la fuente junto a ems para otras propiedades.

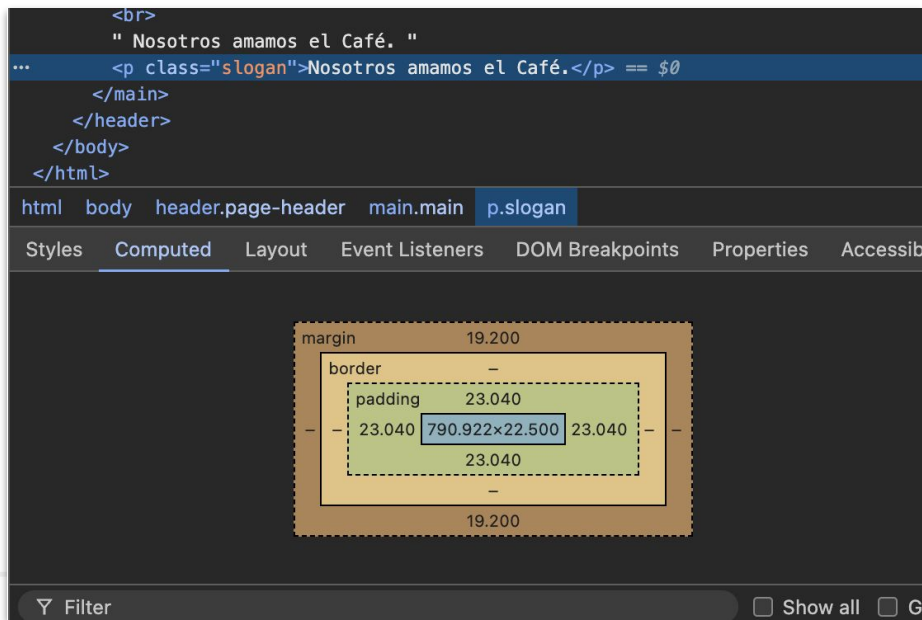
```
33  
34     Nosotros amamos el Café.  
35     <p class="slogan">Nosotros amamos el Café.</p>  
36 </main>  
37  
38 </header>  
39 </body>  
40  
41 </html>  
  
77  
78 body {  
79     font-size: 16px;  
80 }  
81  
82 .slogan {  
83     font-size: 1.2em;  
84     padding: 1.2em;  
85     background-color: #ccc;  
86 }
```

Nosotros amamos el Café.

Nosotros amamos el Café.

El poder de la unidades relativas

```
77  
78 body {  
79     font-size: 16px;  
80 }  
81  
82 .slogan {  
83     font-size: 1.2em;  
84     padding: 1.2em;  
85     background-color: #ccc;  
86 }
```



El problema de la fuente que se encoge

Esto se vuelve evidente cuando usamos ems para el tamaño de fuente en las listas.

```
<h3>Listas anidadas </h3>
<ul>
  <li>Primer nivel
    <ul>
      <li>Segundo nivel
        <ul>
          <li>Tercer nivel
            <ul>
              <li>Cuarto nivel
                <ul>
                  <li>Quinto nivel</li>
                </ul>
              </li>
            </ul>
          </li>
        </ul>
      </li>
    </ul>
  </li>
</ul>
</ul>
```

Esta lista anidada dentro del 1er. Nivel y heredando su tamaño de fuente.

Este anidado dentro del 2do. y heredando el tamaño de la fuente.

Y así sucesivamente...

El problema de la fuente que se encoge

Esto se vuelve evidente cuando usamos ems para el tamaño de fuente en las listas.

```
body {  
  font-size: 16px;  
}  
  
.slogan {  
  font-size: 1.2em;  
  padding: 1.2em;  
  background-color: #ccc;  
}  
  
/* Estilo listas anidadas */  
ul {  
  font-size: 0.8em;  
}
```

Listas anidadas

- Primer nivel
 - Segundo nivel
 - Tercer nivel
 - Cuarto nivel
 - Quinto nivel

El problema de la fuente que se encoge

Esto se puede resolver de la siguiente forma. Listas dentro de listas deben tener el mismo tamaño que el padre.

```
,  
  
/* Estilo listas anidadas */  
ul {  
    font-size: 0.8em;  
}
```

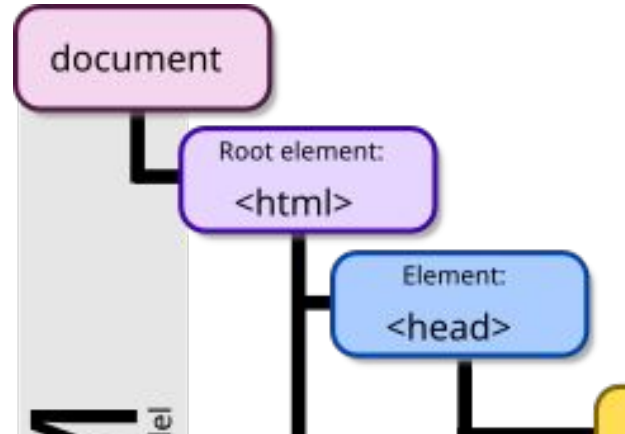
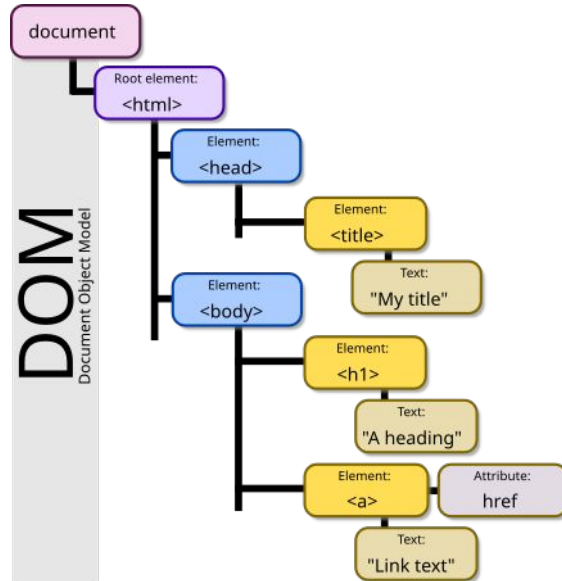
```
ul ul {  
    font-size: 1em;  
}
```

Listas anidadas

- Primer nivel
 - Segundo nivel
 - Tercer nivel
 - Cuarto nivel
 - Quinto nivel

Usando rems para tamaño de fuente

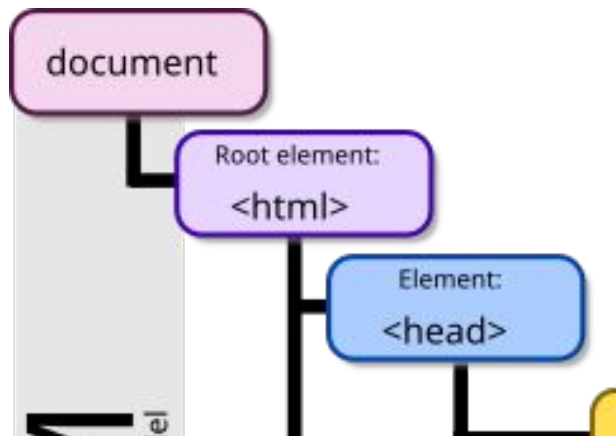
Cuando el navegador analiza un página html, crea una representación de todos los elementos de la página, a esta representación se le llama DOM (Document Object Model).



Usando rems para tamaño de fuente

Rem es la abreviatura de Root em. En vez de ser relativos al elemento actual, los rems son relativos al al elemento raíz.

```
✓ :root {  
  font-size: 1em;  
}  
  
✓ ul {  
  font-size: 0.8rem;  
}
```



Usando rems para tamaño de fuente

Como podemos evidenciar los rems nos simplifican muchas de las complejidades involucradas con los ems.

