

# Hojas de Estilo en Cascada, nivel 2

## Especificación CSS2

Este documento es una copia traducida de la Recomendación del W3C sobre las Hojas de Estilo en Cascada, Nivel 2. Este documento puede contener errores de traducción. Una copia de la versión normativa, en inglés, puede encontrarse en:

<http://www.w3.org/TR/1998/REC-CSS2-19980512>

La versión en castellano puede encontrarse en:

<http://www.carlos-benavidez.net/css2/cover.html>

Traducción: [Carlos Benavidez](#)



REC-CSS2-19980512

# Hojas de Estilo en Cascada, nivel 2

## Especificación CSS2

Recomendación del W3C 12-May-1998

Esta versión:

<http://www.w3.org/TR/1998/REC-CSS2-19980512>

Última versión:

<http://www.w3.org/TR/REC-CSS2>

Versión previa:

<http://www.w3.org/TR/1998/PR-CSS2-19980324>

Editores:

Bert Bos <[bbos@w3.org](mailto:bbos@w3.org)>

Håkon Wium Lie <[howcome@w3.org](mailto:howcome@w3.org)>

Chris Lilley <[chris@w3.org](mailto:chris@w3.org)>

Ian Jacobs <[ij@w3.org](mailto:ij@w3.org)>

## Resumen

Esta especificación define las Hojas de Estilo en Cascada, nivel 2 (CSS2). CSS2 es un lenguaje de hoja de estilo que permite que los autores y los usuarios asocien un estilo (por ejemplo, fuentes, espaciado y señales sonoras) a los documentos estructurados (por ejemplo, documentos HTML y aplicaciones XML). Separando el estilo de presentación del contenido de los documentos, CSS2 simplifica la creación y mantenimiento de los sitios Web.

CSS2 se cimienta en CSS1 (véase [\[CSS1\]](#)) y, con muy pocas excepciones, todas las hojas de estilo CSS1 válidas son hojas de estilo CSS2 válidas. CSS2 brinda soporte a hojas de estilo específicas para cada medio, de modo que los autores puedan adaptar la presentación de sus documentos a los browsers visuales, a los dispositivos sonoros, a las impresoras, a los dispositivos braille, de mano, etc. Esta especificación también soporta el posicionamiento de contenidos, fuentes descargables, disposición de la página, aspectos para la internacionalización, contadores y numeradores automáticos, y algunas características relacionadas con la interfaz del usuario.

## Estado de este documento

Este documento ha sido revisado por los Miembros de W3C y otras partes interesadas y aprobado por el Director como Recomendación del W3C. Es un documento estable y puede ser utilizado como material de referencia o ser citado como referencia normativa en otro documento. El papel del W3C en la elaboración de la Recomendación es atender a la especificación y promover su amplia difusión. Esto incrementa la funcionalidad y la interoperabilidad de la Web.

Una lista de las Recomendaciones actuales del W3C y de otros documentos técnicos se puede encontrar en <http://www.w3.org/TR>.

La discusión pública de las características de las CSS tiene lugar en [www-style@w3.org](mailto:www-style@w3.org).

## Formatos disponibles

La especificación CSS2 está disponible en los siguientes formatos:

HTML:

<http://www.w3.org/TR/1998/REC-CSS2-19980512>

un archivo de texto puro:  
<http://www.w3.org/TR/1998/REC-CSS2-19980512/css2.txt>,  
HTML como archivo tar (gzip):  
<http://www.w3.org/TR/1998/REC-CSS2-19980512/css2.tgz>,  
HTML como archivo zip (éste es un archivo 'zip ' no un 'exe'):  
<http://www.w3.org/TR/1998/REC-CSS2-19980512/css2.zip>,  
así como archivo PostScript (gzip):  
<http://www.w3.org/TR/1998/REC-CSS2-19980512/css2.ps.gz>,  
y un archivo PDF:  
<http://www.w3.org/TR/1998/REC-CSS2-19980512/css2.pdf>.

En caso de una discrepancia entre las distintas formas de la especificación, <http://www.w3.org/TR/1998/REC-CSS2-19980512> se considera la versión definitiva.

## Lenguajes disponibles

La versión en inglés de esta especificación es la única versión normativa. Sin embargo, por traducciones en otros lenguajes ver <http://www.w3.org/Style/css2-updates/translations.html>.

## Erratas

La lista de errores conocidos en esta especificación está disponible en <http://www.w3.org/Style/css2-updates/REC-CSS2-19980512-errata.html>. Por favor, informe de los errores en este documento a [css2-editors@w3.org](mailto:css2-editors@w3.org).

**Nota de la traducción:**  
Los errores a que se refiere el párrafo anterior pertenecen, obviamente, a la versión original en inglés. Esta traducción ya contiene las correcciones del caso.  
Por favor, si desea informar de errores encontrados en esta versión en castellano o tiene cualquier tipo de sugerencia, envíe su mensaje a: [carlos.benavidez@sidar.org](mailto:carlos.benavidez@sidar.org)

## Tabla de Contenidos Abreviada

- [1 Acerca de la Especificación CSS2](#)
- [2 Introducción a CSS2](#)
- [3 Conformidad: Requisitos y Recomendaciones](#)
- [4 Sintaxis y tipos básicos de datos en CSS2](#)
- [5 Selectores](#)
- [6 Asignación de valores a las propiedades, Cascada y Herencia](#)
- [7 Tipos de medios](#)
- [8 Modelo de caja](#)
- [9 Modelo de formato visual](#)
- [10 Detalles del modelo de formato visual](#)
- [11 Efectos visuales](#)
- [12 Contenido generado, numeración automática y listas](#)
- [13 Medios paginados](#)
- [14 Colores y fondos](#)
- [15 Fuentes](#)
- [16 Texto](#)
- [17 Tablas](#)
- [18 Interfaz del usuario](#)
- [19 Hojas de estilo auditivas](#)
- [Apéndice A: Una hoja de estilo de ejemplo para HTML 4.0](#)
- [Apéndice B: Cambios con respecto a CSS1](#)
- [Apéndice C: Implementación y notas de desempeño para fuentes](#)
- [Apéndice D: La gramática de CSS2](#)
- [Apéndice E. Referencias](#)
- [Apéndice F. Indice de propiedades](#)
- [Apéndice G. Indice de descriptores](#)
- [Apéndice H. Indice](#)

## Tabla de Contenidos Completa

- [1 Acerca de la Especificación CSS2](#)
  - [1.1 Leyendo la especificación](#)
  - [1.2 Cómo está organizada la especificación](#)
  - [1.3 Convenciones](#)
    - [1.3.1 Elementos y atributos en el lenguaje del documento](#)
    - [1.3.2 Definición de las propiedades CSS](#)
      - [Valor](#)
      - [Inicial](#)
      - [Se aplica a](#)
      - [Se hereda](#)

- *Porcentajes*
    - *Grupos de medios*
  - 1.3.3 Propiedades resumidas
  - 1.3.4 Notas y ejemplos
  - 1.3.5 Imágenes y descripciones extensas
- 1.4 Reconocimientos
- 1.5 Aviso de Copyright
- **2 Introducción a CSS2**
  - 2.1 Breve guía de CSS2 para HTML
  - 2.2 Breve guía de CSS2 para XML
  - 2.3 El modelo de procesamiento de CSS2
    - 2.3.1 El lienzo
    - 2.3.2 Modelo de destinatarios de CSS2
  - 2.4 Reglas de diseño de CSS
- **3 Conformidad: Requisitos y Recomendaciones**
  - 3.1 Definiciones
  - 3.2 Conformidad
  - 3.3 Condiciones de error
  - 3.4 Tipo de contenido text/css
- **4 Sintaxis y tipos básicos de datos en CSS2**
  - 4.1 Sintaxis
    - 4.1.1 Comandos
    - 4.1.2 Palabras clave
    - 4.1.3 Caracteres y mayúsculas/minúsculas
    - 4.1.4 Estamentos
    - 4.1.5 Reglas-arroba
    - 4.1.6 Bloques
    - 4.1.7 Juegos de reglas, bloques de declaraciones y selectores
    - 4.1.8 Declaraciones y propiedades
    - 4.1.9 Comentarios
  - 4.2 Reglas para el manejo de errores en el análisis
  - 4.3 Valores
    - 4.3.1 Enteros y números reales
    - 4.3.2 Medidas
    - 4.3.3 Porcentajes
    - 4.3.4 URL + URN = URI
    - 4.3.5 Contadores
    - 4.3.6 Colores
    - 4.3.7 Angulos
    - 4.3.8 Tiempos
    - 4.3.9 Frecuencias
    - 4.3.10 Cadenas
  - 4.4 Representación de los documentos CSS
    - 4.4.1 Referencias a caracteres no representados en una codificación de caracteres
- **5 Selectores**
  - 5.1 Equivalencia de patrones
  - 5.2 Sintaxis de los selectores
    - 5.2.1 Agrupamiento
  - 5.3 Selector universal
  - 5.4 Selectores de tipos
  - 5.5 Selectores de descendientes
  - 5.6 Selectores de hijos
  - 5.7 Selectores de hermanos adyacentes
  - 5.8 Selectores de atributos
    - 5.8.1 Atributos equivalentes y valores de los atributos
    - 5.8.2 Valores predeterminados de los atributos en la DTD
    - 5.8.3 Selectores de clases
  - 5.9 Selectores de ID
  - 5.10 Pseudo-elementos y pseudo-clases
  - 5.11 Pseudo-clases
    - 5.11.1 La pseudo-clase :first-child
    - 5.11.2 Las pseudo-clases vínculo: :link y :visited
    - 5.11.3 Las pseudo-clases dinámicas: :hover, :active y :focus
    - 5.11.4 La pseudo-clase de lenguaje: :lang
  - 5.12 Pseudo-elementos
    - 5.12.1 El pseudo-elemento :first-line
    - 5.12.2 El pseudo-elemento :first-letter
    - 5.12.3 Los pseudo-elementos :before y :after
- **6 Asignación de valores a las propiedades, Cascada y Herencia**
  - 6.1 Valores especificados, computados y reales
    - 6.1.1 Valores especificados
    - 6.1.2 Valores computados
    - 6.1.3 Valores reales
  - 6.2 Herencia
    - 6.2.1 El valor 'inherit'
  - 6.3 La regla @import
  - 6.4 Cascada
    - 6.4.1 Orden de la cascada
    - 6.4.2 Las reglas !important
    - 6.4.3 Cálculo de la especificidad de un selector

- 6.4.4 Precedencia de las indicaciones de presentación fuera de CSS
- **7 Tipos de medios**
  - 7.1 Introducción a los tipos de medios
  - 7.2 Especificación de hojas de estilos dependientes de los medios
    - 7.2.1 La regla @media
  - 7.3 Tipos de medios reconocidos
    - 7.3.1 Grupos de medios
- **8 Modelo de caja**
  - 8.1 Dimensiones de la caja
  - 8.2 Ejemplo de márgenes, rellenos y bordes
  - 8.3 Propiedades del margen: 'margin-top', 'margin-right', 'margin-bottom', 'margin-left' y 'margin'
    - 8.3.1 Márgenes cerrados
  - 8.4 Propiedades del relleno: 'padding-top', 'padding-right', 'padding-bottom', 'padding-left' y 'padding'
  - 8.5 Propiedades del borde
    - 8.5.1 Ancho del borde: 'border-top-width', 'border-right-width', 'border-bottom-width', 'border-left-width' y 'border-width'
    - 8.5.2 Color del borde: 'border-top-color', 'border-right-color', 'border-bottom-color', 'border-left-color' y 'border-color'
    - 8.5.3 Estilo del borde: 'border-top-style', 'border-right-style', 'border-bottom-style', 'border-left-style' y 'border-style'
    - 8.5.4 Propiedades resumidas del borde: 'border-top', 'border-bottom', 'border-right', 'border-left' y 'border'
- **9 Modelo de formato visual**
  - 9.1 Introducción al modelo de formato visual
    - 9.1.1 El acceso visual
    - 9.1.2 Bloques de contención
  - 9.2 Control de la generación de cajas
    - 9.2.1 Elementos a nivel de bloque y cajas de bloque
      - *Cajas de bloque anónimas*
    - 9.2.2 Elementos a nivel de línea y cajas de línea
      - *Cajas de línea anónimas*
    - 9.2.3 Cajas compact
    - 9.2.4 Cajas run-in
    - 9.2.5 La propiedad 'display'
  - 9.3 Esquemas de posicionamiento
    - 9.3.1 Elección de un esquema de posicionamiento: la propiedad 'position'
    - 9.3.2 Desplazamiento de las cajas: 'top', 'right', 'bottom', 'left'
  - 9.4 Flujo normal
    - 9.4.1 Contexto de formato de bloque
    - 9.4.2 Contexto de formato de línea
    - 9.4.3 Posicionamiento relativo
  - 9.5 Flotantes
    - 9.5.1 Posicionamiento del flotante: la propiedad 'float'
    - 9.5.2 Control del flujo al costado del flotante: la propiedad 'clear'
  - 9.6 Posicionamiento absoluto
    - 9.6.1 Posicionamiento fijo
  - 9.7 Relaciones entre 'display', 'position' y 'float'
  - 9.8 Comparación entre flujo normal, flotantes y posicionamiento absoluto
    - 9.8.1 Flujo normal
    - 9.8.2 Posicionamiento relativo
    - 9.8.3 Flotar una caja
    - 9.8.4 Posicionamiento absoluto
  - 9.9 Presentación por capas
    - 9.9.1 Especificar el nivel de pila: la propiedad 'z-index'
  - 9.10 Dirección del texto: las propiedades 'direction' y 'unicode-bidi'
- **10 Detalles del modelo de formato visual**
  - 10.1 Definición de "bloque de contención"
  - 10.2 Ancho del contenido: la propiedad 'width'
  - 10.3 Computando anchos y márgenes
    - 10.3.1 Elementos no reemplazados, de línea
    - 10.3.2 Elementos reemplazados, de línea
    - 10.3.3 Elementos no reemplazados en flujo normal, a nivel de bloque
    - 10.3.4 Elementos reemplazados en flujo normal, a nivel de bloque
    - 10.3.5 Elementos no reemplazados, flotantes
    - 10.3.6 Elementos reemplazados, flotantes
    - 10.3.7 Elementos no reemplazados, posicionados absolutamente
    - 10.3.8 Elementos reemplazados, posicionados absolutamente
  - 10.4 Anchos mínimos y máximos: 'min-width' y 'max-width'
  - 10.5 Altura del contenido: la propiedad 'height'
  - 10.6 Computando alturas y márgenes
    - 10.6.1 Elementos no reemplazados, de línea
    - 10.6.2 Elementos reemplazados, de línea, elementos reemplazados en flujo normal, a nivel de bloque y elementos reemplazados, flotantes
    - 10.6.3 Elementos no reemplazados en flujo normal, a nivel de bloque y elementos reemplazados, flotantes
    - 10.6.4 Elementos no reemplazados, posicionados absolutamente
    - 10.6.5 Elementos reemplazados, posicionados absolutamente
  - 10.7 Alturas mínimas y máximas: 'min-height' y 'max-height'
  - 10.8 Cálculo de la altura de la línea: las propiedades 'line-height' y 'vertical-align'

- 10.8.1 Interlineado y medio interlineado
- **11 Efectos visuales**
  - 11.1 Desbordamiento y recorte
    - 11.1.1 Desbordamiento: la propiedad 'overflow'
    - 11.1.2 Recorte: la propiedad 'clip'
  - 11.2 Visibilidad: la propiedad 'visibility'
- **12 Contenido generado, numeración automática y listas**
  - 12.1 Los pseudo-elementos :before y :after
  - 12.2 La propiedad 'content'
  - 12.3 Interacción de :before y :after con los elementos 'compact' y 'run-in'
  - 12.4 Comillas
    - 12.4.1 Especificación de comillas con la propiedad 'quotes'
    - 12.4.2 Inserción de comillas con la propiedad 'content'
  - 12.5 Contadores y numeración automática
    - 12.5.1 Contadores anidados y área de alcance
    - 12.5.2 Estilos de contadores
    - 12.5.3 Contadores en elementos con 'display: none'
  - 12.6 Marcadores y listas
    - 12.6.1 Marcadores: la propiedad 'marker-offset'
    - 12.6.2 Listas: las propiedades 'list-style-type', 'list-style-image', 'list-style-position' y 'list-style'
- **13 Medios paginados**
  - 13.1 Introducción a los medios paginados
  - 13.2 Cajas de página: la regla @page
    - 13.2.1 Márgenes de la página
    - 13.2.2 Tamaño de la página: la propiedad 'size'
      - *Procesamiento de cajas de página que no se ajustan a una hoja de destino*
      - *Ubicación de la caja de página en la hoja*
    - 13.2.3 Marcas de corte: la propiedad 'marks'
    - 13.2.4 Páginas izquierda, derecha y primera
    - 13.2.5 Contenido fuera de la caja de página
  - 13.3 Saltos de página
    - 13.3.1 Saltos antes/después de los elementos: 'page-break-before', 'page-break-after', 'page-break-inside'
    - 13.3.2 Uso de páginas nominadas: 'page'
    - 13.3.3 Saltos dentro de los elementos: 'orphans', 'widows'
    - 13.3.4 Saltos de página permitidos
    - 13.3.5 Saltos de página forzados
    - 13.3.6 Los "mejores" saltos de página
  - 13.4 Cascada en el contexto de página
- **14 Colores y fondos**
  - 14.1 Color del primer plano: la propiedad 'color'
  - 14.2 El fondo
    - 14.2.1 Propiedades del fondo: 'background-color', 'background-image', 'background-repeat', 'background-attachment', 'background-position' y 'background'
  - 14.3 Corrección de la gama
- **15 Fuentes**
  - 15.1 Introducción
  - 15.2 Especificación de la fuente
    - 15.2.1 Propiedades para la especificación de las fuentes
    - 15.2.2 Familias de fuentes: la propiedad 'font-family'
    - 15.2.3 Estilos de la fuente: las propiedades 'font-style', 'font-variant', 'font-weight' y 'font-stretch'
    - 15.2.4 Tamaño de la fuente: las propiedades 'font-size' y 'font-size-adjust'
    - 15.2.5 Propiedad resumida de la fuente: la propiedad 'font'
    - 15.2.6 Familias genéricas de fuentes
      - *serif*
      - *sans-serif*
      - *cursive*
      - *fantasy*
      - *monospace*
  - 15.3 Selección de fuentes
    - 15.3.1 Descripciones de Fuentes y @font-face
    - 15.3.2 Descriptores para Seleccionar una Fuente: 'font-family', 'font-style', 'font-variant', 'font-weight', 'font-stretch' y 'font-size'
    - 15.3.3 Descriptor para la Calificación de los Datos de una Fuente: 'unicode-range'
    - 15.3.4 Descriptor para Valores Numéricos: 'units-per-em'
    - 15.3.5 Descriptor para el Referenciado: 'src'
    - 15.3.6 Descriptores para las Equivalencias: 'panose-1', 'stemv', 'stemh', 'slope', 'cap-height', 'x-height', 'ascent' y 'descent'
    - 15.3.7 Descriptores para la Síntesis: 'widths', 'bbox' y 'definition-src'
    - 15.3.8 Descriptores para la Alineación: 'baseline', 'centerline', 'mathline' y 'topline'
    - 15.3.9 Ejemplos
  - 15.4 Características de las Fuentes
    - 15.4.1 Introducción a las Características de las Fuentes
    - 15.4.2 Nombre completo de la fuente
    - 15.4.3 Unidades coordinadas en el cuadrado eme
    - 15.4.4 Línea de Base Central
    - 15.4.5 Codificación de la Fuente
    - 15.4.6 Nombre de la familia de fuentes

- 15.4.7 Ancho de los signos
    - 15.4.8 Ancho del asta horizontal
    - 15.4.9 Altura de los signos en mayúsculas
    - 15.4.10 Altura de los signos en minúsculas
    - 15.4.11 Línea de Base inferior
    - 15.4.12 Línea de Base matemática
    - 15.4.13 Máxima caja externa
    - 15.4.14 Altura máxima sin acentos
    - 15.4.15 Profundidad máxima sin acentos
    - 15.4.16 Número Panose-1
    - 15.4.17 Rango de caracteres ISO 10646
    - 15.4.18 Línea de Base superior
    - 15.4.19 Ancho del asta vertical
    - 15.4.20 Angulo del asta vertical
  - 15.5 Algoritmo de equivalencia de fuentes
    - 15.5.1 Equivalencia de los valores de los pesos de las fuentes con los nombres de las fuentes
    - 15.5.2 Ejemplos de equivalencias de fuentes
- **16 Texto**
  - 16.1 Sangría: la propiedad 'text-indent'
  - 16.2 Alineación: la propiedad 'text-align'
  - 16.3 Decoración
    - 16.3.1 Subrayado, súper-rayado, tachado y parpadeo: la propiedad 'text-decoration'
    - 16.3.2 Texto con sombra: la propiedad 'text-shadow'
  - 16.4 Espaciado de letras y palabras: las propiedades 'letter-spacing' y 'word-spacing'
  - 16.5 Cambio de minúsculas y mayúsculas: la propiedad 'text-transform'
  - 16.6 Espacios en blanco: la propiedad 'white-space'
- **17 Tablas**
  - 17.1 Introducción a las tablas
  - 17.2 El modelo de tabla CSS
    - 17.2.1 Objetos anónimos de la tabla
  - 17.3 Selectores de columnas
  - 17.4 Las tablas en el modelo de formato visual
    - 17.4.1 Posición y alineación del encabezado
  - 17.5 Composición visual del contenido de las tablas
    - 17.5.1 Capas y transparencia en la tabla
    - 17.5.2 Algoritmos para el ancho de la tabla: la propiedad 'table-layout'
      - *Composición fija de la tabla*
      - *Composición automática de la tabla*
    - 17.5.3 Algoritmos para la altura de la tabla
    - 17.5.4 Alineación horizontal en una columna
    - 17.5.5 Efectos dinámicos en filas y columnas
  - 17.6 Bordes
    - 17.6.1 El modelo de bordes separados
      - *Bordes alrededor de celdas vacías: la propiedad 'empty-cells'*
    - 17.6.2 El modelo de bordes cerrados
      - *Resolución de conflictos de bordes*
    - 17.6.3 Estilos de bordes
  - 17.7 Procesamiento sonoro de las tablas
    - 17.7.1 Lectura de títulos: la propiedad 'speak-header'
- **18 Interfaz del usuario**
  - 18.1 Cursores: la propiedad 'cursor'
  - 18.2 Preferencias de color del usuario
  - 18.3 Preferencias de fuentes del usuario
  - 18.4 Contornos dinámicos: la propiedad 'outline'
    - 18.4.1 Contornos y foco
  - 18.5 Amplificación
- **19 Hojas de estilo auditivas**
  - 19.1 Introducción a las hojas de estilo auditivas
  - 19.2 Propiedades del volumen: 'volume'
  - 19.3 Propiedades del habla: 'speak'
  - 19.4 Propiedades de la pausa: 'pause-before', 'pause-after' y 'pause'
  - 19.5 Propiedades de la señal: 'cue-before', 'cue-after' y 'cue'
  - 19.6 Propiedades de la mezcla: 'play-during'
  - 19.7 Propiedades espaciales: 'azimuth' y 'elevation'
  - 19.8 Propiedades de las características de la voz: 'speech-rate', 'voice-family', 'pitch', 'pitch-range', 'stress' y 'richness'
  - 19.9 Propiedades del discurso: 'speak-punctuation' y 'speak-numeral'
- **Apéndice A: Una hoja de estilo de ejemplo para HTML 4.0**
- **Apéndice B: Cambios con respecto a CSS1**
  - B.1 Nuevo funcionamiento
  - B.2 Descripciones actualizadas
  - B.3 Cambios semánticos a CSS1
- **Apéndice C: Implementación y notas de desempeño para fuentes**
  - C.1 Glosario de términos relacionados con las fuentes
  - C.2 Recuperación de las fuentes
  - C.3 Significado de los números Panose
  - C.4 Deducción de rangos de Unicode para TrueType
  - C.5 Generación automática de descriptores
- **Apéndice D: La gramática de CSS2**

- [D.1 Gramática](#)
  - [D.2 Scanner de léxico](#)
  - [D.3 Comparación de los comandos en CSS2 y CSS1](#)
- **[Apéndice E. Referencias](#)**
  - [E.1 Referencias normativas](#)
  - [E.2 Referencias informativas](#)
- **[Apéndice F. Índice de propiedades](#)**
- **[Apéndice G. Índice de descriptores](#)**
- **[Apéndice H. Índice](#)**

Copyright © 1998 W3C ([MIT](#), [INRIA](#), [Keio](#)), All Rights Reserved.



# 1 Acerca de la Especificación CSS2

## 1.1 Leyendo la especificación

Esta especificación se ha escrito con dos tipos de lectores en mente: autores de CSS e implementadores de CSS. Esperamos que la especificación provea a los autores de las herramientas que necesitan para componer documentos eficaces, atractivos y accesibles, sin agobiarlos con detalles acerca de la implementación de las CSS. Los implementadores, sin embargo, deben encontrar todo lo necesario para desarrollar [aplicaciones del usuario con conformidad](#). La especificación comienza con una presentación general de las CSS y luego se hace más técnica y específica. Para el rápido acceso a la información, un tabla general de contenidos, tablas de contenidos específicas al comienzo de cada sección y un índice facilitan la búsqueda, tanto en la versión electrónica como en la impresa.

La especificación se ha escrito con dos modos de presentación en mente: electrónica e impresa. Aunque ambas presentaciones serán, sin duda, similares, los lectores encontrarán algunas diferencias. Por ejemplo, los vínculos no funcionarán en la versión impresa (obviamente) y los números de las páginas no aparecerán en la versión electrónica. En caso de discrepancia, la versión electrónica es considerada la versión dominante.

## 1.2 Cómo está organizada la especificación

La especificación está organizada en las siguientes secciones:

### Sección 2: Una introducción a CSS2

La introducción incluye un breve manual de instrucciones de CSS2 y una discusión de los fundamentos de diseño que se encuentran detrás de CSS2.

### Secciones 3 - 20: Manual de referencia de CSS2

El cuerpo principal del manual de referencia consiste en la referencia del lenguaje CSS2. Esta referencia define qué puede incluirse en una hoja de estilo CSS2 (sintaxis, propiedades, valores) y cómo las aplicaciones del usuario deben interpretar estas hojas de estilo para aspirar a su [conformidad](#).

### Apéndices:

Los apéndices contienen información sobre [un ejemplo de hoja de estilo para HTML 4.0](#), [cambios efectuados sobre CSS1](#), [implementación y notas sobre el funcionamiento](#), [la gramática de CSS2](#), una lista de [referencias](#) normativas e informativas, y tres índices: uno para las [propiedades](#), uno para los [descriptores](#), y un [índice general](#).

## 1.3 Convenciones

### 1.3.1 Elementos y atributos en el [lenguaje del documento](#)

- Las propiedades CSS, los descriptores y los nombres de pseudo-clase son delimitados por comillas simples.
- Los valores CSS son delimitados por comillas simples.
- Los nombres de los elementos del lenguaje del documento están en mayúsculas.
- Los nombre de los atributos del lenguaje del documento están en minúsculas y delimitados por comillas dobles.

### 1.3.2 Definición de las propiedades CSS

Cada definición de una propiedad CSS comienza con un resumen de la información clave que se asemeja a lo siguiente:

#### 'nombre\_de\_la\_propiedad'

*Valor:* valores permitidos y sintaxis  
*Inicial:* valor inicial  
*Se aplica a:* elementos a los cuáles se aplica esta propiedad  
*Se hereda:* si la propiedad es heredada  
*Porcentajes:* cómo se interpretan los valores de porcentaje  
*Medios:* a qué grupo de medios se aplica la propiedad

#### Valor

Esta parte especifica el conjunto de valores válidos para la propiedad. Los tipos de valor pueden ser designados de varias maneras:

1. palabras clave (ej., auto, disc, etc.)
2. datos básicos, que aparecen entre "<" y ">" (ej., <medida>, <porcentaje>, etc.). En la versión electrónica del documento, cada instancia de un dato básico contiene un vínculo a su definición.
3. tipos que tienen el mismo rango de valores que una propiedad con igual nombre (ej., <'border-width'> <'background-attachment'>, etc.). En este caso, el nombre del tipo es el nombre de la propiedad (completo con comillas) entre "<" y ">" (ej., <'border-width'>). En la versión electrónica del documento, cada instancia de este tipo no-terminal es un vínculo a la correspondiente definición de la propiedad.



4. no-terminales que no comparten el mismo nombre que una propiedad. En este caso, el nombre no-terminal aparece entre "<" y ">", como en <border-width>. Note la diferencia entre <border-width> y <'border-width'>; el último se define en términos del anterior. La definición de un no-terminal se encuentra en su primera aparición dentro de la especificación. En la versión electrónica del documento, cada instancia de este tipo de valor es un vínculo a la correspondiente definición del valor.

**Nota de la traducción:**

Para evitar el desconcierto de quienes (como yo) no dominan los lenguajes de programación, aclaro que la denominación *no-terminal* (en inglés, non-terminal) está relacionada con el proceso de entrada de datos en un programa y las funciones de reconocimiento de esos datos.

Otras palabras en estas definiciones son palabras clave que deben aparecer literalmente, sin comillas (ej., red). La barra inclinada (/) y la coma (,) también deben aparecer literalmente.

Los valores pueden estar organizados de las siguientes maneras:

- Varias palabras yuxtapuestas significa que todas ellas deben aparecer, en el orden dado.
- Una barra (|) separa dos o más alternativas: exactamente una de ellas debe aparecer.
- Una barra doble (||) separa dos o más opciones: una o más de ellas debe aparecer, en cualquier orden.
- Los corchetes ([ ]) son para agrupar.

La yuxtaposición es más fuerte que la barra doble y la barra doble es más fuerte que la barra. De este modo, las siguientes líneas son equivalentes:

a b | c || d e  
[ a b ] | [ c || [ d e ] ]

Cada tipo, palabra clave o grupo encerrado entre corchetes puede ser seguido por uno de los siguientes modificadores:

- Un asterisco (\*) indica que el tipo, palabra o grupo precedente aparece cero o más veces.
- Una signo más (+) indica que el tipo, palabra o grupo precedente aparece una o más veces.
- Un signo de interrogación (?) indica que el tipo, palabra o grupo precedente es opcional.
- Un par de números entre llaves ({A,B}) indica que el tipo, palabra o grupo precedente aparece por lo menos A y a lo sumo B veces.

Los siguientes ejemplos ilustran los diferentes tipos de valor:

Valor: N | NW | NE  
Valor: [ <medida> | thick | thin ]{1,4}  
Valor: [<family-name> , ]\* <family-name>  
Valor: <uri>? <color> [ / <color> ]?  
Valor: <uri> || <color>

**Inicial**

Esta parte especifica el valor inicial de la propiedad. Si la propiedad se hereda, éste es el valor que se ha dado al elemento raíz de la [estructura del documento](#). Por favor, consulte la sección sobre el concepto de [cascada](#) para obtener información acerca de la interacción entre valores especificados en las hojas de estilo, valores heredados y valores iniciales.

**Se aplica a**

Esta parte registra los elementos a los cuales se aplica la propiedad. Todos los elementos son considerados aptos para tener todas las propiedades, pero algunas propiedades no tienen ningún efecto sobre el procesamiento de algunos tipos de elementos. Por ejemplo, 'white-space' sólo afecta a los elementos a nivel de bloque.

**Se hereda**

Esta parte indica si el valor de la propiedad se hereda de un elemento antepasado. Por favor, consulte la sección sobre el concepto de [cascada](#) para obtener información sobre la interacción entre valores especificados en las hojas de estilo, valores heredados y valores iniciales.

**Porcentajes**

Esta parte indica cómo deben ser interpretados los porcentajes cuando aparecen en el valor de la propiedad. Si aquí aparece "N/A", significa que la propiedad no acepta porcentajes como valores.

## Grupos de medios

Esta parte indica los [grupos de medios](#) a los cuales se aplica la propiedad. La condición de [conformidad](#) establece que las aplicaciones del usuario deben soportar esta propiedad en el caso de que soporten el procesamiento de los [tipos de medio](#) incluidos en estos [grupos de medios](#).

### 1.3.3 Propiedades resumidas

Algunas propiedades son fórmulas *resumidas*, significando que permiten a los autores especificar los valores de varias propiedades con una sola propiedad.

Por ejemplo, la propiedad 'font' es una propiedad resumida para definir 'font-style', 'font-variant', 'font-weight', 'font-size', 'line-height' y 'font-family', todas de una sola vez.

Cuando algunos valores son omitidos en la fórmula resumida, a cada propiedad "ausente" le es asignado su valor inicial (ver la sección sobre [cascada](#)).

Las múltiples reglas de estilo de este ejemplo:

```
H1 {
  font-weight: bold;
  font-size: 12pt;
  line-height: 14pt;
  font-family: Helvetica;
  font-variant: normal;
  font-style: normal;
  font-stretch: normal;
  font-size-adjust: none
}
```

pueden re-escribirse con una sola propiedad resumida:

```
H1 { font: bold 12pt/14pt Helvetica }
```

En este ejemplo, 'font-variant', 'font-stretch', 'font-size-adjust' y 'font-style' toman sus valores iniciales.

### 1.3.4 Notas y ejemplos

Todos los ejemplos que ilustran usos no permitidos están claramente señados como "EJEMPLO ILICITO".

Todos los ejemplos de HTML están ajustados a la especificación HTML 4.0 strict DTD (definida en [\[HTML40\]](#)) a menos que se indique otra a través de una declaración del tipo de documento.

Todas las notas son sólo informativas.

Los ejemplos y notas están marcados dentro del código HTML para la especificación y las aplicaciones del usuario CSS1 los procesarán de modo especial.

### 1.3.5 Imágenes y descripciones extensas

La mayoría de las imágenes en la versión electrónica de esta especificación están acompañadas por "descripciones extensas" de lo que representan. Un vínculo a la descripción extensa es señalado por una "[D]" a la derecha de la imagen.

Las imágenes y las descripciones extensas sólo son informativas.

## 1.4 Reconocimientos

Esta especificación es producto del Grupo de Trabajo del W3C en Hojas de Estilo en Cascada y Propiedades de Formateo. Además de los editores de esta especificación, los miembros del Grupo del Trabajo son: Brad Chase (Bitstream), Chris Wilson (Microsoft), Daniel Glazman (Electricité de France), Dave Raggett (W3C/HP), Ed Tecot (Microsoft), Jared Sorensen (Novell), Lauren Wood (SoftQuad), Laurie Anna Kaplan (Microsoft), Mike Wexler (Adobe), Murray Maloney (Grif), Powell Smith (IBM), Robert Stevahn (HP), Steve Byrne (JavaSoft), Steven Pemberton (CWI), Thom Phillabaum (Netscape), Douglas Rand (Silicon Graphics), Robert Pernet (Lotus), Dwayne Dicks (SoftQuad) y Sho Kuwamoto (Macromedia). Agradecemos a ellos sus continuos esfuerzos.

Varios expertos invitados al Grupo de Trabajo han contribuido: George Kersher, Glenn Rippel (Bitstream), Jeff Veen (HotWired), Markku T. Hakkinen (The Productivity Works), Martin Dürst (W3C, formerly Universität Zürich), Roy Platon (RAL), Todd Fahrner (Verso), Tim Boland (NIST), Eric Meyer (Case Western Reserve University) y Vincent Quint (W3C).

La sección de fuentes de la Web fue producida principalmente por Brad Chase (Bitstream) David Meltzer (Microsoft Typography) y Steve Zilles (Adobe). Las siguientes personas también han contribuido de distintas maneras a la sección perteneciente a fuentes: Alex Beamon (Apple), Ashok Saxena (Adobe), Ben Bauermeister (HP), Dave Raggett (W3C/HP), David Opstad (Apple), David Goldsmith (Apple), Ed Tecot (Microsoft), Erik van Blokland (LettError), François Yergeau (Alis), Gavin Nicol (Inso), Herbert van Zijl (Elsevier), Liam Quin, Misha Wolf (Reuters), Paul Haeberli (SGI) y el rezagado Phil Karlton (Netscape).

La sección sobre medios paginados fue redactada en gran parte por Robert Stevahn (HP) y Stephen Waters (Microsoft).

Robert Stevahn (HP), Scott Furman (Netscape) y Scott Isaacs (Microsoft) contribuyeron en el tipado de Posicionamiento CSS.

Mike Wexler (Adobe) fue el redactor del proyecto interno, el cual esbozó muchas de las nuevas características de CSS2.

[T.V. Raman](#) (Adobe) hizo contribuciones esenciales con relación a las Hojas de Estilo en Cascada Auditivas (ACSS) y a los conceptos de presentación auditiva basado en su trabajo en AsTeR (Audio System For Technical Readings). Ha contribuido con un proyecto inicial de la especificación ACSS que formó la actual especificación. Los valores para las propiedades auditivas en el [ejemplo de hoja de estilo para HTML 4.0](#) son de su invención; él los usa actualmente de modo cotidiano en su computadora de audio junto con Emacspeak y el navegador W3 Emacs (creado por William Perry, quien también implementó las extensiones auditivas por parte de W3).

Todd Fahrner (Verso) investigó los navegadores históricos y contemporáneos para desarrollar el ejemplo de hoja de estilo en el apéndice.

Gracias a Jan Kärrman, autor de [html2ps](#) por su gran ayuda creando la versión PostScript de esta especificación.

A través de los encuentros electrónicos y físicos, las siguientes personas han contribuido al desarrollo de CSS2: Alan Borning, Robert Cailliau, Liz Castro, James Clark, Dan Connolly, Donna Converse, Daniel Dardailler, Al Gilman, Daniel Greene, Scott Isaacs, Geir Ivarsøy, Vincent Mallet, Kim Marriott, Brian Michalowski, Lou Montulli, Henrik Frystyk Nielsen, Jacob Nielsen, Eva von Pepel, William Perry, David Siegel, Peter Stuckey y Jason White.

Las discusiones en [www-style@w3.org](mailto:www-style@w3.org) han influido en muchos temas claves para CSS. Especialmente, nos gustaría agradecer a Bjorn Backlund, Todd Fahrner, Lars Marius Garshol, Sue Jordan, Ian Hickson, Susan Lesch, Andrew Marshall, MegaZone, Eric Meyer, Russell O'Connor, David Perrell, Liam Quinn, Jon Seymour, Neil St. Laurent, Taylor, Brian Wilson y Chris Wilson por su participación.

Muchas gracias al Web Accessibility Initiative Protocols and Formats Technical Review Working Group (WAI PF) por ayudar a mejorar la accesibilidad de CSS2.

Muchas gracias a Philippe Le Hégarret, cuyo validador CSS contribuyó a asegurar ejemplos correctos y una gramática rigurosa.

Un especial agradecimiento a Arnaud Le Hors, cuya contribución técnica hace que este documento funcione.

Adam Costello mejoró esta especificación realizando una revisión detallada.

Por último, gracias a Tim Berners-Lee sin quien nada de esto habría sido posible.

## 1.5 Aviso de Copyright

**Copyright © 1997 [World Wide Web Consortium](#), ([Massachusetts Institute of Technology](#), [Institut National de Recherche en Informatique et en Automatique](#), [Keio University](#)). Todos los derechos reservados.**

Los documentos del sitio del [W3C](#) son proporcionados por los propietarios del copyright bajo la siguiente licencia. Obteniendo, usando y/o copiando este documento, o el documento del W3C del que esta declaración está vinculada, usted acuerda que ha leído, entendido y cumplirá con los siguiente términos y condiciones:

El permiso para usar, copiar y distribuir el contenido de este documento, o del documento del W3C del que esta declaración está vinculada, por cualquier medio, para cualquier propósito y sin cargo o retribución es concedido por este medio, con la condición de incluir lo siguiente en *TODAS* las copias del documento, o cualquiera de sus partes, que se usen:

1. Un vínculo o URI al documento original del W3C.
2. El anterior aviso de copyright del autor original, si no existe, un aviso con esta fórmula: "[Copyright © World Wide Web Consortium](#), ([Massachusetts Institute of Technology](#), [Institut National de Recherche en Informatique et en Automatique](#), [Keio University](#)). Todos los derechos reservados."
3. *Si existe*, el ESTADO del documento del W3C.

Cuando el espacio lo permita, se debe proporcionar el texto completo de este **AVISO**. Además, se deberá otorgar crédito a los propietarios del copyright por cualquier programa, documento u otros artículos o productos que se elaboren de conformidad con la implementación del contenido de este documento o cualquier parte del mismo.

No se otorga ningún derecho para crear modificaciones o derivados de acuerdo con esta licencia.

**ESTE DOCUMENTO SE ENTREGA "TAL CUAL ESTÁ" Y LOS PROPIETARIOS DEL COPYRIGHT NO OTORGAN REPRESENTACIONES O GARANTÍAS, EXPRESAS O IMPLÍCITAS, INCLUYENDO, PERO SIN LIMITARSE A, GARANTÍAS DE COMERCIALIZACIÓN, DE IDONEIDAD PARA DETERMINADO PROPÓSITO, DE NO CONTRAVENCIÓN O TÍTULO; QUE EL CONTENIDO DEL DOCUMENTO ES ADECUADO A CUALQUIER PROPÓSITO; NI QUE LA IMPLEMENTACIÓN DE TALES CONTENIDOS NO INFRINGEN NINGUNA PATENTE DE TERCEROS, PROPIEDAD INTELECTUAL, MARCAS REGISTRADAS U OTROS DERECHOS.**

**LOS PROPIETARIOS DEL COPYRIGHT NO SERÁN RESPONSABLES POR NINGÚN DAÑO DIRECTO, INDIRECTO, ESPECIAL O DERIVADO DE CUALQUIER USO DE ESTE DOCUMENTO O EL FUNCIONAMIENTO O IMPLEMENTACIÓN DEL CONTENIDO DEL MISMO.**

El nombre y la marca registrada de los propietarios del copyright NO pueden ser usados en publicidades o anuncios relacionados a este documento o su contenido sin una autorización previa explícita y por escrito. El derecho al copyright de este documento sigue siendo en todo momento de los propietarios del copyright.

# 2 Introducción a CSS2

## 2.1 Breve guía de CSS2 para HTML

En esta guía, le mostraremos cuán fácil puede ser diseñar unas simples hojas de estilo. Para esta guía, necesitará saber un poco del código HTML (ver [\[HTML40\]](#)) y alguna terminología básica de composición gráfica.

Comenzamos con un pequeño documento HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>La página de Bach</TITLE>
  </HEAD>
  <BODY>
    <H1>La página de Bach</H1>
    <P>Johann Sebastian Bach fue un prolífico compositor.
  </BODY>
</HTML>
```

Para establecer el color azul para el texto de los elementos H1, puede escribir la siguiente regla CSS:

```
H1 { color: blue }
```

Una regla CSS consta de dos partes principales: un **selector** ('H1') y una declaración ('color: blue'). La declaración tiene dos partes: una propiedad ('color') y un valor ('blue'). Con tal que el ejemplo anterior intente influir en una sola propiedad necesaria para el procesamiento de un documento HTML, ya lo califica como una hoja de estilo en sí mismo. Combinado con otras hojas de estilo (un rasgo fundamental de CSS es que las hojas de estilo se combinan) determinará la presentación final del documento.

La especificación HTML 4.0 define de qué manera las reglas de las hojas de estilo pueden especificarse para los documentos HTML: ya sea dentro del documento HTML o a través de una hoja de estilo externa. Para poner la hoja de estilo dentro del documento, use el elemento STYLE:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>La página de Bach</TITLE>
    <STYLE type="text/css">
      H1 { color: blue }
    </STYLE>
  </HEAD>
  <BODY>
    <H1>La página de Bach</H1>
    <P>Johann Sebastian Bach fue un prolífico compositor.
  </BODY>
</HTML>
```

Para una mayor flexibilidad, recomendamos que los autores especifiquen hojas de estilo externas; ellas pueden cambiarse sin modificar el documento fuente HTML y pueden compartirse entre varios documentos. Para vincular una hoja de estilo externa puede usar el elemento LINK:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>La página de Bach</TITLE>
    <LINK rel="stylesheet" href="bach.css" type="text/css">
  </HEAD>
  <BODY>
    <H1>La página de Bach</H1>
    <P>Johann Sebastian Bach fue un prolífico compositor.
  </BODY>
</HTML>
```

El elemento LINK especifica:

- el tipo de vínculo: a una hoja de estilo ("stylesheet").
- la ubicación de la hoja de estilo a través del atributo "href".

- el tipo de hoja de estilo que se vincula: "text/css".

Para mostrar la íntima relación entre una hoja de estilo y el sistema estructurado de marcas, continuamos usando el elemento STYLE en esta guía. Agreguemos más colores:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>La página de Bach</TITLE>
    <STYLE type="text/css">
      BODY { color: red }
      H1 { color: blue }
    </STYLE>
  </HEAD>
  <BODY>
    <H1>La página de Bach</H1>
    <P>Johann Sebastian Bach fue un prolífico compositor.
  </BODY>
</HTML>
```

La hoja de estilo contiene ahora dos reglas: la primera establece el color del elemento BODY como 'red' , en tanto la segunda coloca el color del elemento H1 en 'blue'. Como no se ha especificado ningún valor para el color del elemento P, éste heredará el color de su elemento padre, es decir, BODY. El elemento H1 también es un elemento hijo de BODY pero la segunda regla reemplaza el valor heredado. En CSS existen a menudo tales conflictos entre diferentes valores y esta especificación describe cómo resolverlos.

CSS2 tiene más de 100 propiedades diferentes, incluyendo 'color'. Veamos algunos otros:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>La página de Bach</TITLE>
    <STYLE type="text/css">
      BODY {
        font-family: "Gill Sans", sans-serif;
        font-size: 12pt;
        margin: 3em;
      }
    </STYLE>
  </HEAD>
  <BODY>
    <H1>La página de Bach</H1>
    <P>Johann Sebastian Bach fue un prolífico compositor.
  </BODY>
</HTML>
```

Lo primero que se puede observar es que varias declaraciones se agrupan en un bloque encerrado por llaves ({...}) y separado por un punto y coma, no obstante la última declaración también debe estar seguida por un punto y coma.

La primera declaración en el elemento BODY establece la familia de fuentes "Gill Sans". Si esa fuente no está disponible, la aplicación del usuario (a menudo llamada "browser" o navegador) usará la familia 'sans-serif' que es una de las cinco familias de fuentes genéricas que todas las aplicaciones del usuario reconocen. Los elementos hijos de BODY heredarán el valor de la propiedad 'font-family'.

La segunda declaración establece el tamaño de la fuente del elemento BODY en 12 puntos. La unidad de medida "punto" es normalmente usada en la tipografía impresa para indicar el tamaño de la fuente y otras dimensiones. Es un ejemplo de unidad de medida absoluta que no varía según el entorno.

La tercer declaración usa una unidad de medida relativa que varía de acuerdo a las circunstancias. La unidad "eme" está relacionada con el tamaño de la fuente del elemento. En este caso el resultado es que el margen alrededor del elemento BODY es tres veces mayor que el tamaño de la fuente.

## 2.2 Breve guía de CSS2 para XML

CSS puede usarse con cualquier formato de documento estructurado, por ejemplo con aplicaciones del eXtensible Markup Language [XML10]. De hecho, XML depende más de las hojas de estilo que HTML puesto que los autores pueden construir elementos propios que las aplicaciones del usuario no saben como mostrar.

Este es un simple fragmento en XML:

```
<ARTICULO>
  <TITULO>Federico el Grande se encuentra con Bach</TITULO>
  <AUTOR>Johann Nikolaus Forkel</AUTOR>
  <PARRAFO>
    Una tarde, justo cuando preparaba su <INSTRUMENTO>flauta</INSTRUMENTO>
    y sus músicos estaban dispuestos, un oficial le alcanzó una lista
    de los forasteros que habían arribado.
  </PARRAFO>
</ARTICULO>
```

Para que este fragmento sea visualizado a modo de un documento, debemos declarar primero cuáles son los elementos a nivel de línea (es decir, que no dan lugar a saltos de líneas) y cuáles son a nivel de bloque (es decir, que producen saltos de línea).

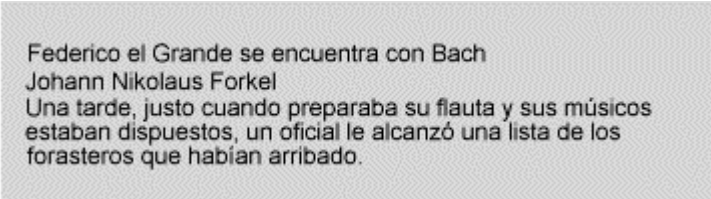
```
INSTRUMENTO { display: inline }
ARTICULO, TITULO, AUTOR, PARRAFO { display: block }
```

La primer regla declara que INSTRUMENTO será a nivel de línea y la segunda regla, con su lista de selectores separados por comas, declara que todos los demás elementos serán a nivel de bloque.

Una de las ideas de vincular una hoja de estilo a un documento XML es emplear una instrucción de procesamiento:

```
<?XML:stylesheet type="text/css" href="bach.css"?>
<ARTICULO>
  <TITULO>Federico el Grande se encuentra con Bach</TITULO>
  <AUTOR>Johann Nikolaus Forkel</AUTHR>
  <PARRAFO>
    Una tarde, justo cuando preparaba su <INSTRUMENTO>flauta</INSTRUMENTO>
    y sus músicos estaban dispuestos, un oficial le alcanzó una lista
    de los forasteros que habían arribado.
  </PARRAFO>
</ARTICULO>
```

Una aplicación del usuario visual dará este formato al ejemplo anterior:



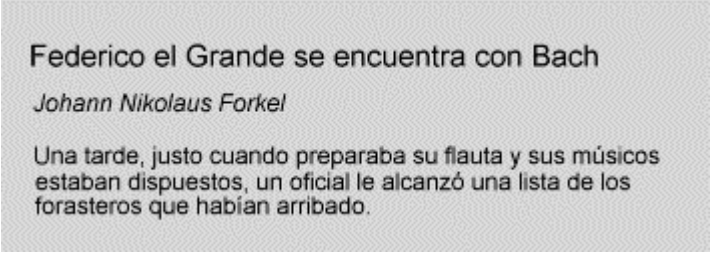
[D]

Observe que la palabra "flauta" permanece dentro del párrafo debido a que es el contenido del elemento a nivel de línea INSTRUMENTO.

Pero aún el texto no tiene el formato que se podría esperar. Por ejemplo, el tamaño de la fuente del título debería ser mayor que la del resto del texto y quizás se prefiera mostrar el nombre del autor en itálica:

```
INSTRUMENTO { display: inline }
ARTICULO, TITULO, AUTOR, PARRAFO { display: block }
TITULO { font-size: 1.3em }
AUTOR { font-style: italic }
ARTICULO, TITULO, AUTOR, PARRAFO { margin: 0.5em }
```

Una aplicación del usuario visual dará este formato al ejemplo anterior:



[D]



Agregando más reglas a la hoja de estilo le permitirá mejorar aún más la presentación del documento.

## 2.3 El modelo de procesamiento de CSS2

Esta sección presenta un posible modelo de cómo trabajan las aplicaciones del usuario que soportan CSS. Este es sólo un modelo conceptual; las implementaciones reales pueden variar.

En este modelo, una aplicación del usuario procesa una fuente atravesando los siguientes pasos:

1. Analiza el documento fuente y crea una [estructura del documento](#).
2. Identifica el [tipo de medio](#) al que está dirigido.
3. Recupera todas las hojas de estilo asociadas con el documento que se especifican para el [tipo de medio](#) al que está dirigido.
4. Toma nota de cada elemento de la estructura del documento, asignando un valor individual a cada [propiedad](#) que es aplicable al [tipo de medio](#) al que está dirigido. Los valores de las propiedades se asignan de acuerdo a los mecanismos descritos en la sección de [cascada y herencia](#).

Una parte del cálculo de los valores depende del algoritmo de procesamiento apropiado para el [tipo de medio](#) al que está dirigido. Por ejemplo, si el medio al que se dirige es la pantalla, las aplicaciones del usuario aplican el [modelo de composición visual](#). Si el destino es la página impresa, las aplicaciones del usuario aplican el [modelo de página](#). Si el medio de destino es un dispositivo de procesamiento sonoro (ej., un sintetizador de palabras), las aplicaciones del usuario aplican el [modelo de procesamiento auditivo](#).

5. De la estructura del documento registrada, genera una *estructura del formato*. A menudo, la estructura del formato se parece mucho a la estructura del documento, pero también puede diferir en forma significativa, especialmente cuando los autores hacen uso de pseudo-elementos y contenidos generados. Primero, la estructura del formato no necesita ser en forma de "árbol" --la naturaleza de la estructura depende de la implementación--. Segundo, la estructura del formato puede contener mayor o menor información que la estructura del documento. Por ejemplo, si un elemento en la estructura del documento tiene un valor de 'none' (nada) para la propiedad 'display', ese elemento no generará nada en la estructura del formato. Un elemento de lista, por otro lado, puede generar más información en la estructura del formato: el contenido del elemento y la información del estilo de lista (ej., la imagen de la viñeta).

Observe que las aplicaciones del usuario CSS no alteran la estructura del documento durante esta fase. En particular, el contenido generado por causa de las hojas de estilo no es enviado nuevamente al procesador del lenguaje del documento (ej., para un nuevo análisis).

6. Transfiere la estructura del formato al medio al que está dirigido (ej., imprime el resultado, lo muestra en pantalla, lo procesa como palabra, etc.).

El paso 1 está fuera del alcance de esta especificación (ver, por ejemplo, [\[DOM\]](#)).

Los pasos 2-5 son tratados en el cuerpo principal de esta especificación.

El paso 6 está fuera del alcance de esta especificación.

### 2.3.1 El lienzo

En todos los medios, el término *lienzo* describe "el espacio donde la estructura del formato es procesada". El lienzo es infinito por cada dimensión del espacio, pero el procesamiento generalmente tiene lugar en una región limitada del lienzo, establecida por la aplicación del usuario de acuerdo con el medio al que está dirigido. Por ejemplo, las aplicaciones del usuario que procesan para pantalla, generalmente imponen un ancho mínimo y eligen un ancho inicial basado en las dimensiones del [acceso visual](#). Las aplicaciones del usuario que procesan una página generalmente imponen restricciones en el alto y el ancho. Las aplicaciones auditivas pueden imponer límites en el terreno del audio, pero no en el tiempo.

**Nota de la traducción:**

La traducción de *lienzo* es literal del inglés "*canvas*". Por tratarse de un uso metafórico del término, no debe buscarse una relación muy estricta con la idea que describe.

### 2.3.2 Modelo de destinatarios de CSS2

Los [selectores](#) CSS2 y las propiedades permiten a las hojas de estilo referirse a las siguientes partes de un documento o aplicación del usuario:

- Los elementos de la estructura del documento y ciertas relaciones entre ellos (ver la sección de [selectores](#)).
- Los atributos de los elementos de la estructura del documento y los valores de esos atributos (ver la sección de [selectores de atributo](#)).
- Algunas partes del contenido de un elemento (ver los pseudo-elementos [:first-line](#) y [:first-letter](#)).
- Los elementos de la estructura del documento cuando se encuentran en cierto estado (ver la sección de [pseudo-clases](#)).
- Algunos aspectos del [lienzo](#) en el cual el documento será procesado.

- Alguna información del sistema (ver la sección de [interfaz del usuario](#)).

## 2.4 Reglas de diseño de CSS

CSS2, como fue antes CSS1, se basa en una serie de reglas de diseño:

- **Compatibilidad hacia atrás y hacia adelante.** Las aplicaciones del usuario CSS2 serán capaces de entender las hojas de estilo CSS1. Las aplicaciones del usuario CSS1 podrán leer las hojas de estilo CSS2 y descartar las partes que no entienden. Además, las aplicaciones del usuario que no soporten CSS serán capaces de mostrar los documentos estilísticamente mejorados. Por supuesto, los efectos estilísticos hechos posible por CSS no serán procesados, pero todo el contenido será presentado.
- **Complementariedad con documentos estructurados.** Las hojas de estilo complementan los documentos estructurados (ej., HTML y aplicaciones XML) proveyendo información estilística del texto marcado. Debe ser fácil cambiar la hoja de estilo con poco o ningún impacto en el sistema de marcas.
- **Independencia del vendedor, la plataforma y el dispositivo.** Las hojas de estilo permiten a los documentos permanecer independientes del vendedor, la plataforma y el dispositivo. Las mismas hojas de estilo son también independientes del vendedor y la plataforma, pero CSS2 permite dirigir una hoja de estilo a un grupo de dispositivos (ej., impresoras).
- **Mantenibilidad.** Apuntando a una hoja de estilo desde los documentos, los responsables de los sitios en la Web pueden simplificar el mantenimiento y conservar un estilo y un efecto consistente a todo lo largo del sitio. Por ejemplo, si el color del fondo de las páginas de una organización cambia, sólo un archivo necesita ser cambiado.
- **Simplicidad.** CSS2 es más compleja que CSS1, pero sigue siendo un lenguaje de estilo simple que es humanamente legible y posible de escribir. Las propiedades CSS se mantienen independientes unas de otras en la medida de lo posible y generalmente sólo hay un modo de conseguir un efecto determinado.
- **Rendimiento de la red.** CSS proporciona una compacta codificación para presentar los contenidos. Comparado con los archivos de imágenes o de audio que son usados frecuentemente por los autores para conseguir ciertos efectos en el procesamiento, las hojas de estilo, la mayoría de las veces, disminuyen el tamaño del contenido. Además, menos conexiones de la red tienen que ser abiertas, lo cual ayuda a incrementar el rendimiento de la red.
- **Flexibilidad.** Las CSS pueden ser aplicadas al contenido de varias maneras. La característica clave es la capacidad de formar una cascada de estilos con la información especificada en la hoja de estilo predeterminada (aplicación del usuario), las hojas de estilo del usuario, las hojas de estilo vinculadas, el encabezamiento del documento y en los atributos de los elementos que forman el cuerpo del documento.
- **Riqueza.** Proporcionando a los autores un abundante juego de efectos de procesamiento, aumenta la riqueza de la Web como medio de expresión. Los diseñadores han estado ambicionando la funcionalidad comúnmente encontrada en los programas de edición y de presentaciones gráficas. Algunos de los efectos requeridos entran en conflicto con la independencia del dispositivo, pero CSS2 llega muy lejos satisfaciendo las demandas de los diseñadores.
- **Combinación con lenguajes alternativos.** El juego de propiedades de CSS descritas en esta especificación conforman un sólido modelo de aplicación de formatos para presentaciones visuales y auditivas. Este modelo puede ser accedido mediante el lenguaje CSS, pero la combinación con otros lenguajes también es posible. Por ejemplo, un programa en JavaScript puede cambiar dinámicamente el valor de la propiedad 'color' de un determinado elemento.
- **Accesibilidad.** Varias características de CSS harán a la Web más accesible para los usuarios con discapacidades:
  - Las propiedades para controlar la apariencia de las fuentes permiten a los autores eliminar las inaccesibles imágenes de texto renderizado.
  - Las propiedades de posicionamiento permite a los autores eliminar los artilugios con el sistema de marcas (ej., imágenes invisibles) para forzar la composición.
  - La semántica de las reglas `!important` indica que los usuarios con particulares requerimientos de presentación pueden suplantar las hojas de estilo del autor.
  - El nuevo valor 'inherit' (heredado) para todas las propiedades mejora la generalidad del funcionamiento en cascada y permite una más fácil y consistente sintonía en el estilo.
  - El avanzado soporte de medios, incluyendo grupos de medios y los tipos de medios braille, de relieve y tty permitirán a los usuarios y autores confeccionar páginas para esos dispositivos.
  - Las propiedades auditivas ofrecen control sobre la salida de voz y audio.
  - Los selectores de atributos, la función 'attr()' y la propiedad 'content' brindan acceso al contenido alternativo.
  - Los contadores y la numeración de secciones y párrafos pueden mejorar la navegabilidad del documento y economizar espacio de sangrado (importante en los dispositivos braille). Las propiedades 'word-spacing' y 'text-indent' también eliminan la necesidad de usar espacios en blanco extras en el documento.

**Nota.** Para mayor información acerca del diseño de documentos accesibles usando CSS y HTML, por favor, consulte [\[WAI-PAGEAUTH\]](#).

# 3 Conformidad: Requisitos y Recomendaciones

## 3.1 Definiciones

En esta sección, iniciamos la especificación formal de CSS2, comenzando con el contrato entre autores, usuarios e implementadores.

Las palabras clave ("DEBE"), ("NO DEBE"), ("REQUERIDO"), ("DEBERÍA"), ("NO DEBERÍA"), ("DEBERÁ"), ("NO DEBERÁ"), ("RECOMENDADO"), ("PUEDE") y ("OPCIONAL") en este documento serán interpretadas tal como se describe en RFC 2119 (ver [RFC2119](#)). Sin embargo, por razones de legibilidad, estas palabras no aparecen en mayúsculas en esta especificación.

### Nota de la traducción:

Muy sucintamente, la interpretación que debe hacerse de los términos mencionados anteriormente es: DEBE/REQUERIDO: Significan que el ítem es un requerimiento absoluto de la especificación.

DEBERÍA/DEBERÁ/RECOMENDADO: Significan que pueden existir razones válidas en circunstancias particulares para ignorar este ítem.

PUEDE/OPCIONAL: Significan que el ítem es totalmente opcional.

*Para cumplir estrictamente con los niveles de requerimiento, se recuerda que la única versión normativa es <http://www.w3.org/TR/1998/REC-CSS2-19980512> y que esta traducción puede contener errores.*

A veces, esta especificación recomienda algunas prácticas adecuadas para los autores y las aplicaciones del usuario. Estas recomendaciones no son normativas y la conformidad con esta especificación no depende de su cumplimiento. Estas recomendaciones contienen las expresiones "Recomendamos...", "Esta especificación recomienda...", o alguna frase similar.

### Hoja de estilo

Un conjunto de estamentos que especifican la forma de presentación de un documento.

Las hojas de estilo pueden tener tres orígenes distintos: [autor](#), [usuario](#), y [aplicación del usuario](#). La interacción de estas fuentes se describe en la sección sobre [cascada y herencia](#).

### Hoja de estilo válida

La validez de una hoja de estilo depende del nivel de CSS usado por la hoja de estilo. Todas las hojas de estilo CSS1 válidas son hojas de estilo CSS2 válidas. Sin embargo, algunos [cambios efectuados sobre CSS1](#) significan que unas pocas hojas de estilo CSS1 tendrán una semántica ligeramente diferente en CSS2.

Una hoja de estilo CSS2 debe estar redactada de acuerdo a la [gramática de CSS2](#). Además, debe contener sólo reglas-arroba, nombres de propiedades y valores de propiedades definidos en esta especificación. Una regla-arroba, nombre de propiedad o valor de propiedad **ilícito** (inválido) es aquel que no es válido.

### Documento fuente

El documento al cual una o más hojas de estilo se refieren. Este está codificado en algún lenguaje que representa al documento como una estructura de [elementos](#). Cada elemento consta de un nombre que identifica el tipo de elemento, opcionalmente algunos [atributos](#), y un [contenido](#) (posiblemente vacío).

### Lenguaje del documento

El lenguaje de codificación del documento fuente (ej., HTML o una aplicación XML).

### Elemento

(Un término SGML, ver [ISO8879](#).) Las estructuras sintácticas primarias del lenguaje del documento. La mayoría de las reglas de las hojas de estilo CSS usan el nombre de estos elementos (como "P", "TABLE" y "OL" para HTML) para especificar la información acerca de su procesamiento.

### Elemento reemplazado

Un elemento del cual el intérprete de CSS sólo conoce las dimensiones intrínsecas. En HTML, los elementos IMG, INPUT, TEXTAREA, SELECT y OBJECT pueden ser ejemplos de elementos reemplazados. Por ejemplo, el contenido del elemento IMG es a menudo reemplazado por la imagen que el atributo "src" designa. CSS no define cómo son establecidas las dimensiones intrínsecas.

### Dimensiones intrínsecas

El ancho y altura definidos por el propio elemento, no impuestos por el entorno. En CSS2 se asume que todos los elementos reemplazados --y sólo los elementos reemplazados-- poseen dimensiones intrínsecas.

### Atributo

Un valor asociado a un elemento consistente en un nombre y un valor (textual) asociado.

### Contenido

El contenido asociado a un elemento en el documento fuente; no todos los elementos tienen contenido, en cuyo caso son llamados **vacíos**. El contenido de un elemento puede incluir texto y puede incluir varios sub-elementos, en cuyo caso el elemento es llamado **padre** de esos sub-elementos.

### Contenido procesado

El contenido de un elemento después del procesamiento que se efectúa sobre él según las hojas de estilo pertinentes que han sido aplicadas. El contenido procesado de un [elemento reemplazado](#) proviene de fuera del documento fuente. El contenido procesado también puede ser el texto alternativo de un elemento (ej., el valor del atributo HTML "alt"), y puede incluir ítems insertados implícita o explícitamente por la hoja de estilo, como viñetas, numeradores, etc.

### Estructura del documento

La estructura de elementos codificados en el documento fuente. Cada elemento en esta estructura tiene exactamente un padre, con la excepción del elemento **raíz** que no tiene ninguno.

**Hijo**

Un elemento A es llamado hijo de un elemento B si, y sólo si, B es el padre de A.

**Descendiente**

Un elemento A es llamado descendiente de un elemento B ya sea que (1) A es hijo de B o (2) A es hijo de algún elemento C que es descendiente de B.

**Antepasado**

Un elemento A es llamado antepasado de un elemento B si, y sólo si, B es descendiente de A.

**Hermano**

Un elemento A es llamado hermano de un elemento B si, y sólo si, B y A comparten el mismo elemento padre.

Un elemento A es hermano precedente si viene antes que B en la estructura del documento. Un elemento A es hermano siguiente si viene después de B en la estructura del documento.

**Elemento precedente**

Un elemento A es llamado elemento precedente de un elemento B si, y sólo si, (1) A es un antecesor de B o (2) A es hermano precedente de B.

**Elemento siguiente**

Un elemento A es llamado elemento siguiente de un elemento B si, y sólo si, B es un elemento precedente de A.

**Autor**

Un autor es una persona que escribe documentos y las hojas de estilo asociadas. Una **herramienta de autor** genera documentos y las hojas de estilo asociadas.

**Usuario**

Un usuario es una persona que interactúa con una aplicación del usuario para ver, oír o usar de algún modo un documento y sus hojas de estilo asociadas. El usuario puede proporcionar una hoja de estilo personal que codifica sus preferencias individuales.

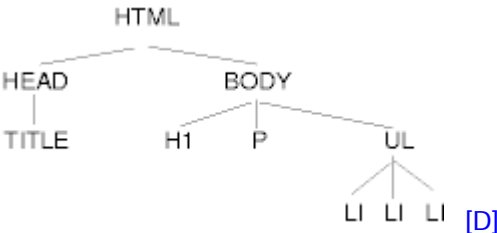
**Aplicación del usuario (AU)**

Una aplicación del usuario es cualquier programa que interpreta un documento escrito en el lenguaje del documento y aplica hojas de estilo asociadas según los términos de esta especificación. Una aplicación del usuario puede mostrar un documento, leerlo en voz alta, permitir que sea impreso, convertirlo a otro formato, etc.

Este es un ejemplo de un documento fuente codificado en HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <TITLE>Mi página personal</TITLE>
  <BODY>
    <H1>Mi página personal</H1>
    <P>¡Bienvenidos a mi página! Permítanme hablarles de mis compositores favoritos:
    <UL>
      <LI> Elvis Costello
      <LI> Johannes Brahms
      <LI> Georges Brassens
    </UL>
  </BODY>
</HTML>
```

Esto resulta en la siguiente estructura:



De acuerdo a la definición de HTML, los elementos HEAD serán inferidos durante el análisis y devendrán parte de la estructura del documento aún cuando las marcas HEAD no estén en la fuente del documento. De modo similar, el analizador sabe donde terminan los elementos P y LI, aunque no existan las marcas </P> y </LI> en la fuente.

3.2 Conformidad

Esta sección sólo define conformidad con la especificación CSS2. Puede haber otros futuros niveles de CSS que requieran a las aplicaciones del usuario implementar un conjunto de características diferente para obtener conformidad.

En general, los siguientes puntos deben ser observados por las aplicaciones del usuario que pretendan conformidad con esta especificación:

- 1. Debe soportar uno o más [tipos de medios](#) de CSS2.

2. Para cada documento fuente, debe intentar recuperar todas las hojas de estilo asociadas que sean apropiadas para los tipos de medio soportados. Si no puede recuperar todas las hojas de estilo asociadas (por ejemplo, debido a errores en la red), debe mostrar el documento usando aquellas que puede recuperar.
3. Debe analizar las hojas de estilo de acuerdo con esta especificación. En particular, debe reconocer todas las reglas-arroba, bloques, declaraciones y selectores (ver la [gramática de CSS2](#)). Si una aplicación del usuario encuentra una propiedad que se aplica a un tipo de medio soportado, la aplicación del usuario debe analizar el valor de acuerdo a la definición de la propiedad. Esto significa que la aplicación del usuario debe aceptar todos los valores válidos y debe ignorar las declaraciones con valores inválidos. Las aplicaciones del usuario deben ignorar reglas aplicadas a [tipos de medio](#) que no son soportados.
4. Para cada elemento en una [estructura del documento](#), debe asignar un valor por cada propiedad aplicable, de acuerdo con la definición de propiedad y las reglas de [cascada y herencia](#).
5. Si el documento fuente viene con hojas de estilo alternativas (tal como la palabra clave "alternate" en HTML 4 [[HTML40](#)]), la AU debe permitir al usuario seleccionar una de entre estas hojas de estilo y aplicar la seleccionada.

Sin embargo, no todas las aplicaciones del usuario deben observar cada punto:

- Una aplicación del usuario que permite *reconocer* hojas de estilo debe respetar los puntos 1 - 3.
- Una herramienta de autor sólo necesita generar [hojas de estilo válidas](#)
- Una aplicación del usuario que *procesa* un documento con hojas de estilo asociadas debe respetar los puntos 1 - 5 y procesar el documento de acuerdo a los requerimientos específicos de los medios planteados en esta especificación. Los [valores](#) pueden ser aproximados cuando la aplicación del usuario así lo requiera.

La incapacidad de una aplicación del usuario para implementar parte de esta especificación debido a las limitaciones de un dispositivo en particular (ej., una aplicación del usuario no puede presentar colores sobre un monitor monocromático o una página) no implica la no conformidad.

Esta especificación recomienda que una aplicación del usuario permita al usuario especificar sus propias hojas de estilo.

### 3.3 Condiciones de error

En general, este documento no especifica un comportamiento por parte de las aplicaciones de usuario en el manejo de errores (ej., cómo se comportan cuando no pueden encontrar un recurso designado por un URI).

Sin embargo, las aplicaciones del usuario deben observar las [reglas para el manejo de los errores de análisis](#).

Puesto que las aplicaciones del usuario pueden variar en el modo de manejar las condiciones de error, los autores y usuarios no deben confiar en un comportamiento específico para la recuperación de un error.

### 3.4 El tipo de contenido text/css

Las hojas de estilo CSS que existen en archivos separados son enviadas a través de Internet como una secuencia de bytes acompañada por la información de codificación (ver [[HTML40](#)], capítulo 5). La estructura de la transmisión, llamada **entidad del mensaje**, es definida por RFC 2045 y RFC 2068 (ver [[RFC2045](#)] y [[RFC2068](#)]). Una entidad del mensaje con un tipo de contenido "text/css" representa un documento CSS independiente. El tipo de contenido "text/css" ha sido registrado por RFC 2318 ([[RFC2318](#)]).

# 4 Sintaxis y tipos básicos de datos en CSS2

## 4.1 Sintaxis

Esta sección describe una gramática (y reglas de *análisis con compatibilidad futura*) común a cualquier versión de CSS (incluyendo CSS2). Las futuras versiones de CSS adherirán a esta sintaxis central, aunque pueden agregar restricciones sintácticas adicionales.

Estas descripciones son normativas. También están complementadas por las reglas normativas de gramática presentadas en el [Apéndice D](#).

### 4.1.1 Comandos

Todos los niveles de CSS --nivel 1, nivel 2 y cualquier futuro nivel-- usan la misma sintaxis central. Esto permite a las AU analizar (aún cuando no las entiendan completamente) las hojas de estilo escritas en niveles de CSS que no existían en el momento en que la AU fue creada. Los diseñadores pueden usar esta característica para crear hojas de estilo que funcionen con aplicaciones de usuario antiguas, mientras ejercitan también las posibilidades de los niveles más nuevos de CSS.

A nivel de léxico, las hojas de estilo CSS consisten de una secuencia de comandos. La lista de comandos para CSS2 es la que sigue. Las definiciones usan expresiones regulares al estilo de Lex. Los códigos octales se refieren a ISO 10646 ([ISO10646](#)). Como en Lex, en caso de múltiples equivalencias, la equivalencia más larga determina el comando.

| Comando       | Definición  |
|---------------|---|
| IDENT         | {ident}   |
| ATKEYWORD     | @{ident}  |
| STRING        | {string}  |
| HASH          | #{name}   |
| NUMBER        | {num}   |
| PERCENTAGE    | {num}%  |
| DIMENSION     | {num}{ident}  |
| URI           | url\(({w}\{string\}{w}\) url\(({w}([!#\$%&*~] {nonascii} {escape}))*{w}\) |
| UNICODE-RANGE | U\[0-9A-F?]{1,6}(-[0-9A-F]{1,6})?   |
| CDO           | <!--  |
| CDC           | -->   |
| ;             | ;   |
| {             | \{  |
| }             | \}  |
| (             | \(  |
| )             | \)  |
| [             | \[  |
| ]             | \]  |
| S             | [ \t\r\n\f]+  |
| COMMENT       | \/\^[^]*\^[^]*(\/\^[^]*\^[^]+)\/  |
| FUNCTION      | {ident}\(   |
| INCLUDES      | ~=  |
| DASHMATCH     | =   |
| DELIM         | cualquier otro carácter no equivalente con las reglas anteriores          |

Las macros entre llaves ({} arriba son definidas como sigue:

| Macro    | Definición                     |
|----------|--------------------------------|
| ident    | {nmstart}{nmchar}*             |
| name     | {nmchar}+                      |
| nmstart  | [a-zA-Z] {nonascii} {escape}   |
| nonascii | [\0-\177]                      |
| unicode  | \\[0-9a-f]{1,6}[\t\r\n\f]?     |
| escape   | {unicode} \\[ -~\200-\4177777] |



| Macro   | Definición  |
|---------|---|
| nmchar  | [a-zA-Z0-9]   {nonascii}   {escape}                   |
| num     | [0-9]+   [0-9]*\.[0-9]+                               |
| string  | {string1}   {string2}                                 |
| string1 | \"([\\t !#\$%&(-~] \\{nl} \\' {nonascii} {escape})*\" |
| string2 | '([\\t !#\$%&(-~] \\{nl} \\\" {nonascii} {escape})*'  |
| nl      | \\n \\r\\n \\r \\f                                    |
| w       | [ \\t\\r\\n\\f]*                                      |

Abajo está la sintaxis central para CSS. Las secciones siguientes describen cómo usarla. El [Apéndice D](#) describe una gramática más restrictiva que es más cercana al lenguaje de CSS nivel 2.

```
hoja de estilo : [ CDO | CDC | S | estamento ]*;
estamento    : regla | regla-arroba;
regla-arroba  : ATKEYWORD S* otros* [ bloque | ';' S* ];
bloque        : '{' S* [ otros | bloque | ATKEYWORD S* | ';' ]* '}' S*;
regla         : selector? '{' S* declaración? [ ';' S* declaración? ]* '}' S*;
selector      : otros+;
declaración   : propiedad ':' S* valor;
propiedad     : IDENT S*;
valor         : [ otros | bloque | ATKEYWORD S* ]+;
otros         : [ IDENT | NUMBER | PERCENTAGE | DIMENSION | STRING
                  | DELIM | URI | HASH | UNICODE-RANGE | INCLUDES
                  | FUNCTION | DASHMATCH | '(' otros* ')' | '[' otros* ']' ]
S*;
```

Los comandos COMMENT no aparecen en la gramática (para mantenerla legible), pero cualquier cantidad de estos comandos pueden aparecer en cualquier lugar entre otros comandos.

El comando S en la gramática anterior está colocado para los espacios en blanco. Solamente los caracteres "espacio" (código 32 en Unicode), "tabulador" (9), "avance de línea" (10), "retorno de carro" (13) y "avance de hoja" (12) pueden aparecer en los espacios en blanco. Otros caracteres que producen espacios, tales como "espacio eme" (8195) y "espacio ideográfico" (12288) nunca forman parte de los espacios en blanco.

### 4.1.2 Palabras clave

Las palabras clave toman la forma de identificadores. Las palabras clave no deben ir entre comillas ("..." o '...'). De este modo,

```
red
```

es una palabra clave, pero

```
"red"
```

no lo es. (Es una [cadena](#).) Otros ejemplos ilícitos:

```
width: "auto";
border: "none";
font-family: "serif";
background: "red";
```

### 4.1.3 Caracteres y mayúsculas/minúsculas

Las reglas siguientes siempre sostienen:

- Todas las hojas de estilo CSS son insensibles a las mayúsculas/minúsculas, excepto para las partes que no están bajo el control de CSS. Por ejemplo, la diferenciación entre mayúsculas y minúsculas en los valores de los atributos HTML "id" y "class", los nombres de las fuentes y los URI está fuera del alcance de esta especificación. Advierta en particular que los nombres de los elementos en HTML no hacen distinción entre mayúsculas y minúsculas, pero sí lo hacen en XML.
- En CSS2, los *identificadores* (incluyendo los nombres de los elementos, clases e ID de los [selectores](#)) pueden contener sólo caracteres [A-Za-z0-9] y los caracteres 161 en adelante en ISO 10646, más el guión (-); no pueden comenzar con un guión o un número. También pueden contener caracteres con escape y cualquier carácter de ISO 10646 en forma de código numérico (ver el siguiente ítem). Por ejemplo, el identificador "B&W?" puede escribirse como "B\&W?" o "B\26 W\3F".

Observe que Unicode es simétricamente equivalente a ISO 10646 (ver [\[UNICODE\]](#) e [\[ISO10646\]](#)).



- En CSS2, una barra invertida (\) indica tres tipos de caracteres de escape.

Primero, dentro de una [cadena](#), una barra invertida seguida de una nueva línea es ignorada (es decir, la cadena se evalúa para no contener la barra invertida o la nueva línea).

Segundo, cancela el significado de los caracteres especiales de CSS. Cualquier carácter (excepto un número hexadecimal) puede ser escapado con una barra invertida para quitar su significado especial. Por ejemplo, "\" es una cadena consistente en una comilla doble. Los pre-procesadores de las hojas de estilo no deben quitar estas barras invertidas de una hoja de estilo debido a que ello puede cambiar el significado de la misma.

Tercero, el escape con barra invertida permite a los autores referirse a caracteres que no pueden ponerse fácilmente en un documento. En este caso, la barra invertida es seguida por hasta seis números hexadecimales (0..9A..F), que representan el carácter en ISO 10646 ([ISO10646](#)) con ese número. Si un número o letra sigue al número hexadecimal, el final del número debe ser aclarado. Hay dos maneras de hacer eso:

1. con un espacio (u otro carácter de espacio en blanco): "\26 B" ("&B")
2. proporcionando exactamente 6 números hexadecimales: "\000026B" ("&B")

De hecho, estos dos métodos puede combinarse. Sólo un carácter de espacio en blanco es ignorado después del escape hexadecimal. Advierta que esto significa que un espacio "real" después de la secuencia de escape debe ir él mismo con escape o duplicado.

- Los escapes con barra invertida son siempre considerados como parte de un [identificador](#) o de una cadena (i.e., "\7B" no es una puntuación, aún cuando "{" sí lo es, y "\32" está permitido al comienzo el nombre de una clase, aunque "2" no lo está).

## 4.1.4 Estamentos

Una hoja de estilo CSS, en cualquier versión de CSS, consiste de un listado de *estamentos* (ver la gramática arriba). Hay dos clases de estamentos: *reglas-arroba* y *juegos de reglas*. Puede haber [espacios en blanco](#) alrededor de los estamentos.

En esta especificación, la expresiones "inmediatamente ante" o "inmediatamente después" significan sin ninguna interposición de espacios en blanco o comentarios.

## 4.1.5 Reglas-arroba

Las Reglas-arroba comienzan con una *palabra clave-arroba*, un carácter '@' seguido inmediatamente por un [identificador](#) (por ejemplo, '@import', '@page').

Una regla-arroba consiste en todo lo que hay hasta, e incluyendo, el siguiente punto y coma (;) o el siguiente [bloque](#), cualquiera sea lo primero en aparecer. Una aplicación del usuario CSS que encuentra una regla-arroba irreconocible debe [ignorar](#) la totalidad la regla-arroba y continuar el análisis después de ella.

Las aplicaciones del usuario deben [ignorar](#) cualquier regla '@import' que aparezca dentro de un [bloque](#) o que no preceda a todos los juegos de reglas.

Asumiendo, por ejemplo, que el analizador CSS2 encuentra esta hoja de estilo:

```
@import "subs.css";
H1 { color: blue }
@import "list.css";
```

La segunda '@import' es ilícita de acuerdo a CSS2. El analizador CSS2 [ignora](#) la regla-arroba completa, reduciendo de hecho la hoja de estilo a:

```
@import "subs.css";
H1 { color: blue }
```

En el ejemplo siguiente, la segunda regla '@import' no es válida, por cuanto aparece dentro de un [bloque](#) '@media'.

```
@import "subs.css";
@media print {
  @import "print-main.css";
  BODY { font-size: 10pt }
}
H1 {color: blue }
```

### 4.1.6 Bloques

Un *bloque* comienza con una llave ( { ) y termina con la correspondiente llave hacia la derecha ( } ). En medio puede entrar cualquier carácter, excepto que los paréntesis ( ( ) ), los corchetes ( [ ] ) las llaves ( { } ) deben aparecer siempre con el par correspondiente y pueden anidarse. Las comillas simples ( ' ) y dobles ( " ) también deben ir de a pares, y los caracteres encerrados por ellas son analizados como una cadena. Ver los [Comandos](#) arriba por la definición de una cadena.

Aquí hay un ejemplo de bloque. Observe que la llave derecha entre las comillas dobles no se corresponden con la llave de apertura del bloque, y que la segunda comilla simple es un [carácter con escape](#), y de este modo no se corresponde con la primer comilla simple:

```
{ causta: "}" + ( {7} * '\'' ) }
```

Advierta que la regla anterior no es válida en CSS2, pero aún sigue siendo un bloque tal como se define arriba.

### 4.1.7 Juegos de reglas, bloques de declaraciones y selectores

Un juego de reglas (también llamado "regla") consiste de un selector seguido de un bloque de declaraciones.

Un *bloque de declaraciones* (también llamado bloque- { } en el siguiente texto) comienza con una llave izquierda ( { ) y termina con la llave derecha ( } ) correspondiente. En medio de ellas debe haber una lista de cero o más declaraciones separadas por punto y coma ( ; ).

El *selector* (ver también la sección sobre [selectores](#)) consiste en todo lo que hay hasta (pero sin incluir) la primer llave izquierda ( { ). Un selector siempre va junto con un bloque- { }. Cuando una aplicación del usuario no puede analizar el selector (es decir, no es un CSS2 válido), debe [ignorar](#) también el bloque- { }.

CSS2 le da un significado especial a la coma ( , ) en los selectores. Sin embargo, como se desconoce si la coma puede adquirir otros significados en las versiones futuras de CSS, el estamento entero debe ser [ignorado](#) si hay un error en cualquier parte del selector, aún cuando el resto del selector pueda parecer razonable en CSS2.

Por ejemplo, como el "&" no es un comando válido en un selector CSS2, la aplicación del usuario CSS2 debe [ignorar](#) la segunda línea completa y no poner el color de H3 como rojo:

```
H1, H2 {color: green }
H3, H4 & H5 {color: red }
H6 {color: black }
```

Aquí hay un ejemplo más complejo. Los primeros dos pares de llaves están dentro de una cadena y no señalan el cierre del selector. Este es un estamento CSS2 válido.

```
P[ejemplo="public class foo\
{\
    private int x;\
\
    foo(int x) {\
        this.x = x;\
    }\
\
}"] { color: red }
```

### 4.1.8 Declaraciones y propiedades

Una *declaración* puede ser vacía o consistir en una propiedad, seguida por dos puntos ( : ), seguidos por un valor. Alrededor de cada uno de estos pueden haber [espacios en blanco](#).

Debido a la manera en que trabajan los selectores, las declaraciones múltiples para el mismo selector pueden organizarse en grupos separados por punto y coma ( ; ).

De este modo, las siguientes reglas:

```
H1 { font-weight: bold }
H1 { font-size: 12pt }
H1 { line-height: 14pt }
H1 { font-family: Helvetica }
H1 { font-variant: normal }
H1 { font-style: normal }
```

son equivalentes a:

```
H1 {
  font-weight: bold;
  font-size: 12pt;
  line-height: 14pt;
  font-family: Helvetica;
  font-variant: normal;
  font-style: normal
}
```

Una propiedad es un [identificador](#). Cualquier carácter puede aparecer en el valor, pero los paréntesis ("()"), los corchetes ("[]"), las llaves ("{}"), las comillas simples (') y las comillas dobles (") deben ir con su par correspondiente, y los punto y coma que no formen parte de una cadena deben ir con [escape](#). Los paréntesis, los corchetes y las llaves pueden anidarse. Dentro de las comillas, los caracteres son tomados como una cadena.

La sintaxis de los valores se especifica separadamente para cada propiedad, pero en todos los casos, los valores están compuestos de identificadores, cadenas, números, medidas, porcentajes, URI, colores, ángulos, tiempos y frecuencias.

Una aplicación del usuario debe [ignorar](#) una declaración con un nombre de propiedad no válido o un valor ilícito. Cada propiedad CSS2 tiene sus propias restricciones sintácticas y semánticas para los valores que acepta.

Por ejemplo, asumiendo que un analizador CSS2 encuentra esta hoja de estilo:

```
H1 { color: red; font-style: 12pt } /* Valor ilícito: 12pt */
P { color: blue; font-vendor: any; /* Propiedad no válida: font-vendor */
  font-variant: small-caps }
EM EM { font-style: normal }
```

La segunda declaración en la primer línea contiene un valor ilícito: '12pt'. La segunda declaración en la segunda línea contiene una propiedad indefinida 'font-vendor'. El analizador CSS2 [ignorará](#) estas declaraciones, reduciendo de hecho la hoja de estilo a:

```
H1 { color: red; }
P { color: blue; font-variant: small-caps }
EM EM { font-style: normal }
```

## 4.1.9 Comentarios

Los comentarios comienzan con los caracteres "/\*" y terminan con los caracteres "\*/". Pueden aparecer en cualquier lugar entre los comandos y su contenido no tiene ninguna influencia en el procesamiento. Los comentarios no pueden anidarse.

CSS también permite los delimitadores de comentarios de SGML ("<!--" y "-->") en determinados lugares, pero estos no delimitan comentarios CSS. Están permitidos para que las reglas de estilo que aparecen en un documento fuente HTML (en el elemento STYLE) puedan ocultarse en las aplicaciones del usuario anteriores a HTML 3.2. Ver la especificación HTML 4.0 ([HTML40](#)) para más información.

## 4.2 Reglas para el manejo de errores en el análisis

En algunos casos, las aplicaciones del usuario deben ignorar parte de una hoja de estilo ilícita. Esta especificación utiliza el término *ignorar* para indicar que la aplicación del usuario analice la parte ilícita (con el fin de localizar su comienzo y su final) pero que por otro lado actúe como si allí no hubiera nada.

Para asegurarse de que nuevas propiedades y nuevos valores para las propiedades existentes puedan agregarse en el futuro, se exige a las aplicaciones del usuario cumplir con las siguientes reglas cuando se encuentren con estos casos:

- **Propiedades desconocidas.** Las aplicaciones del usuario deben [ignorar](#) una [declaración](#) con una propiedad desconocida. Por ejemplo, si la hoja de estilo es:

- 
- ```
H1 { color: red; rotation: 70minutes }
```

la aplicación del usuario tratará esto como si la hoja de estilo hubiera sido

```
H1 { color: red }
```

- **Valores ilícitos.** Las aplicaciones del usuario deben ignorar una declaración con un valor ilícito. Por ejemplo:

- 
- ```
IMG { float: left } /* correcto en CSS2 */
```

- `IMG { float: left here } /* "here" no es un valor de 'float' */`
- `IMG { background: "red" } /* las palabras clave no pueden ir entre comillas en CSS2 */`
- `IMG { border-width: 3 } /* debe especificarse la unidad en los valores de medida */`

Un analizador CSS2 admitirá la primer regla e [ignorar](#)á el resto, como si la hoja de estilo hubiera sido:

```
IMG { float: left }
IMG { }
IMG { }
IMG { }
```

Una aplicación del usuario que conforma a una especificación de CSS futura puede aceptar una o más de las otras reglas también.

- **Palabras clave-arroba inválidas.** Las aplicaciones del usuario deben [ignorar](#) una palabra clave-arroba no válida junto con todo lo que le sigue hasta (e incluyendo) el punto y coma (;) o bloque ({...}) siguientes, cualquiera sea el que aparezca primero. Por ejemplo, considere lo siguiente:

- 
- `@three-dee {`
- `@background-lighting {`
- `azimuth: 30deg;`
- `elevation: 190deg;`
- `}`
- `H1 { color: red }`
- `}`
- `H1 { color: blue }`

La regla-arroba '@three-dee' no forma parte de CSS2. Por consiguiente, la regla-arroba entera (hasta, e incluyendo, la tercer llave derecha) es [ignorada](#). Una aplicación del usuario CSS2 la [ignora](#), reduciendo de hecho la hoja de estilo a:

```
H1 { color: blue }
```

## 4.3 Valores

### 4.3.1 Enteros y números reales

Algunos tipos de valores pueden tener valores enteros (denotados por <entero>) o por valores con números reales (denotados por <número>). Los números reales y los enteros se especifican en notación decimal solamente. Un <entero> consiste en uno o más dígitos de "0" a "9". Un <número> puede ser un <entero> o puede ser cero o más dígitos seguidos por un punto (.) seguido por uno o más dígitos. Ambos, enteros y números reales, pueden estar precedidos por un "-" o "+" para indicar su signo.

Advierta que muchas propiedades que permiten un entero o un número real como valor, en realidad restringen el valor a cierto rango, a menudo a un valor no negativo.

### 4.3.2 Medidas

Las medidas se refieren a las dimensiones horizontales y verticales.

El formato de un valor de medida (denotado por <medida> en esta especificación) es un signo opcional ('+' o '-'), siendo '+' el predeterminado) seguido inmediatamente por un <número> (con o sin punto decimal) seguido inmediatamente por un identificador de la unidad (ej., px, deg, etc.). Después de la medida '0', el identificador de la unidad es opcional.

Algunas propiedades permiten valores de medida negativos, pero esto puede complicar el modelo del formato y pueden existir limitaciones propias de la implementación. Si un valor de medida negativo no es soportado, debe ser convertido al valor más cercano que pueda ser soportado.

Hay dos tipos de unidades de medida: relativas y absolutas. Las unidades de *medidas relativas* especifican una medida en relación a otra propiedad de medida. Las hojas de estilo que utilizan unidades relativas modificarán su escala de un medio a otro más fácilmente (ej., de un monitor de computadora a una impresora laser).

Las unidades relativas son:

- **em:** el tamaño ('font-size') de la fuente relevante
- **ex:** la 'altura de la x' de la fuente relevante
- **px:** pixeles, relacionado con los dispositivos visuales

```
H1 { margin: 0.5em }      /* em */
H1 { margin: 1ex }       /* ex */
P  { font-size: 12px }    /* px */
```

La unidad 'em' es igual al valor computado de la propiedad 'font-size' del elemento en el que se usa. La excepción es cuando 'em' aparece en el valor de la propiedad 'font-size' en sí misma, en cuyo caso se refiere al tamaño de la fuente del elemento padre. Puede usarse para longitudes verticales u horizontales. (Esta unidad también es denominada cuadratín en los textos sobre tipografía.)

La unidad 'ex' es definida por la propiedad 'x-height' de la fuente. La altura de la x es denominada así debido a que es igual a la altura de la letra "x" minúscula. Sin embargo, 'ex' es definida aún para aquellas fuentes que no contienen la letra "x".

La regla:

```
H1 { line-height: 1.2em }
```

significa que la altura de línea de los elementos H1 será un 20% mayor que el tamaño de fuente de los elementos H1. Por otro lado:

```
H1 { font-size: 1.2em }
```

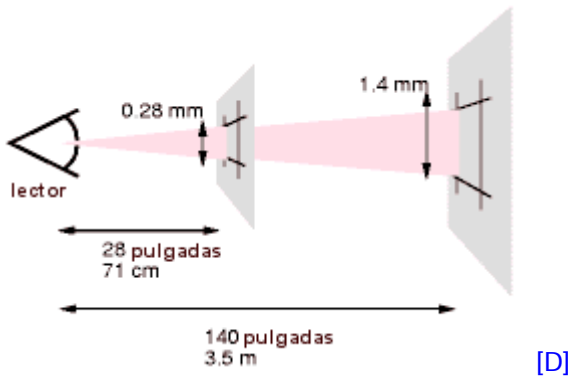
significa que la propiedad font-size de los elementos H1 será un 20% mayor que el tamaño de fuente heredado por los elementos H1.

Cuando se especifica en la raíz de la [estructura del documento](#) (ej., "HTML" en HTML), 'em' y 'ex' se refieren al [valor inicial](#) de la propiedad.

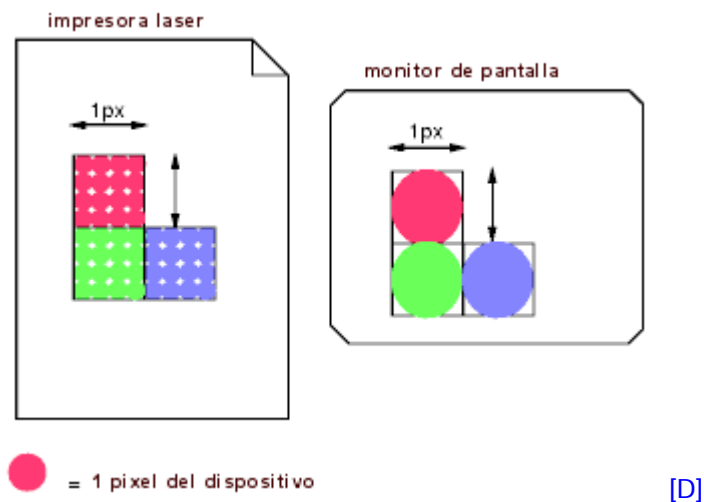
Las unidades pixeles están relacionadas con la resolución de los dispositivos visuales, esto es, la mayoría de las veces un monitor de computadora. Si la densidad de pixeles del dispositivo de salida es muy diferente de un típico monitor de computadora, la aplicación del usuario debe re-escalar los valores de los pixeles. Se recomienda que el *pixel de referencia* sea el ángulo visual de un pixel en un dispositivo con una densidad de pixeles de 90dpi y una distancia al lector del largo de su brazo. Para una longitud nominal del brazo de 28 pulgadas, el ángulo visual es de aproximadamente 0.0227 grados.

Para una distancia de lectura de la longitud de un brazo, 1 px corresponde aproximadamente a 0.28 mm (1/90 pulgadas). Cuando se imprime en una impresora laser, con una distancia de lectura un poco menor a la longitud del brazo (55 cm, 21 pulgadas), 1 px es aproximado a 0.21 mm. En una impresora de 300 puntos por pulgada (dpi), puede redondearse en 3 puntos (0.25 mm); en una impresora de 600 dpi, puede redondearse en 5 puntos.

Las dos imágenes de abajo ilustran el efecto de la distancia de lectura en el tamaño del pixel y el efecto de la resolución de un dispositivo. En la primer imagen, una distancia de lectura de 71 cm (28 pulgadas) resulta en un px de 0.28 mm, mientras que una distancia de lectura de 3.5 m (12 pies) requiere un px de 1.4 mm.



En la segunda imagen, un área de 1px por 1px es cubierta por un solo punto en un dispositivo de baja resolución (una pantalla de computadora), mientras que la misma área es cubierta por 16 puntos en un dispositivo de alta resolución (tal como una impresora laser de 400 dpi).



Los elementos hijos no heredan los valores relativos especificados para sus padres; ellos (generalmente) heredan los [valores computados](#).

En las siguientes reglas, el valor computado de 'text-indent' de los elementos H1 será de 36pt, no de 45pt, si H1 es un hijo del elemento BODY.

```
BODY {
  font-size: 12pt;
  text-indent: 3em; /* i.e., 36pt */
}
H1 { font-size: 15pt }
```

La unidades de *medida absoluta* son útiles solamente cuando la propiedades físicas del medio de salida son conocidas. Las unidades absolutas son:

- **in**: inches -- 1 inch (pulgada) es igual a 2.54 centímetros.
- **cm**: centímetros
- **mm**: milímetros
- **pt**: puntos -- los puntos usados por CSS2 equivalen a un 1/72 de pulgada.
- **pc**: picas -- 1 pica es igual a 12 puntos.

```
H1 { margin: 0.5in } /* inches (pulgadas) */
H2 { line-height: 3cm } /* centímetros */
H3 { word-spacing: 4mm } /* milímetros */
H4 { font-size: 12pt } /* puntos */
H4 { font-size: 1pc } /* picas */
```

En caso de que la medida especificada no pueda ser soportada, las aplicaciones del usuario deben aproximarla al valor real.

### 4.3.3 Porcentajes

El formato de un valor en porcentaje (denotado por <porcentaje> en esta especificación) es un signo opcional ('+' o '-', siendo '+' el predeterminado) seguido inmediatamente por un <número> seguido inmediatamente por '%'.

Los valores expresados en porcentajes son siempre relativos a otro valor, por ejemplo una medida. Cada propiedad que admite porcentajes también define el valor al cual se refiere el porcentaje. El valor puede ser aquel de otra propiedad para el mismo elemento, una propiedad para un elemento antepasado o un valor en el contexto del formato (por ejemplo, el ancho de un [bloque de contención](#)). Cuando un porcentaje es atribuido a una propiedad del elemento raíz y el porcentaje es definido en referencia al valor heredado de alguna propiedad, el valor resultante es la cantidad del porcentaje del [valor inicial](#) de esta propiedad.

Como los elementos hijos (generalmente) heredan los [valores computados](#) de sus padres, en el siguiente ejemplo, el hijo del elemento P heredará un valor de 12pt para 'line-height', no el valor del porcentaje (120%):

```
P { font-size: 10pt }
P { line-height: 120% } /* 120% de 'font-size' */
```

### 4.3.4 URL + URN = URI

Los URL (de "Uniform Resource Locators", Localizadores Uniformes de Recursos, ver [RFC1738](#) y [RFC1808](#)) proporciona la dirección de un recurso en la Web. Una nueva forma en perspectiva para identificar recursos es llamada URN (de "Uniform Resource Name", Nombre Uniforme de Recurso). Juntos son llamados URI (de

"Uniform Resource Identifiers", Identificadores Uniformes de Recursos, ver [\[URI\]](#)). Esta especificación usa el término URI.

Los valores de los URI en esta especificación son denotados por <uri>. La expresión funcional usada para designar los URI en los valores de la propiedad es "url()", como en:

```
BODY { background: url("http://www.bg.com/pinkish.gif") }
```

El formato del valor de un URI es 'url(' seguido de un [espacio en blanco](#) opcional, seguido de una comilla simple (') o comilla doble (") opcional, seguida por el URI propiamente dicho, seguido de una comilla simple (') o comilla doble (") opcional, seguida por un espacio en blanco opcional, seguido de ')'. Las dos comillas tienen que ser las mismas.

Un ejemplo sin comillas:

```
LI { list-style: url(http://www.redballs.com/redball.png) disc }
```

Los paréntesis, las comas, los espacios en blanco, las comillas simples (') y las comillas dobles (") que aparecen en un URI deben ir escapadas con una barra invertida: \'(', '\)', '\,', '\', \'.

Dependiendo del tipo de URI, se podría escribir los caracteres de arriba como escapes en el URI (donde "(" = %28, ")" = %29, etc.) como se describe en [\[URI\]](#).

Con el fin de crear hojas de estilo modulares que no sean dependientes de la posición absoluta de un recurso, los autores pueden usar URI relativos. Los URI relativos (como se define en [\[RFC1808\]](#)) son convertidos a URI completos usando un URI de base. La RFC 1808, sección 3, define el algoritmo normativo para este proceso. Para las hojas de estilo CSS, el URI de base es aquel de la hoja de estilo, no el del documento fuente.

Por ejemplo, suponga que la siguiente regla:

```
BODY { background: url("yellow") }
```

está localizada en una hoja de estilo designada por el URI:

```
http://www.myorg.org/style/basic.css
```

El fondo de BODY en el documento fuente será un mosaico de cualquier imagen que sea descripta por el recurso designado por el URI

```
http://www.myorg.org/style/yellow
```

Las aplicaciones del usuario pueden variar el modo de manejar los URI que designan recursos no disponibles o inaplicables.

### 4.3.5 Contadores

Los contadores son denotados por identificadores (ver las propiedades 'counter-increment' y 'counter-reset'). Para referirse al valor de un contador, se usa la notación 'counter(<identificador>)' o 'counter(<identificador>, <estilo-de-lista>)'. El estilo predeterminado es 'decimal'.

Para referirse a una secuencia de contadores anidados del mismo nombre, la notación es 'counters(<identificador>, <cadena>)' o 'counters(<identificador>, <cadena>, <estilo-de-lista>)'. Ver ["Contadores anidados y área de alcance"](#) en el capítulo sobre [contenido generado](#).

En CSS2, el valor de los contadores sólo puede ser referenciado desde la propiedad 'content'. Note que 'none' es un posible <estilo-de-lista>: 'counter(x, none)' genera una cadena vacía.

Aquí hay una hoja de estilo que numera los párrafos (P) para cada capítulo (H1). Los párrafos son numerados con números romanos, seguidos por un punto y un espacio:

```
P { counter-increment: par-num }
H1 { counter-reset: par-num }
P:before { content: counter(par-num, upper-roman) ". " }
```

Los contadores que no están en el [área de alcance](#) de ningún 'counter-reset', se asume que han sido reiniciados a 0 por un 'counter-reset' en el elemento raíz.

### 4.3.6 Colores

Un <color> es o una palabra clave o una especificación numérica RGB.



La lista de los nombres de las palabras clave es: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white y yellow. Estos 16 colores son definidos en HTML 4.0 ([HTML40]). Además de estas palabras clave de colores, los usuarios pueden especificar palabras clave que corresponden a los colores usados por ciertos objetos en el entorno del usuario. Por favor, consulte la sección sobre los [colores del sistema](#) para más información.

```
BODY {color: black; background: white }
H1 { color: maroon }
H2 { color: olive }
```

El modelo de color RGB es usado en la especificación numérica del color. Todos estos ejemplos especifican el mismo color:

```
EM { color: #f00 }           /* #rgb */
EM { color: #ff0000 }        /* #rrggbb */
EM { color: rgb(255,0,0) }    /* rango de enteros 0 - 255 */
EM { color: rgb(100%, 0%, 0%) } /* rango flotante 0.0% - 100.0% */
```

El formato de un valor RGB en notación hexadecimal es un '#' seguido inmediatamente por tres o seis caracteres hexadecimales. La notación RGB de tres dígitos (#rgb) es convertida en la forma de seis dígitos (#rrggbb) por repetición de dígitos, no por la adición de ceros. Por ejemplo, #fb0 se expande a #ffbb00. Esto garantiza que el blanco (#ffffff) pueda ser especificado con la forma reducida (#fff) y quitar cualquier dependencia con la profundidad de color del monitor.

El formato de un valor RGB en la notación funcional es 'rgb(' seguido por una lista separada por comas de tres valores numéricos (ya sean tres valores enteros o tres valores de porcentajes) seguidos de ')'. El valor entero 255 corresponde a 100%, y a F o FF en la notación hexadecimal: rgb(255,255,255) = rgb(100%,100%,100%) = #FFF. Los [espacios en blanco](#) son permitidos alrededor de los valores numéricos.

Todos los colores RGB son especificados en el espacio de color sRGB (ver [SRGB]). Las aplicaciones del usuario pueden variar en la fidelidad con la que representan estos colores, pero el uso de sRGB proporciona una definición mensurable, objetiva y sin ambigüedades, de lo que debe ser el color, el cual puede ser relacionado con estándares internacionales (ver [COLORIMETRY]).

Las aplicaciones del usuario con conformidad pueden limitar su esfuerzo en el despliegue del color para realizar una corrección de gama en los mismos. sRGB especifica una gama de monitor de 2.2 bajo ciertas condiciones de percepción. Las aplicaciones del usuario deben ajustar los colores dados en CSS de manera tal que, en combinación con una gama de monitor "natural" en el dispositivo de salida, se produzca una gama de monitor efectiva de 2.2. Ver la sección sobre [corrección de gama](#) por más detalles. Advierta que sólo los colores especificados en CSS son afectados; ej., se espera que las imágenes contengan su propia información de color.

Los valores fuera de la gama del dispositivo deben ser recortados: los valores de rojo, verde y azul tienen que ser cambiados para adecuarlos al rango soportado por el dispositivo. Para un monitor CRT típico, cuya gama es la misma que sRGB, las tres reglas de abajo son equivalentes:

```
EM { color: rgb(255,0,0) }    /* rango de enteros 0 - 255 */
EM { color: rgb(300,0,0) }    /* recortado a rgb(255,0,0) */
EM { color: rgb(255,-10,0) }  /* recortado a rgb(255,0,0) */
EM { color: rgb(110%, 0%, 0%) } /* recortado a rgb(100%,0%,0%) */
```

Otros dispositivos, tales como impresoras, tiene gamas distintas a sRGB; algunos colores fuera del rango 0...255 de sRGB serán representables (dentro de la gama del dispositivo), mientras que otros colores dentro del rango 0...255 de sRGB estarán fuera de la gama del dispositivo y de ese modo serán recortados.

**Nota.** Si bien los colores pueden agregar una significativa cantidad de información al documento y los hacen más legible, considere por favor que ciertas combinaciones de color pueden causar problemas a los usuarios con ceguera para los colores. Si usa una imagen de fondo o pone un color de fondo, por favor ajuste sus colores adecuadamente.

### 4.3.7 Angulos

Los valores de ángulos (denotados por <ángulo> en el texto) son usados con las [hojas de estilo auditivas](#).

Su formato es un signo opcional ('+' o '-', siendo '+' el predeterminado) seguido inmediatamente por un <número> seguido inmediatamente por un identificador de la unidad del ángulo.

Los identificadores de la unidad de ángulo son:

- **deg:** degrees (grados)
- **grad:** grads
- **rad:** radians

Los valores de ángulos pueden ser negativos. Deben ser normalizados en un rango de 0-360deg por la aplicación del usuario. Por ejemplo, -10deg y 350deg son equivalentes.

Por ejemplo, un ángulo derecho es de '90deg' o '100grad' o '1.570796326794897rad'.

### 4.3.8 Tiempos

Los valores de tiempo (denotados por <tiempo> en el texto) son usados con las [hojas de estilo auditivas](#).

Su formato es un <número> seguido inmediatamente por un identificador de la unidad de tiempo.

Los identificadores para la unidad de tiempo son:

- **ms**: milisegundos
- **s**: segundos

Los valores de tiempo no pueden ser negativos.

### 4.3.9 Frecuencias

Los valores de la frecuencia (denotados por <frecuencia> en el texto) son usados con las [hojas de estilo en cascada auditivas](#).

Su formato es un <número> seguido inmediatamente por un identificador de la unidad de frecuencia.

Los identificadores de la unidad de frecuencia son:

- **Hz**: Hertz
- **kHz**: kilo Hertz

Los valores de la frecuencia no pueden ser negativos.

Por ejemplo, 200Hz (o 200hz) es un sonido bajo y 6kHz (o 6khz) es un sonido agudo.

### 4.3.10 Cadenas

Las cadenas pueden escribirse con comillas dobles o con comillas simples. Las comillas dobles no pueden aparecer dentro de comillas dobles, a menos que sean escapadas (como "\"" o como "\"22"). Igualmente para las comillas simples ("'" o "\"27").

```
"ésta es una 'cadena'"
"ésta es una \"cadena\" "
'ésta es una "cadena" '
'ésta es una \'cadena\' '
```

Una cadena no puede contener una nueva línea directamente. Para incluir una nueva línea en una cadena, use el escape "\A (la A hexadecimal es el carácter para el avance de línea en Unicode, pero representa la noción genérica de "nueva línea" en CSS). Ver la propiedad 'content' por un ejemplo.

Es posible separar las cadenas en varias líneas, por razones estéticas u otras razones, pero en tales casos la nueva línea tiene que ser escapada con la barra invertida (\). Por ejemplo, los siguientes dos selectores son exactamente lo mismo:

```
A[TITLE="un título n\
o demasiado largo"] { /*...*/ }
A[TITLE="un título no demasiado largo"] { /*...*/ }
```

## 4.4 Representación de los documentos CSS

Una hoja de estilo CSS es una secuencia de caracteres del Conjunto Universal de Caracteres (Universal Character Set) (ver [\[ISO10646\]](#)). Para su transmisión y almacenamiento, estos caracteres deben ser codificados por una codificación de caracteres que soporte el conjunto de caracteres disponibles en US-ASCII (ej., ISO 8859-x, SHIFT JIS, etc.). Por una buena introducción a los juegos de caracteres y las codificaciones de caracteres, por favor consulte la especificación HTML 4.0 ([\[HTML40\]](#), capítulo 5). Ver también la especificación XML 1.0 ([\[XML10\]](#), secciones 2.2 y 4.3.3, y Apéndice F).

Cuando una hoja de estilo está incrustada en otro documento, tal como en el elemento STYLE o el atributo "style" de HTML, la hoja de estilo comparte la misma codificación de caracteres que todo el documento.

Cuando una hoja de estilo reside en un archivo separado, las aplicaciones del usuario deben observar las siguientes prioridades cuando determinan la codificación de caracteres de un documento (de la prioridad más alta a la más baja):

1. Un parámetro HTTP "charset" en un campo "Content-Type".
2. La regla-arroba @charset.
3. Mecanismos del lenguaje del documento referenciado (ej., en HTML, el atributo "charset" del elemento LINK).

A lo sumo una regla @charset puede aparecer en una hoja de estilo externa --no debe aparecer en una hoja de estilo incrustada-- y debe aparecer bien al comienzo del documento, sin ser precedida por ningún carácter. Después de "@charset", los autores especifican el nombre de una codificación de caracteres. El nombre debe ser un nombre de conjunto de caracteres como son descritos en el registro IANA (Ver [IANA](#)). También, vea [CHARSETS](#) por una lista completa de conjuntos de caracteres). Por ejemplo:

@charset "ISO-8859-1";

Esta especificación no ordena cuáles codificaciones de caracteres deben soportar las aplicaciones del usuario.

Note que la confianza en la estructura de @charset teóricamente plantea un problema debido a que no hay información *a priori* sobre cómo está codificada. En la práctica, sin embargo, las codificaciones de uso generalizado en Internet están basadas en ASCII, UTF-16, UCS-4, o (raramente) en EBCDIC. Esto significa que, en general, los valores del byte inicial de un documento habilitan a la aplicación del usuario para detectar la familia de codificación fidedignamente, lo cual provee suficiente información para decodificar la regla @charset, la que a su vez determina la codificación de caracteres exacta.

### 4.4.1 Referencias a caracteres no representados en una codificación de caracteres

Una hoja de estilo puede tener que referirse a caracteres que no pueden ser representados en la actual codificación de caracteres. Estos caracteres deben ser escritos como referencias [escapadas](#) a los caracteres ISO 10646. Estos escapes sirven al mismo propósito que los caracteres numéricos referenciados en los documentos HTML o XML (ver [HTML40](#), capítulos 5 y 25).

El mecanismo de escape de caracteres debería ser usado sólo cuando unos pocos caracteres deben representarse de este modo. Si la mayor parte del documento requiere de escapes, los autores deberían codificarlo con una codificación más apropiada (ej., si el documento contiene una gran cantidad de caracteres en griego, los autores podrían usar "ISO-8859-7" o "UTF-8").

Los procesadores intermedios que utilizan una codificación de caracteres diferente pueden traducir estas secuencias con escape a secuencias de bytes de esa codificación. Los procesadores intermedios no deben, por otro lado, alterar las secuencias de escape que cancelen el significado especial de un carácter en ASCII.

Las [aplicaciones del usuario con conformidad](#) deben asignar en Unicode a todos los caracteres en cualquier codificación de caracteres que reconozca (o debe comportarse como si lo hubiera hecho).

Por ejemplo, un documento transmitido como ISO-8859-1 (Latin-1) no puede contener directamente las letras griegas: "....." (griego: "kouros") tiene que escribirse como "\3BA\3BF\3C5\3C1\3BF\3C2".

***Nota. En HTML 4.0, las referencias con caracteres numéricos son interpretadas en los valores del atributo "style" pero no en el contenido del elemento STYLE. Debido a esta asimetría, recomendamos que los autores utilicen el mecanismo de escape de caracteres de CSS en lugar de referencias con caracteres numéricos para ambos, el atributo "style" y el elemento STYLE. Por ejemplo, recomendamos:***

<SPAN style="voice-family: D\FC rst">...</SPAN>

***en lugar de:***

<SPAN style="voice-family: D&#252;rst">...</SPAN>

# 5 Selectores

## 5.1 Equivalencia de patrones

En CSS, las reglas de las equivalencias de patrones determinan cuales reglas de estilo se aplican a los elementos en la [estructura del documento](#). Estos patrones, denominados selectores, pueden ir desde el simple nombre de un elemento hasta complejos patrones contextuales. Si todas las condiciones en el patrón son verdaderas para un determinado elemento, el selector *equivale* al elemento.

La distinción entre mayúsculas/minúsculas en los nombres de los elementos de los selectores, dentro del lenguaje del documento, depende del lenguaje del documento. Por ejemplo, en HTML, los nombres de los elementos son insensibles a la diferencia mayúsculas/minúsculas, pero en XML sí hacen distinción entre mayúsculas y minúsculas.

La siguiente tabla resume la sintaxis de los selectores CSS2:

| Patrón                         | Significado  | Descrito en la sección                       |
|--------------------------------|--|--|
| *                              | Equivale a cualquier elemento.   | <a href="#">Selector universal</a>           |
| E                              | Equivale a cualquier elemento E (ej., un elemento de tipo E).  | <a href="#">Selectores de tipos</a>          |
| E F                            | Equivale a cualquier elemento F que es descendiente de un elemento E.  | <a href="#">Selectores de descendientes</a>  |
| E > F                          | Equivale a cualquier elemento F que es hijo de un elemento E.  | <a href="#">Selectores de hijos</a>          |
| E:first-child                  | Equivale al elemento E cuando E es el primer hijo de su padre.   | <a href="#">La pseudo-clase :first-child</a> |
| E:link<br>E:visited            | Equivale al elemento E si E es el ancla de origen de un hipervínculo cuyo destino aún no se ha visitado (:link) o que ya se visitó (:visited).                                     | <a href="#">La pseudo-clase link</a>         |
| E:active<br>E:hover<br>E:focus | Equivale a E durante ciertas acciones del usuario.   | <a href="#">Las pseudo-clases dinámicas</a>  |
| E:lang(c)                      | Equivale a un elemento de tipo E si está en el idioma (humano) c (el lenguaje del documento especifica como se determina el idioma).   | <a href="#">La pseudo-clase :lang()</a>      |
| E + F                          | Equivale a cualquier elemento F precedido inmediatamente por un elemento E.  | <a href="#">Selectores de adyacentes</a>     |
| E[foo]                         | Equivale a cualquier elemento E con el atributo "foo" asignado (cualquier valor).  | <a href="#">Selectores de atributos</a>      |
| E[foo="warning"]               | Equivale a cualquier elemento E cuyo atributo "foo" tiene un valor exactamente igual a "warning".  | <a href="#">Selectores de atributos</a>      |
| E[foo~="warning"]              | Equivale a cualquier elemento E cuyo atributo "foo" tiene un valor consistente en una lista de valores separados por espacios, uno de los cuales es exactamente igual a "warning". | <a href="#">Selectores de atributos</a>      |
| E[lang]="en"]                  | Equivale a cualquier elemento E cuyo atributo "lang" tiene una lista de valores separados por guiones que comienzan (desde la izquierda) con "en".                                 | <a href="#">Selectores de atributos</a>      |
| DIV.warning                    | <i>Sólo HTML</i> . Igual que DIV[class~="warning"].  | <a href="#">Selectores de clase</a>          |
| E#myid                         | Equivale a cualquier elemento E cuyo ID es igual a "myid".   | <a href="#">Selectores de ID</a>             |

## 5.2 Sintaxis de los selectores

Un *selector simple* es tanto un [selector de tipos](#) como un [selector universal](#) seguido inmediatamente por cero o más [selectores de atributos](#), [selectores de ID](#) o [pseudo-clases](#), en cualquier orden. El selector simple es equivalente si todos sus componentes son equivalentes.

Un *selector* es una cadena de uno o más selectores simples separados por combinadores. Los *combinadores* son: espacio en blanco, ">" y "+". El espacio en blanco puede aparecer entre un combinador y los selectores simples que están próximos.

Los elementos de la estructura del documento que equivalen a un selector son llamados *sujetos* del selector. Un selector que consiste en un solo selector simple se corresponde con cualquier elemento que satisface sus requisitos. Anteponiendo un selector simple y un combinador a una cadena se imponen restricciones adicionales para la equivalencia, de este modo los sujetos de un selector son siempre un sub-conjunto de los elementos que equivalen al selector simple del extremo derecho.

Un [pseudo-elemento](#) puede añadirse al último selector simple en una cadena, en cuyo caso la información del estilo se aplica a la sub-parte de cada sujeto.

## 5.2.1 Agrupamiento

Cuando varios selectores comparten las mismas declaraciones, pueden agruparse en una lista separada por comas.

En este ejemplo, condensamos tres reglas con idénticas declaraciones en una sola. Así,

```
H1 { font-family: sans-serif }
H2 { font-family: sans-serif }
H3 { font-family: sans-serif }
```

es equivalente a:

```
H1, H2, H3 { font-family: sans-serif }
```

CSS ofrece también otros mecanismos "resumidos", incluyendo las [declaraciones múltiples](#) y las [propiedades resumidas](#).

## 5.3 Selector universal

El selector universal, escrito "\*", se corresponde con el nombre de cualquier tipo de elemento. Es equivalente con cualquier elemento en la [estructura del documento](#).

Si el selector universal no es el único componente de un [selector simple](#), el "\*" puede omitirse. Por ejemplo:

- \*[LANG=fr] y [LANG=fr] son lo mismo.
- \*.warning y .warning son lo mismo.
- \*#myid y #myid son lo mismo.

## 5.4 Selectores de tipos

Un *selector de tipo* se corresponde con el nombre de un tipo de elemento en el lenguaje del documento. Un selector de tipo equivale con cada instancia del tipo de elemento en la estructura del documento.

La siguiente regla equivale a todos los elementos H1 de la estructura del documento:

```
H1 { font-family: sans-serif }
```

## 5.5 Selectores de descendientes

A veces, los autores pueden querer selectores que se correspondan con un elemento que es el descendiente de otro elemento en la estructura del documento (ej., "Que equivalga a aquellos elementos EM que están contenidos en un elemento H1"). Los selectores de descendientes expresan tal relación dentro de un patrón. Un selector de descendiente se compone de dos o más selectores separados por un [espacio en blanco](#). Un selector de descendiente de la forma "A B" equivale cuando un elemento B en un descendiente arbitrario de algún elemento [antepasado](#) A.

Por ejemplo, considere las siguientes reglas:

```
H1 { color: red }
EM { color: red }
```

Aunque la intención de estas reglas es agregar énfasis al texto cambiando su color, el efecto se perderá en un caso como:

```
<H1>El título es <EM>muy</EM> importante</H1>
```

Resolvemos este caso complementando las reglas anteriores con una regla que asigna un color azul al texto siempre que un EM aparezca en cualquier lugar dentro de un H1:

```
H1 { color: red }
EM { color: red }
H1 EM { color: blue }
```

La tercer regla se corresponderá con EM en el siguiente fragmento:

```
<H1>Este <SPAN class="myclass">título es <EM>muy</EM>
importante</SPAN></H1>
```

El siguiente selector:

```
DIV * P
```

equivale a un elemento P que es nieto o descendiente aún más lejano de un elemento DIV. Observe el uso opcional del espacio en blanco a ambos lados del "\*".

El selector en la siguiente regla, que combina selectores de descendientes con [selectores de atributos](#), equivalen a cualquier elemento que (1) tiene asignado el atributo "href" y (2) está dentro de un P que a su vez está dentro de un DIV:

```
DIV P *[href]
```

## 5.6 Selectores de hijos

Un *selector de hijo* equivale cuando un elemento es el [hijo](#) de algún elemento. Un selector de hijo se compone de dos o más selectores separados por ">".

La siguiente regla asigna el estilo de todos los elementos P que son hijos de BODY:

```
BODY > P { line-height: 1.3 }
```

El siguiente ejemplo combina selectores de descendientes con selectores de hijos:

```
DIV OL>LI P
```

Equivale a un elemento P que es descendiente de un LI; el elemento LI debe ser hijo de un elemento OL; el elemento OL debe ser descendiente de un DIV. Observe que el espacio en blanco opcional alrededor del combinador ">" ha sido descartado.

Por información acerca de la selección del primer hijo de un elemento, por favor, ver la sección acerca de la pseudo-clase [:first-child](#) abajo.

## 5.7 Selectores de hermanos adyacentes

Los selectores de hermanos adyacentes tiene la siguiente sintaxis: E1 + E2, donde E2 es el sujeto del selector. El selector equivale si E1 y E2 comparten el mismo padre en la estructura del documento y E1 precede inmediatamente a E2.

En algunos contextos, los elementos adyacentes generan objetos con formato cuya presentación es operada automáticamente (ej., cerrando los márgenes verticales entre cajas adyacentes). El selector "+" permite a los autores especificar un estilo adicional para los elementos adyacentes.

De este modo, la siguiente regla establece que cuando un elemento P sigue inmediatamente a un elemento MATH, no debe tener sangría:

```
MATH + P { text-indent: 0 }
```

El siguiente ejemplo reduce el espacio vertical que separa un H1 y un H2 que lo sigue inmediatamente:

```
H1 + H2 { margin-top: -5mm }
```

La siguiente regla es similar a la del ejemplo anterior, excepto que se agrega un selector de atributo. Así, el formato especial ocurre sólo cuando H1 tiene especificado `class="opener"`:

```
H1.opener + H2 { margin-top: -5mm }
```

## 5.8 Selectores de atributos

CSS2 permite a los autores especificar reglas que sean equivalentes con atributos definidos en el documento fuente.

## 5.8.1 Atributos equivalentes y valores de los atributos

Los selectores de atributos puede equivaler de cuatro maneras:

- [att]  
Equivale con los elementos que tienen asignado el atributo "att", cualquiera sea el valor del atributo.
- [att=val]  
Equivale cuando el valor del atributo "att" del elemento es exactamente "val".
- [att~=val]  
Equivale cuando el valor del atributo "att" del elemento es una lista de "palabras" separadas por espacios, una de las cuales es exactamente "val". Si se usa este selector, las palabras en el valor no deben contener espacios (ya que ellas están separadas por espacios).
- [att|=val]  
Equivale cuando el valor del atributo "att" del elemento es una lista de "palabras" separadas por guiones, comenzando con "val". La correspondencia siempre comienza al principio del valor del atributo. Esto está pensado principalmente para permitir equivalencias con el sub-código del lenguaje (ej., el atributo "lang" en HTML) tal como se describe en RFC 1766 ([RFC1766]).

Los valores de los atributo deben ser identificadores o cadenas. La distinción entre mayúsculas/minúsculas en los nombres y valores de los atributos de los selectores depende del lenguaje del documento.

Por ejemplo, el siguiente selector de atributo equivale a todos los elementos H1 que especifican el atributo "title", cualquiera sea su valor:

```
H1[title] { color: blue; }
```

En el ejemplo siguiente, el selector equivale a todos los elementos SPAN cuyo atributo "class" tiene exactamente el valor "ejemplo":

```
SPAN[class=ejemplo] { color: blue; }
```

Múltiples selectores de atributos pueden ser usados para referirse a varios atributos de un elemento o, incluso, varias veces al mismo atributo.

Aquí, el selector equivale a todos los elementos SPAN cuyo atributo "hola" tiene exactamente el valor "Cleveland" y cuyo atributo "adios" tiene exactamente el valor "Columbus":

```
SPAN[hola="Cleveland"][adios="Columbus"] { color: blue; }
```

Los siguientes selectores ilustran la diferencia entre "=" y "~=". El primer selector equivale, por ejemplo, al valor "copyright copyleft copyeditor" para el atributo "rel". El segundo selector sólo será equivalente cuando el atributo "href" tenga el valor "http://www.w3.org/".

```
A[rel~="copyright"]  
A[href="http://www.w3.org/"]
```

La siguiente regla esconde a todos los elementos cuyo valor del atributo "lang" es "fr" (es decir, la lengua es el francés).

```
*[LANG=fr] { display : none }
```

La siguiente regla será equivalente a valores del atributo "lang" que empiecen con "en", incluyendo "en", "en-US" y "en-cockney":

```
*[LANG|= "en"] { color : red }
```

De modo similar, la siguiente regla de una hoja de estilo auditiva permite leer un escrito en voz alta con voces diferentes para cada rol:

```
DIALOGUE[character=romeo]  
  { voice-family: "Lawrence Olivier", charles, male }  
  
DIALOGUE[character=julietta]  
  { voice-family: "Vivien Leigh", victoria, female }
```



## 5.8.2 Valores predeterminados de los atributos en la DTD

Las correspondencias tienen lugar en los valores de los atributos en la estructura del documento. Para lenguajes de documento fuera de HTML, los valores predeterminados de los atributos pueden definirse en una DTD o en cualquier otra parte. Las hojas de estilo deben diseñarse de manera que trabajen aún cuando los valores predeterminados no estén incluidos en la estructura del documento.

**Nota de la traducción:**  
DTD es el acrónimo de Document Type Definition (Definición del tipo de documento) y consiste en la definición de la sintaxis de la construcción de marcas que utiliza el documento.

Por ejemplo, considere un elemento EJEMPLO con un atributo "notación" que tiene un valor predeterminado como "decimal". El fragmento DTD podría ser

```
<!ATTLIST EJEMPLO notación (decimal,octal) "decimal">
```

Si la hoja de estilo contiene las reglas

```
EJEMPLO[notación=decimal] { /*... opciones de la propiedad predeterminada ...*/ }
EJEMPLO[notación=octal] { /*... otras opciones ...*/ }
```

entonces para tomar los casos en los cuales este atributo es establecido por defecto y no explícitamente, podría agregarse la siguiente regla:

```
EJEMPLO { /*... opciones de la propiedad predeterminada ...*/ }
```

Debido a que este selector es menos específico que un selector de atributo, deberá ser usado para el caso predeterminado. Debe tomarse la precaución de que todos los otros valores del atributo que no tengan el mismo estilo que el predeterminado sean explícitamente cubiertos.

## 5.8.3 Selectores de clases

Para las hojas de estilo usadas con HTML, los autores pueden usar el punto (.) como una alternativa a "~=" cuando la correspondencia sea con el atributo "class". Así, en HTML, "DIV.valor" y "DIV[class~=valor]" tiene el mismo significado. El valor del atributo debe seguir inmediatamente al ".".

Por ejemplo, podemos asignar información de estilo a todos los elementos con class~="pastoral" como sigue:

```
*.pastoral { color: green } /* todos los elementos con class~=pastoral */
o simplemente
.pastoral { color: green } /* todos los elementos con class~=pastoral */
```

El siguiente sólo asigna estilo a los elementos H1 con class~="pastoral":

```
H1.pastoral { color: green } /* elementos H1 con class~=pastoral */
```

Dadas estas reglas, la primera instancia de H1 abajo no tendría el texto en verde, mientras que la segunda sí:

```
<H1>Verde, no</H1>
<H1 class="pastoral">Muy verde</H1>
```

Para equivaler a un subconjunto de valores "class", cada valor debe ir precedido por un ".", en cualquier orden.

Por ejemplo, la siguiente regla equivale a cualquier elemento P a cuyo atributo "class" le ha sido asignado una lista de valores separados por espacios que incluyen "pastoral" y "marino":

```
P.pastoral.marino { color: green }
```

Esta regla equivale en el caso de class="pastoral rojo azul marino" pero no se corresponde en class="pastoral rojo".

**Nota.** CSS le otorga tanto poder al atributo "class", que los autores podrían conceptualmente diseñar su propio "lenguaje del documento" basados en elementos con casi ninguna presentación asociada (tales como DIV y SPAN en HTML) y asignando información acerca del estilo a través del atributo "class". Los autores deben evitar esta práctica ya que los elementos estructurales de un lenguaje de documento comúnmente tienen significados aceptados y reconocidos mientras que las clases definidas por el autor no pueden tenerlos.

## 5.9 Selectores de ID

Los lenguajes del documento pueden contener atributos que se declaran como del tipo ID. Lo que hace especial a un atributo del tipo ID es que dos de tales atributos no pueden tener el mismo valor; cualquiera sea el lenguaje del documento, un atributo ID puede ser usado para identificar a su elemento de manera excluyente. En HTML todos los atributos ID son denominados "id"; las aplicaciones XML pueden nombrar de manera diferente a los atributos ID, pero se aplica la misma restricción.

El atributo ID de un lenguaje de documento le permite a los autores asignar un identificador a una instancia del elemento dentro de la estructura del documento. Los selectores ID de CSS equivalen a una instancia del elemento basado en su identificador. Un selector ID de CSS contiene un "#" seguido inmediatamente por el valor de ID.

El siguiente selector de ID equivale al elemento H1 cuyo atributo ID tiene el valor "capitulo1":

```
H1#capitulo1 { text-align: center }
```

En el siguiente ejemplo, la regla de estilo equivale al elemento que tiene el valor de ID "z98y". La regla será, de este modo, equivalente al elemento P:

```
<HEAD>
  <TITLE>Equivalencia con P</TITLE>
  <STYLE type="text/css">
    *#z98y { letter-spacing: 0.3em }
  </STYLE>
</HEAD>
<BODY>
  <P id=z98y>Texto espaciado</P>
</BODY>
```

En el ejemplo que sigue, sin embargo, la regla de estilo sólo equivale a un elemento H1 que tiene un valor ID de "z98y". La regla no será equivalente con el elemento P en este ejemplo:

```
<HEAD>
  <TITLE>Equivalencia con H1 solamente</TITLE>
  <STYLE type="text/css">
    H1#z98y { letter-spacing: 0.5em }
  </STYLE>
</HEAD>
<BODY>
  <P id=z98y>Texto espaciado</P>
</BODY>
```

Los selectores de ID tiene una precedencia mayor que los selectores de atributos. Por ejemplo, en HTML, el selector #p123 es más específico que [ ID=p123 ] en términos de [cascada](#).

**Nota.** En XML 1.0 [XML10], la información acerca de qué atributos contiene el ID de un elemento está contenida en una DTD. Cuando analiza XML, las AU no siempre leen la DTD, y así no pueden saber qué es el ID de un elemento. Si el diseñador de una hoja de estilo conoce o sospecha que éste será el caso, deberá usar en cambio los selectores de atributos normales: [name=p371] en lugar de #p371. Sin embargo, el orden en la cascada de los selectores de atributos normales es diferente de los selectores de ID. Puede llegar a ser necesario agregar una prioridad "!important" a las declaraciones: [name=p371] {color: red ! important}. Por supuesto, los elementos de los documentos XML 1.0 sin una DTD no tiene ID en absoluto.

## 5.10 Pseudo-elementos y pseudo-clases

En CSS2, el estilo se atribuye a un elemento basado en su posición dentro de la [estructura del documento](#). Este sencillo modelo es suficiente para muchos casos, pero algunos temas comunes en la edición pueden no ser posibles por medio de la [estructura del documento](#). Por ejemplo, en HTML 4.0 (ver [HTML40](#)), ningún elemento se refiere a la primera línea de una párrafo y, por consiguiente, ningún selector CSS simple puede referirse a ella.

CSS introduce los conceptos de *pseudo-elementos* y *pseudo-clases* para permitir aplicar el formato basado en información que está fuera de la estructura del documento.

- Los pseudo-elementos crean abstracciones acerca de la estructura del documento más allá de aquellas especificadas por el lenguaje del documento. Por ejemplo, los lenguajes de documento no ofrecen mecanismos para acceder a la primera letra o a la primera línea del contenido de un elemento. Los pseudo-elementos de CSS permiten a los diseñadores de hojas de estilo referirse a esta información inaccesible por otros medios. Los pseudo-elementos también pueden proporcionar a los diseñadores de hojas de estilo un modo de asignar estilos a un elemento que no existe en el documento fuente (ej., los pseudo-elementos `:before` y `:after` permite acceder al contenido generado).
- Las pseudo-clases clasifican a los elementos basado en características más allá de su nombre, atributos o contenido; en principio, atributos que no pueden deducirse de la estructura del documento. Las pseudo-clases pueden ser dinámicas, en el sentido de que un elemento puede adquirir o perder una pseudo-clase a medida que el usuario interactúa con el documento. La excepción es `:first-child`, que *puede* deducirse de la estructura del documento.

Ni los pseudo-elementos ni las pseudo-clases aparecen en el documento fuente o la estructura del documento.

Las pseudo-clases son permitidas en cualquier lugar dentro de un selector mientras que los pseudo-elementos sólo pueden aparecer después del [sujeto](#) del selector.

Los nombres de los pseudo-elementos y las pseudo-clases no hacen distinción entre mayúsculas y minúsculas.

Algunas pseudo-clases son mutuamente excluyentes, mientras otras pueden aplicarse simultáneamente al mismo elemento. En caso de reglas conflictivas, el [orden de cascada](#) normal determina el resultado.

[Las aplicaciones del usuario de conformidad con HTML](#) pueden [ignorar](#) todas las reglas con `:first-line` o `:first-letter` en el selector o, alternativamente, puede dar soporte sólo a un subconjunto de las propiedades en estos pseudo-elementos.

## 5.11 Pseudo-clases

### 5.11.1 La pseudo-clase `:first-child`

La pseudo-clase `:first-child` equivale a un elemento que es el primer hijo de algún otro elemento.

En el ejemplo siguiente, el selector equivale a cualquier elemento `P` que sea el primer hijo de un elemento `DIV`. La regla elimina el sangrado del primer párrafo de un `DIV`:

```
DIV > P:first-child { text-indent: 0 }
```

Este selector sería equivalente a `P` dentro de `DIV` en el siguiente fragmento:

```
<P> El último P antes de la nota.
<DIV class="nota">
  <P> El primer P dentro de la nota.
</DIV>
```

pero no se correspondería con el segundo `P` en el siguiente fragmento:

```
<P> El último P antes de la nota.
<DIV class="nota">
  <H2>Nota</H2>
  <P> El primer P dentro de la nota.
</DIV>
```

La siguiente regla especifica 'bold' para el peso de la fuente de cualquier elemento `EM` que es algún descendiente de un elemento `P` que es un primer hijo:

```
P:first-child EM { font-weight : bold }
```

Observe que toda vez que las cajas [anónimas](#) no forman parte de la estructura del documento, no se cuentan en el cálculo del primer hijo.

Por ejemplo, el `EM` en:

```
<P>abc <EM>default</EM>
```

es el primer hijo de `P`.

Los siguientes dos selectores son equivalentes:

```
* > A:first-child    /* A es el primer hijo de cualquier elemento */
A:first-child        /* Igual */
```

## 5.11.2 Las pseudo-clases vínculo: :link y :visited

Las aplicaciones del usuario normalmente muestran los vínculos no visitados de un modo diferenciado de aquellos previamente visitados. CSS proporciona las pseudo-clases ':link' y ':visited' para distinguirlos:

- La pseudo-clase :link se aplica a los vínculos que aún no han sido visitados.
- La pseudo-clase :visited se aplica una vez que el vínculo ha sido visitado por el usuario.

**Nota.** Después de cierta cantidad de tiempo, las aplicaciones del usuario pueden elegir devolver un vínculo visitado a la condición de (no visitado) ':link'.

Los dos estados son mutuamente excluyentes.

El lenguaje del documento determina qué elementos son las anclas de origen de los hipervínculos. Por ejemplo, en HTML 4.0, las pseudo-clases link se aplica a los elementos A con el atributo "href". De este modo, las siguientes dos declaraciones CSS2 tienen un efecto similar:

```
A:link { color: red }
:link  { color: red }
```

Si el siguiente link:

```
<A class="externo" href="http://out.side/">link externo</A>
```

ha sido visitado, esta regla:

```
A.externo:visited { color: blue }
```

provocará que sea azul.

## 5.11.3 Las pseudo-clases dinámicas: :hover, :active y :focus

Las aplicaciones del usuario interactivas a veces cambian el procesamiento en respuesta a las acciones del usuario. CSS proporciona tres pseudo-clases para los casos comunes:

- La pseudo-clase :hover se aplica mientras el usuario señala un elemento (con algún dispositivo para apuntar), pero no lo activa. Por ejemplo, una aplicación del usuario visual puede aplicar esta pseudo-clase cuando el cursor (el puntero del ratón) se mueve por encima de una caja generada por el elemento. Las aplicaciones del usuario que no ofrecen soporte a los [medios interactivos](#) no tienen que apoyar esta pseudo-clase. Algunas aplicaciones del usuario con conformidad que soportan [medios interactivos](#) pueden no ser adecuadas para soportar esta pseudo-clase (ej., lápices ópticos).
- La pseudo-clase :active se aplica mientras un elemento está siendo activado por el usuario. Por ejemplo, el lapso durante el cual el usuario presiona el botón del ratón y lo suelta.
- La pseudo-clase :focus se aplica mientras un elemento tiene el foco (acepta eventos del teclado u otras formas de entrada de texto).

Estas pseudo-clases no son mutuamente excluyentes. Un elemento puede recibir varias de ellas al mismo tiempo.

CSS no define qué elementos pueden estar en los estados anteriores o cómo se ingresa y se sale de los estados. Mediante el uso de scripts se puede decidir si los elementos reaccionan a los eventos del usuarios o no, y los distintos dispositivos y aplicaciones del usuario pueden tener diferentes maneras de apuntar o activar los elementos.

Las aplicaciones del usuario no son obligadas a redibujar el documento en uso a causa de las transiciones de las pseudo-clases. Por ejemplo, un hoja de estilo puede especificar que la propiedad 'font-size' de un vínculo :active debe ser mayor que la de un vínculo inactivo, pero como esto puede ocasionar que las letras cambien de posición cuando el lector selecciona el vínculo, una AU puede ignorar la regla de estilo correspondiente.

```
A:link      { color: red }      /* vínculos no visitados */
A:visited   { color: blue }     /* vínculos visitados */
A:hover     { color: yellow }   /* el usuario señala el vínculo */
A:active    { color: lime }     /* vínculos activos */
```

Observe que A:hover debe ir después que las reglas A:link y A:visited, de otro modo las reglas de cascada ocultarán la propiedad 'color' de la regla A:hover. También, debido a que A:active está ubicada después de A:hover, el color activo (lime) será aplicado cuando el usuario active y señale el elemento A.

Un ejemplo de combinación de pseudo-clases dinámicas:

```
A:focus { background: yellow }
A:focus:hover { background: white }
```

El último selector equivale a los elementos A que se encuentran en la pseudo-clase :focus y en la pseudo-clase :hover.

Para información acerca de la representación del foco en los contornos, consulte por favor la sección sobre [foco dinámico en los contornos](#).

*Nota. En CSS1, la pseudo-clase ':active' era mutuamente excluyente con ':link' y ':visited'. Este ya no es el caso. Un elemento puede ser ambos, ':visited' y ':active' (o ':link' y ':active') y la reglas normales de cascada determinan qué propiedades se aplican.*

### 5.11.4 La pseudo-clase de lenguaje: :lang

Si el lenguaje del documento especifica cómo se determina el lenguaje humano de un elemento, es posible escribir selectores en CSS que coincidan con elementos basado en su lenguaje. Por ejemplo, en HTML [\[HTML40\]](#), el lenguaje se determina por una combinación del atributo "lang", el elemento META y probablemente por la información del protocolo (tal como en los encabezados HTTP). XML utiliza un atributo denominado xml:lang, y puede haber otros métodos específicos del lenguaje del documento para determinar el lenguaje.

La pseudo-clase ':lang(C)' equivale si el elemento está en lenguaje C. Aquí C es un código de lenguaje tal como está especificado en HTML 4.0 [\[HTML40\]](#) y RFC 1766 [\[RFC1766\]](#). Es validado de igual manera que el operador '|='.

Las siguientes reglas determinan las comillas para un documento HTML que está en francés o en alemán:

```
HTML:lang(fr) { quotes: '« ' ' »' }
HTML:lang(de) { quotes: '»' '«' '\2039' '\203A' }
:lang(fr) > Q { quotes: '« ' ' »' }
:lang(de) > Q { quotes: '»' '«' '\2039' '\203A' }
```

El segundo par de reglas especifica realmente la propiedad 'quotes' de los elementos Q de acuerdo al lenguaje de su padre. Esto se hace porque la elección del tipo de comillas se basa típicamente en el lenguaje del elemento que rodea la comilla, no en las propias comillas: como este fragmento de francés "à l'improviste" en medio de un texto en inglés usa el tipo de comillas propias del inglés.

## 5.12 Pseudo-elementos

### 5.12.1 El pseudo-elemento :first-line

El pseudo-elemento :first-line aplica estilos especiales a la primera línea de un párrafo. Por ejemplo:

```
P:first-line { text-transform: uppercase }
```

La regla anterior significa "convertir las letras de la primera línea de cada párrafo en mayúsculas". Sin embargo, el selector "P:first-line" no equivale a ningún elemento HTML real. Se corresponde con un pseudo-elemento que las [aplicaciones del usuario con conformidad](#) insertarán al comienzo de cada párrafo.

Observe que la medida de la primer línea depende de un número de factores, incluyendo el ancho de la página, el tamaño de la fuente, etc. De este modo, un párrafo HTML común como:

```
<P>Este es algún párrafo largo en HTML que será
partido en varias líneas. La primer línea será identificada
por una secuencia ficticia de marcas. Las otras líneas
serán tratadas como líneas comunes dentro del
párrafo.</P>
```

las líneas que resultan partidas como sigue:

```
ESTE ES ALGÚN PÁRRAFO LARGO EN HTML QUE
será partido en varias líneas. La primer línea será
identificada por una secuencia ficticia de marcas.
Las otras líneas serán tratadas como líneas
comunes dentro del párrafo.
```

pueden ser "re-escritas" por las aplicaciones del usuario para incluir la *secuencia ficticia de marcas* para :first-line. Esta secuencia ficticia de marcas ayuda a mostrar cómo se heredan las propiedades.

```
<P><P:first-line> Este es algún párrafo
largo en HTML que</P:first-line> será partido
en varias líneas. La primer línea será identificada
por una secuencia ficticia de marcas. Las otras líneas
serán tratadas como líneas comunes dentro del
párrafo.</P>
```

Si un pseudo-elemento divide un elemento real, el efecto deseado a menudo se consigue con una secuencia ficticia de marcas que cierra y reabre el elemento. De este modo, si marcamos el párrafo anterior con un elemento SPAN:

```
<P><SPAN class="test"> Este es algún párrafo
largo en HTML que será partido en varias líneas.</SPAN>
La primer línea será identificada por una secuencia ficticia
de marcas. Las otras líneas serán tratadas como líneas comunes
dentro del párrafo.</P>
```

la aplicación del usuario puede generar las marcas de inicio y cierre apropiadas para SPAN cuando inserta la secuencia ficticia de marcas para :first-line.

```
<P><P:first-line><SPAN class="test"> Este es algún párrafo
largo en HTML que</SPAN></P:first-line><SPAN class="test">
será partido en varias líneas.</SPAN> La primer línea será
identificada por una secuencia ficticia de marcas. Las otras
líneas serán tratadas como líneas comunes dentro del párrafo.</P>
```

El pseudo-elemento :first-line sólo puede ser aplicado en elementos a nivel de bloque.

El pseudo-elemento :first-line es similar a un elemento a nivel de línea, pero con ciertas restricciones. Sólo las siguientes propiedades se aplican al pseudo-elemento :first-line: [propiedades de la fuente](#), [propiedades del color](#), [propiedades del fondo](#), 'word-spacing', 'letter-spacing', 'text-decoration', 'vertical-align', 'text-transform', 'line-height', 'text-shadow' y 'clear'.

### 5.12.2 El pseudo-elemento :first-letter

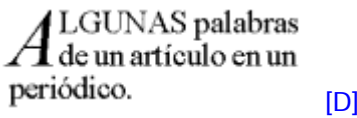
El pseudo-elemento :first-letter puede ser usado para las "capitulares" y "capitulares caídas", que son efectos tipográficos de uso frecuente. Esta clase de letra inicial es similar a un elemento a nivel de línea cuya propiedad 'float' es 'none', de otro modo es similar a un elemento flotante.

Estas son las propiedades que se aplican a los pseudo-elementos :first-letter: [propiedades de la fuente](#), [propiedades del color](#), [propiedades del fondo](#), 'text-decoration', 'vertical-align' (sólo si 'float' es 'none'), 'text-transform', 'line-height', [propiedades de los márgenes](#), [propiedades del relleno](#), [propiedades de los bordes](#), 'float', 'text-shadow' y 'clear'.

La siguiente CSS2 hará una capitular caída que abarca dos líneas:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Capitular caída</TITLE>
    <STYLE type="text/css">
      P
      { font-size: 12pt; line-height: 12pt }
      P:first-letter { font-size: 200%; font-style: italic;
                       font-weight: bold; float: left }
      SPAN
      { text-transform: uppercase }
    </STYLE>
  </HEAD>
  <BODY>
    <P><SPAN>Algunas</SPAN> palabras de un artículo en un periódico.</P>
  </BODY>
</HTML>
```

Este ejemplo sería formateado como sigue:



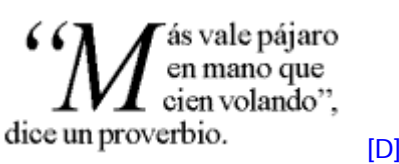
La secuencia ficticia de marcas es:

```
<P>
<SPAN>
<P:first-letter>
A
</P:first-letter>lgunas
</SPAN>
palabras de un artículo en un periódico.
</P>
```

Observe que la marca para el pseudo-elemento :first-letter es contigua al contenido (es decir, el carácter inicial), mientras que la marca de inicio del pseudo-elemento :first-line se inserta después de la marca de inicio del elemento al que está asignado.

Con el propósito de conseguir el formato de las capitulares caídas tradicionales, las aplicaciones del usuario pueden aproximar los tamaños de la fuente, por ejemplo para emparejar las líneas de base. Además, el contorno del signo puede ser tenido en cuenta al establecer el formato.

La puntuación (es decir, los caracteres definidos en Unicode [\[UNICODE\]](#) en las clases de puntuación "open" (Ps), "close" (Pe) y "other" (Po)), que precede a la primera letra debe se incluida, como en:



Los pseudo-elementos :first-letter equivalen a partes de los elementos [a nivel de bloque](#) solamente.

Algunos lenguajes pueden tener reglas específicas acerca de cómo tratar ciertas combinaciones de letras. En alemán, por ejemplo, si la combinación de las letras "ij" aparece al comienzo de una palabra, ambas letras deberían ser consideradas dentro del pseudo-elemento :first-letter.

El siguiente ejemplo ilustra cómo pueden interactuar los pseudo-elementos que se superponen. La primera letra de cada elemento P será verde con una tamaño de fuente de '24pt'. El resto de la primer línea formateada será azul mientras que el resto del párrafo será rojo.

```
P { color: red; font-size: 12pt }
P:first-letter { color: green; font-size: 200% }
P:first-line { color: blue }
```

```
<P>Un texto corto que ocupa dos líneas</P>
```

Asumiendo que el corte de la línea ocurre antes de la palabra "ocupa", la secuencia ficticia de marcas para este fragmento podría ser:

```
<P>
<P:first-line>
<P:first-letter>
U
</P:first-letter>n texto corto que
</P:first-line>
ocupa dos líneas
</P>
```

Observe que el elemento :first-letter está dentro del elemento :first-line. Las propiedades adjudicadas a :first-line son heredadas por :first-letter, pero son anuladas si la misma propiedad es determinada en :first-letter.

### 5.12.3 Los pseudo-elementos :before y :after

Los pseudo-elementos ':before' y ':after' pueden ser usados para insertar un contenido generado antes o después del contenido del elemento. Son explicados en la sección acerca de [texto generado](#).

```
H1:before {content: counter(chapno, upper-roman) ". "}
```



Cuando los pseudo-elementos :first-letter y :first-line son combinados con :before y :after, se aplican a la primera letra o línea del elemento que incluye el texto insertado.

```
P.special:before {content: "Especial! "}  
P.special:first-letter {color: #ffd800}
```

Esto procesará la "E" de "Especial!" en un amarillo oro.

# 6 Asignación de valores a las propiedades, Cascada y Herencia

## 6.1 Valores especificados, computados y reales

Una vez que una aplicación del usuario ha analizado un documento y construido una [estructura del documento](#), debe asignar, para cada elemento de la estructura, un valor por cada propiedad que es aplicada a los [tipos de medios](#) a los cuales se dirige.

El valor final de una propiedad es el resultado de un cálculo en tres pasos: el valor se determina por medio de la especificación (el "valor especificado"), luego resuelto a un valor absoluto si es necesario (el "valor computado"), y finalmente transformado de acuerdo a las limitaciones del entorno local (el "valor real").

### 6.1.1 Valores especificados

Las aplicaciones del usuario deben primero asignar un valor especificado para una propiedad basadas en los siguientes mecanismos (en orden de precedencia):

1. Si la [cascada](#) da como resultado un valor, lo usa.
2. De lo contrario, si la propiedad es [heredada](#), usa el valor del elemento padre, generalmente el valor computado.
3. De otro modo, usa el valor inicial de la propiedad. El valor inicial de cada propiedad es indicado en la definición de la propiedad.

Puesto que no tiene ningún padre, la raíz de la [estructura del documento](#) no puede utilizar valores del elemento padre; en este caso, se usa el valor inicial si es necesario.

### 6.1.2 Valores computados

Los valores especificados pueden ser absolutos (es decir, no se especifican con relación a otro valor, como en 'red' o '2mm') o relativos (es decir, son especificados con relación a otro valor, como en 'auto', '2em' y '12%'). Para los valores absolutos, no es necesario ningún cálculo para encontrar el valor computado.

Los valores relativos, por otro lado, deben ser transformados en valores computados: los porcentajes deben ser multiplicados por un valor de referencia (cada propiedad define qué valor es ése), los valores con unidades relativas (em, ex, px) deben convertirse en absolutos multiplicándolos por el tamaño adecuado de la fuente o el pixel, los valores 'auto' deben ser computados con las fórmulas dadas con cada propiedad, ciertas palabras clave ('smaller', 'bolder', 'inherit') deben ser reemplazadas de acuerdo a sus definiciones.

En la mayoría de los casos, los elementos heredan los valores computados. Hay, sin embargo, algunas propiedades cuyo valor especificado puede heredarse (ej., el valor numérico de la propiedad 'line-height'). En los casos donde los elementos hijos no heredan el valor computado, esto se describe en la definición de la propiedad.

### 6.1.3 Valores reales

Un valor computado está en principio listo para ser usado, pero una aplicación del usuario puede no estar en condiciones de hacer uso del valor en un ámbito determinado. Por ejemplo, una aplicación del usuario sólo puede procesar los bordes con un ancho expresado en pixeles enteros y por consiguiente debe aproximar el ancho computado. El valor real es el valor computado después de cualquier aproximación que le haya sido aplicada.

## 6.2 Herencia

Algunos valores son heredados por los hijos de un elemento en la [estructura del documento](#). Cada propiedad [define](#) si se hereda o no.

Suponga que hay un elemento H1 con un elemento enfatizado (EM) adentro:

```
<H1>El título <EM>es</EM> importante!</H1>
```

Si no se ha asignado ningún color al elemento EM, el énfasis en "es" heredará el color del elemento padre, de modo que si H1 tiene un color azul, el elemento EM será igualmente azul.

Para poner una propiedad de estilo "predeterminada" en un documento, los autores pueden especificar la propiedad en la raíz de la estructura del documento. En HTML, por ejemplo, los elementos HTML o BODY pueden servir para esta función. Observe que esto servirá aún cuando el autor omita la marca BODY en la fuente HTML debido a que el analizador HTML inferirá la marca ausente.

Por ejemplo, como la propiedad 'color' es heredada, todos los descendientes del elemento BODY heredarán el color 'black':

```
BODY { color: black; }
```

Los valores de porcentaje especificados no son heredados; los valores computados sí lo son.

Por ejemplo, dada la siguiente hoja de estilo:

```
BODY { font-size: 10pt }
H1 { font-size: 120% }
```

y este fragmento del documento:

```
<BODY>
  <H1>Un <EM>gran</EM> titular</H1>
</BODY>
```

la propiedad 'font-size' del elemento H1 tendrá el valor computado '12pt' (el 120% de 10pt, el valor del padre). Como el valor computado de 'font-size' es heredado, el elemento EM tendrá el valor computado '12pt' también. Si la aplicación del usuario no tiene la fuente de 12pt disponible, el valor real de 'font-size' para H1 y también para EM podría ser, por ejemplo, '11pt'.

### 6.2.1 El valor 'inherit'

Cada propiedad también puede tener un valor especificado como 'inherit', lo cual significa que, para un elemento determinado, la propiedad toma el mismo [valor computado](#) que la propiedad del padre del elemento. El valor heredado, que normalmente sólo se usa como un valor de respaldo, puede ser reforzado poniendo explícitamente 'inherit'.

#### Nota de la traducción:

En la versión original se aclara que el valor 'inherit' se aplica aún a las propiedades cuyo valor no sería de otro modo heredado.

En el ejemplo de abajo, las propiedades 'color' y 'background' son definidas para el elemento BODY. En todos los demás elementos, el valor de 'color' será heredado y el fondo será transparente. Si estas reglas son parte de la hoja de estilo del usuario, el texto negro sobre un fondo blanco será forzado en todo el documento.

```
BODY {
  color: black !important;
  background: white !important;
}

* {
  color: inherit !important;
  background: transparent;
}
```

## 6.3 La regla @import

La regla '@import' permite a los usuarios importar hojas de estilo desde otras hojas de estilo. Cualquier regla @import debe preceder a todas las reglas especificadas en una hoja de estilo. La palabra clave '@import' debe ir seguida por el URI de la hoja de estilo a incluir. También se permite una cadena; será interpretada como si contuviera url(..) en torno a ella.

Las siguientes líneas son equivalentes en su significado e ilustran ambas sintaxis de '@import' (una con "url()") y otra con sólo una cadena:

```
@import "mystyle.css";
@import url("mystyle.css");
```

Para que las aplicaciones del usuario puedan evitar recuperar recursos para los [tipos de medios](#) no soportados, los autores pueden especificar reglas @import dependientes de los medios. Estas importaciones condicionales especifican los tipos de medios, separados por comas, después del URI.

Las siguientes reglas tienen el mismo efecto que si la hoja de estilo importada hubiera sido cubierta por una regla @media para el mismo medio, pero puede ahorrar a la AU una transmisión infructuosa.

```
@import url("fineprint.css") print;
@import url("bluish.css") projection, tv;
```

En ausencia de cualquier tipo de medios, la importación es incondicional. La especificación de 'all' para el medio tiene el mismo efecto.

## 6.4 Cascada

Las hojas de estilo pueden tener tres orígenes diferentes: el autor, el usuario y la aplicación del usuario.

- **Autor:** El autor especifica las hojas de estilo para un documento fuente de acuerdo a las convenciones del lenguaje del documento. Por ejemplo, en HTML, las hojas de estilo pueden incluirse en el documento o vincularse externamente.
- **Usuario:** El usuario puede especificar información de estilo para un documento particular. Por ejemplo, el usuario puede especificar un archivo que contenga una hoja de estilo o la aplicación del usuario puede proporcionar una interfaz que genere una hoja de estilo del usuario (o comportarse como si lo hubiese hecho).
- **Aplicaciones del usuario:** Las [aplicaciones del usuario con conformidad](#) deben aplicar una *hoja de estilo predeterminada* (o comportarse como si lo hubiesen hecho) previa a todas las hojas de estilo para un documento. La hoja de estilo predeterminada de una aplicación del usuario debe presentar los elementos del lenguaje de documento de un modo que satisfaga las expectativas generales de presentación del lenguaje del documento (ej., en los navegadores visuales, el elemento EM en HTML es presentado usando una fuente itálica). Ver "[Una ejemplo de hoja de estilo para HTML 4.0](#)" por una hoja de estilo preterminada que se recomienda para los documentos HTML 4.0.

Observe que la hoja de estilo predeterminada puede cambiar si las opciones del sistema son modificadas por el usuario (ej., los colores del sistema). Sin embargo, debido a las limitaciones en la implementación interna de las aplicaciones del usuario, puede resultar imposible cambiar los valores en la hoja de estilo predeterminada.

Las hojas de estilo con estos tres orígenes se superpondrán en su acción e interactuarán de acuerdo a la cascada.

La cascada de CSS, asigna una fuerza a cada regla de estilo. Cuando varias reglas se aplican, la de mayor fuerza toma preponderancia.

De forma predefinida, las reglas en las hojas de estilo del autor tienen más fuerza que las reglas en las hojas de estilo del usuario. La preponderancia se revierte, sin embargo, con la reglas "!important". Todas las reglas que el usuario y el autor especifican tienen mayor fuerza que las reglas de la hoja de estilo predeterminada en la AU.

Las hojas de estilo importadas también forman la cascada y su fuerza depende del orden en la importación. Las reglas especificadas en una hoja de estilo dada sustituyen las reglas importadas desde otras hojas de estilo. Las hojas de estilo importadas pueden a su vez importar y sustituir otras hojas de estilo, recursivamente, y se aplican las mismas reglas de preponderancia.

### 6.4.1 Orden de la cascada

Para encontrar el valor para una combinación elemento/propiedad, las aplicaciones del usuario deben aplicar el siguiente orden de disposición:

1. Hallar todas las declaraciones que se aplican al elemento y la propiedad en cuestión, para el [tipo de medio](#) al que está dirigido. Las declaraciones se aplican si el selector asociado [coincide](#) con el elemento en cuestión.
2. La primera disposición de las declaraciones se hace por fuerza y origen: para las declaraciones normales, las hojas de estilo del autor sustituyen las hojas de estilo del usuario que sustituye la hoja de estilo predeterminada. Para las declaraciones "!important", las hojas de estilo del usuario sustituyen las hojas de estilo del autor que sustituyen la hoja de estilo predeterminada. La declaración "!important" sustituye a las declaraciones normales. Una hoja de estilo importada tiene el mismo origen que la hoja de estilo que la ha importado.
3. La segunda disposición es por la [especificidad](#) del selector: los selectores más específicos sustituirán a los más generales. Los pseudo-elementos y las pseudo-clases se cuentan como elementos y clases normales respectivamente.
4. Finalmente, se dispone por el orden especificado: si dos reglas tienen la misma fuerza, origen y especificidad, la última en ser especificada vence. Las reglas en las hojas de estilo importadas se considera que están antes que cualquier regla en la propia hoja de estilo.

Fuera de la definición de "!important" para las declaraciones individuales, esta estrategia le otorga a las hojas de estilo del autor mayor fuerza que a las del lector. Por eso es importante que la aplicación del usuario le brinde al usuario la posibilidad de neutralizar la influencia de determinada hoja de estilo, por ej., a través de un menú desplegable.

### 6.4.2 Las reglas !important

CSS intenta establecer un balance de poder entre las hojas de estilo del autor y del usuario. De modo predefinido, las reglas en una hoja de estilo del autor sustituye las de la hoja de estilo del usuario (ver regla 3 de cascada).

Sin embargo, para balancear, una declaración "important" (las palabras clave "!" e "important" siguiendo a la declaración) toman preponderancia sobre una declaración normal. Ambas hojas de estilo, las del autor y las del usuario, pueden contener declaraciones "important", y las reglas "important" del usuario sustituyen a las reglas "important" del autor. Esta característica mejora la accesibilidad de los documentos dándole a los usuarios con requerimientos especiales (fuentes grandes, combinaciones de colores, etc.) el control sobre la presentación.

**Nota. Este es un cambio semántico sobre CSS1. En CSS1, las reglas "important" del autor tomaban preponderancia sobre las reglas.**

La declaración de una propiedad resumida (ej., 'background') como "important" equivale a declarar todas sus sub-propiedades como "important".

La primer regla en la hoja de estilo del usuario en el ejemplo siguiente contiene una declaración "important", que sustituye a la correspondiente declaración en la hoja de estilo del autor. La segunda declaración también resultará vencedora debido a que a sido marcada como "important". Sin embargo, la tercer regla en la hoja de estilo del usuario no es "important" y entonces perderá con la segunda regla en la hoja de estilo del autor (que definió el estilo en una propiedad resumida). Asimismo, la tercer regla del autor perderá con la segunda regla del autor debido a que la segunda regla es "important". Esto muestra que las declaraciones "important" tienen la misma función también dentro de las hojas de estilo del autor.

```
/* De la hoja de estilo del usuario */
P { text-indent: 1em ! important }
P { font-style: italic ! important }
P { font-size: 18pt }

/* De la hoja de estilo del autor */
P { text-indent: 1.5em !important }
P { font: 12pt sans-serif !important }
P { font-size: 24pt }
```

### 6.4.3 Cálculo de la especificidad de un selector

La especificidad de un selector se calcula como sigue:

- se cuenta el número de atributos ID en el selector (= a)
- se cuenta el número de los otros atributos y pseudo-clases en el selector (= b)
- se cuenta el número de nombres de elementos en el selector (= c)
- se ignoran los pseudo-elementos.

La concatenación de los tres números a-b-c (en un sistema numérico de base mayor) proporciona la especificidad.

Algunos ejemplos:

```
*           {} /* a=0 b=0 c=0 -> especificidad = 0 */
LI          {} /* a=0 b=0 c=1 -> especificidad = 1 */
UL LI       {} /* a=0 b=0 c=2 -> especificidad = 2 */
UL OL+LI    {} /* a=0 b=0 c=3 -> especificidad = 3 */
H1 + *[REL=up] {} /* a=0 b=1 c=1 -> especificidad = 11 */
UL OL LI.red {} /* a=0 b=1 c=3 -> especificidad = 13 */
LI.red.level {} /* a=0 b=2 c=1 -> especificidad = 21 */
#x34y       {} /* a=1 b=0 c=0 -> especificidad = 100 */
```

En HTML, los valores del atributo "style" de un elemento son reglas de la hoja de estilo. Estas reglas no tienen selectores, pero para el propósito del paso 3 del algoritmo de cascada, se considera que tienen un selector ID (especificidad: a=1, b=0, c=0). Para el propósito del paso 4, se considera que están después de todas otras reglas.

```
<HEAD>
<STYLE type="text/css">
  #x97z { color: blue }
</STYLE>
</HEAD>
<BODY>
<P ID=x97z style="color: red">
</BODY>
```

En el ejemplo de arriba, el color del elemento P sería rojo. Aunque la especificidad es la misma para ambas declaraciones, la declaración en el atributo "style" sustituye a la del elemento STYLE debido a la regla 4 de la cascada.

### 6.4.4 Precedencia de las indicaciones de presentación fuera de CSS

La AU puede elegir cumplir con las indicaciones de presentación de otras fuentes aparte de las hojas de estilo, por ejemplo, el elemento FONT o el atributo "align" en HTML. Si es así, las indicaciones de presentación fuera de CSS deben ser traducidas a las correspondientes reglas CSS con una especificidad igual a cero. Se asume que las reglas están al comienzo de la hoja de estilo del autor y pueden ser sustituidas por las reglas subsecuentes de la hoja de estilo

***Nota. En una fase de transición, esta política facilitará que los atributos estilísticos co-existan con las hojas de estilo.***

***Nota. En CSS1, a las indicaciones de presentación fuera de CSS se les otorgaba una especificidad igual a 1, no igual a 0. El cambio se debe a la introducción del selector universal, que tiene una especificidad de 0.***

# 7 Tipos de medios

## 7.1 Introducción a los tipos de medios

Uno de los rasgos más importantes de las hojas de estilo es que especifican cómo debe ser presentado un documento en diferentes medios: en la pantalla, en papel, con un sintetizador de voz, con un dispositivo braille, etc.

Ciertas propiedades CSS son diseñadas sólo para ciertos medios (ej., la propiedad 'cue-before' para las aplicaciones del usuario auditivas). En ocasiones, sin embargo, las hojas de estilo para diferentes tipos de medios pueden compartir una propiedad, pero requieren de valores diferentes para esa propiedad. Por ejemplo, la propiedad 'font-size' es útil para la pantalla y los medios impresos. Sin embargo, los dos medios son lo suficientemente diferentes como para requerir diferentes valores para la propiedad común; un documento necesitará típicamente una fuente mayor en una pantalla de computadora que sobre papel. La experiencia también demuestra que las fuentes sans-serif son más fáciles de leer en pantalla, mientras que las fuentes con serif son más fáciles de leer sobre papel. Por estas razones, es necesario expresar que una hoja de estilo --o una sección de una hoja de estilo-- se aplica a ciertos tipos de medios.

## 7.2 Especificación de hojas de estilos dependientes de los medios

Actualmente hay dos maneras de especificar la dependencia de las hojas de estilos:

- Especificando el medio de destino desde una hoja de estilo con las reglas-arroba @media o @import.

```
@import url("loudvoice.css") aural;
@media print {
  /* la hoja de estilo para impresión va aquí */
}
```

- Especificando el medio de destino dentro del lenguaje del documento. Por ejemplo, en HTML 4.0 ([HTML40]), el atributo "media" en el elemento LINK especifica el medio de destino de una hoja de estilo externa:

- 
- <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
- <HTML>
- <HEAD>
- <TITLE>Vínculo a un medio</TITLE>
- <LINK rel="stylesheet" type="text/css"
- media="print, handheld" href="foo.css">
- </HEAD>
- <BODY>
- <P>El cuerpo...
- </BODY>
- </HTML>

La regla @import es definida en el [capítulo sobre cascada](#).

### 7.2.1 La regla @media

Una regla @media especifica los [tipos de medios](#) de destino (separados por comas) de un juego de reglas (delimitadas por llaves). La construcción @media le permite a la hoja de estilo reglas para distintos medios en la misma hoja de estilo:

```
@media print {
  BODY { font-size: 10pt }
}
@media screen {
  BODY { font-size: 12pt }
}
@media screen, print {
  BODY { line-height: 1.2 }
}
```

## 7.3 Tipos de medios reconocidos

Un *tipo de medio* de CSS designa un juego de propiedades CSS. Una aplicación del usuario que aspira a dar soporte a un tipo de medio por nombre debe implementar todas las propiedades que se aplican a ese tipo de medio.



Los nombres elegidos para los tipos de medios de CSS reflejan los dispositivos de destino para los cuales las propiedades relevantes tienen sentido. En el siguiente listado de tipos de medios de CSS, las descripciones entre paréntesis no son normativas. Solamente brindan un sentido de la clase de dispositivo del tipo de medio a que se hace referencia.

**all**

Apropiado para todos los dispositivos.

**aural**

Propuesto para los sintetizadores de voz. Ver la sección sobre [hojas de estilo auditivas](#) para más detalles.

**braille**

Propuesto para dispositivos táctiles braille.

**embossed**

Propuesto para impresoras de páginas braille.

**handheld**

Propuesto para dispositivos de mano (típicamente de pantalla pequeña, monocromos, de limitado ancho de banda).

**print**

Propuesto para material opaco en páginas y para documentos percibidos en pantalla en modo de vista previa a la impresión. Por favor, consulte la sección sobre [medios paginados](#) por información acerca de los formatos de publicación que son específicos de los medios paginados.

**projection**

Propuesto para las presentaciones proyectadas, por ejemplo, proyectores o transparencias impresas. Por favor, consulte la sección sobre [medios paginados](#) por información acerca de los formatos de publicación que son específicos de los medios paginados.

**screen**

Propuesto primariamente para pantallas de computadora a color.

**tty**

Propuesto para medios que usan una grilla de caracteres de espacio fijo, tales como teletipos, terminales o dispositivos portátiles con limitada capacidad del monitor. Los autores no deben usar las [unidades en pixeles](#) con el tipo de medio "tty".

**tv**

Propuesto para dispositivos de tipo televisión (baja resolución, color, pantalla con limitado desplazamiento, sonido disponible).

Los nombres de los tipos de medios no diferencian entre mayúsculas y minúsculas.

Debido al acelerado cambio de las tecnologías, CSS2 no especifica una lista definitiva de tipos de medios que pueden ser valores para @media.

***Nota. Las futuras versiones de CSS pueden extender esta lista. Los autores no deben contar con nombres de tipos de medios que aún no han sido definidos por una especificación CSS.***

### 7.3.1 Grupos de medios

Cada definición de propiedad en CSS especifica los tipos de medios para los cuales la propiedad debe ser implementada por una [aplicación del usuario con conformidad](#). Como las propiedades generalmente se aplican a varios medios, la sección "Se aplica a los medios" de cada definición de propiedad lista los grupos de medios en vez de los tipos de medio individuales. Cada propiedad se aplica a todos los tipos de medio en los grupos de medios listados en su definición.

CSS2 define los siguientes grupos de medios:

- **contínuos o paginados.** "Ambos" significa que la propiedad en cuestión se aplica a ambos grupos de medios.
- **visuales, auditivos, o táctiles.**
- **grillas** (para dispositivos de grilla de caracteres) o **bitmap.** "Ambos" significa que la propiedad en cuestión se aplica a ambos grupos de medios.
- **interactivos** (para dispositivos que permiten la interacción con el usuario) o **estáticos** (para aquellos que no). "Ambos" significa que la propiedad en cuestión se aplica a ambos grupos de medios.
- **todos** (incluye a todos los tipos de medios)

La siguiente tabla muestra la relación entre los grupos de medios y los tipos de medios:

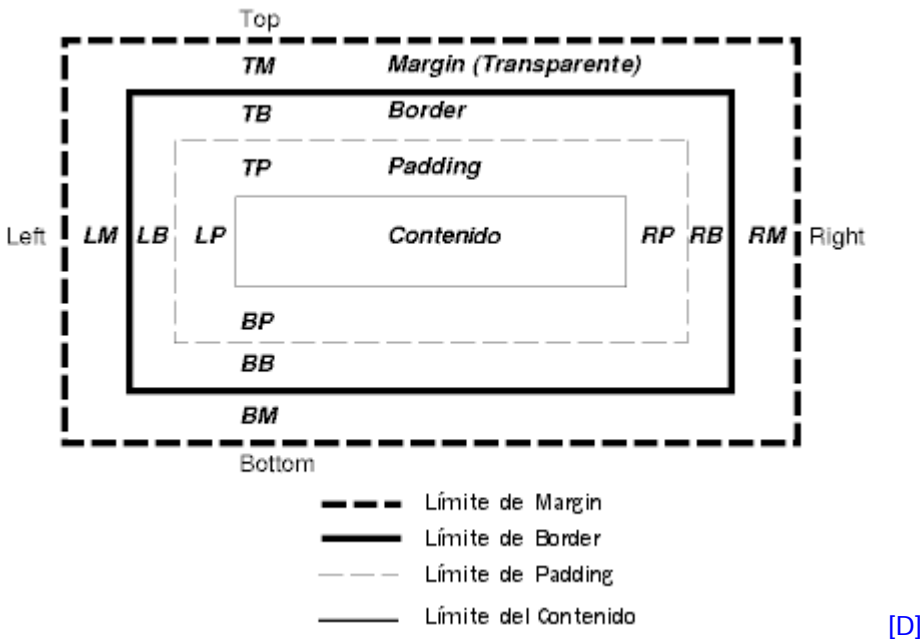
| <i>Relación entre los grupos de medios y los tipos de medios</i> |                     |                             |                |                        |
|--|---------------------|-----------------------------|----------------|------------------------|
| Tipos de Medios  | Grupos de Medios    |                             |                |                        |
|  | contínuos/paginados | visuales/auditivos/táctiles | grillas/bitmap | interactivos/estáticos |
| aural  | contínuos           | auditivos                   | N/A            | ambos                  |
| braille  | contínuos           | táctiles                    | grillas        | ambos                  |
| emboss   | paginados           | táctiles                    | grillas        | ambos                  |
| handheld   | ambos               | visuales                    | ambos          | ambos                  |
| print  | paginados           | visuales                    | bitmap         | estáticos              |
| projection   | paginados           | visuales                    | bitmap         | estáticos              |
| screen   | contínuos           | visuales                    | bitmap         | ambos                  |
| tty  | contínuos           | visuales                    | grillas        | ambos                  |
| tv   | ambos               | visuales, auditivos         | bitmap         | ambos                  |

# 8 Modelo de caja

El modelo de caja de CSS describe las cajas rectangulares que son generadas por los elementos en la [estructura del documento](#) y compuestas de acuerdo al [modelo de formato visual](#). La [caja de página](#) es una clase especial de caja que se describe en detalle en la sección sobre [medios paginados](#).

## 8.1 Dimensiones de la caja

Cada caja tiene un *área de contenido* (ej., texto, una imagen, etc.) y las áreas circundantes opcionales de *padding*, *border* y *margin*; el tamaño de cada área es especificado por las propiedades que se definen abajo. El siguiente diagrama muestra cómo se relacionan estas áreas y la terminología usada para referirse a las partes de *margin*, *border* y *padding*:



[D]

*Margin*, *border* y *padding* pueden ser divididos en los segmentos *left*, *right*, *top* y *bottom* (ej., en el diagrama, "LM" para *left margin*, "RP" para *right padding*, "TB" para *top border*, etc.).

El perímetro de cada una de las cuatro áreas (*contenido*, *padding*, *border* y *margin*) es llamado "límite", de manera que cada caja tiene cuatro límites:

### límite del contenido o límite interno

El límite del contenido rodea al [contenido procesado](#) del elemento.

### límite de padding

El límite de padding (relleno) rodea a la caja de relleno. Si padding tiene un ancho de 0, el límite del relleno es el mismo que el límite de contenido. El límite del relleno de una caja define el límite de la [caja de contención](#) establecida por la caja.

### límite de border

El límite de border (borde) rodea el borde de la caja. Si border tiene un ancho de 0, el límite del borde es el mismo que el límite del relleno (padding).

### límite de margin o límite externo

El límite de margin (margen) rodea el margen de la caja. Si margin tiene un ancho de 0, el límite del margen es el mismo que el límite del borde (border).

Cada límite puede dividirse en límite *left* (izquierdo), *right* (derecho), *top* (superior) y *bottom* (inferior).

Las dimensiones del área del contenido de una caja --el *ancho del contenido* y la *altura del contenido*-- dependen de varios factores: si el elemento que genera la caja tiene asignadas las propiedades 'width' o 'height', si la caja contiene texto u otras cajas, si la caja es una tabla, etc. El ancho y la altura de la caja son discutidos en el capítulo sobre [detalles del modelo de composición visual](#).

El *ancho de la caja* está dado por la suma de los márgenes, bordes y rellenos izquierdos y derechos, y el ancho del contenido. La *altura* está dada por la suma de los márgenes, bordes y rellenos superiores e inferiores, y la altura del contenido.

El estilo del fondo de las distintas áreas de una caja es determinado como sigue:

- *Área del contenido:* La propiedad 'background' del elemento generador.
- *Área del relleno (padding):* La propiedad 'background' del elemento generador.
- *Área del borde (border):* Las [propiedades del borde](#) del elemento generador.
- *Área del margen (margin):* Los márgenes son siempre transparentes.

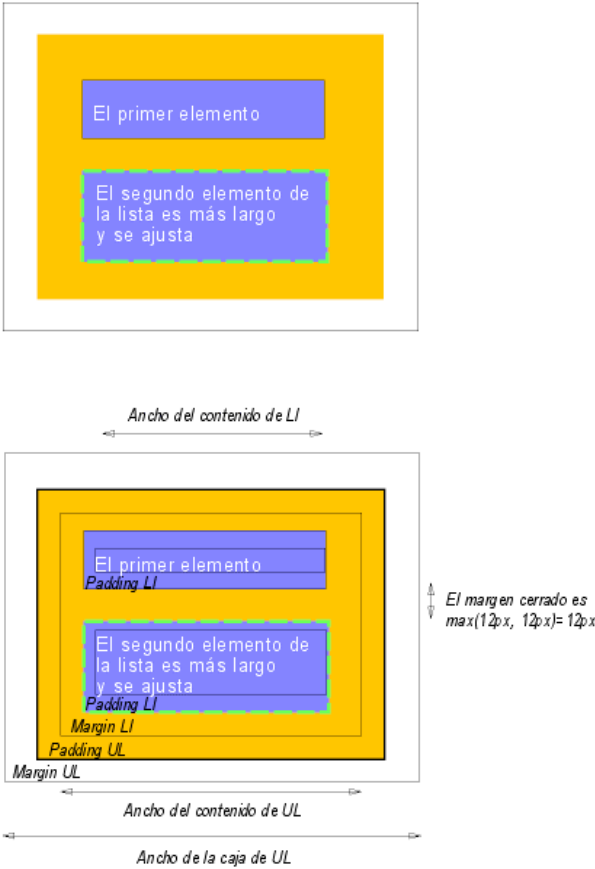
## 8.2 Ejemplo de márgenes, rellenos y bordes

Este ejemplo ilustra cómo interactúan los márgenes, los rellenos y los bordes. El documento HTML de ejemplo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Ejemplo de márgenes, rellenos y bordes</TITLE>
    <STYLE type="text/css">
      UL {
        background: green;
        margin: 12px 12px 12px 12px;
        padding: 3px 3px 3px 3px;
        /* No se especifica border */
      }
      LI {
        color: black;
        background: gray;
        margin: 12px 12px 12px 12px;
        padding: 12px 0px 12px 12px;
        list-style: none
        /* el color del texto es negro */
        /* Contenido, padding será gris */
        /* Note 0px para padding right */
        /* sin viñeta antes de un ítem */
        /* Ningún borde definido */
      }
      LI.withborder {
        border-style: dashed;
        border-width: medium;
        /* pone el ancho de border para todos
los lados */
        border-color: black;
      }
    </STYLE>
  </HEAD>
  <BODY>
    <UL>
      <LI>El primer elemento
      <LI class="withborder">El segundo elemento de la lista es
        más largo y se ajusta.
    </UL>
  </BODY>
</HTML>
```

resulta en una [estructura del documento](#) con (entre otras relaciones) un elemento UL que tiene dos LI hijos.

El primero de los diagramas siguientes ilustra el resultado que produciría este ejemplo. El segundo ilustra la relación entre los márgenes, el relleno y los bordes del elemento UL y la de sus elementos hijos LI.



Observe que:

- El [ancho del contenido](#) de cada caja de LI es calculado de arriba hacia abajo; el [bloque de contención](#) de cada caja LI es establecido por el elemento UL.
- La altura de cada caja LI está dada por su [altura del contenido](#), más los rellenos, bordes y márgenes superiores e inferiores. Observe que los márgenes verticales entre las cajas LI se [cierran](#).
- El relleno derecho de las cajas LI ha sido colocado de un ancho igual a cero (la propiedad 'padding'). El efecto es evidente en la segunda ilustración.
- Los márgenes de las cajas de LI son transparentes --los márgenes siempre lo son-- por eso el color del fondo (green) de las áreas de relleno y contenido de UL se observa a través de ellos.
- El segundo elemento LI especifica un borde cortado (la propiedad 'border-style').

### 8.3 Propiedades del margen: 'margin-top', 'margin-right', 'margin-bottom', 'margin-left' y 'margin'

Las propiedades del margen especifican el ancho del [área del margen](#) de una caja. La propiedad resumida 'margin' determina el margen para los cuatro lados mientras que las otras propiedades sólo determinan su lado respectivo.

Las propiedades definidas en esta sección se refieren al tipo de valor para **<margin-width>**, que puede tomar uno de los siguientes valores:

**<medida>**

Especifica un ancho fijo.

**<porcentaje>**

El porcentaje es calculado con respecto al *ancho* del [bloque de contención](#) de la caja generada. Esto es así para 'margin-top' y 'margin-bottom', excepto en el [contexto de la página](#), donde los porcentajes se refieren a la altura de la caja de la página.

**auto**

Ver la sección sobre [computando anchos y márgenes](#) por su comportamiento.

Los valores negativos para las propiedades de los márgenes son permitidos, pero pueden existir limitaciones específicas de la implementación.

**'margin-top', 'margin-right', 'margin-bottom', 'margin-left'**

*Valor:* <margin-width> | inherit  
*Inicial:* 0  
*Se aplica a:* todos los elementos  
*Se hereda:* no  
*Porcentajes:* se refieren al ancho del bloque de contención  
*Medios:* visuales

Estas propiedades determinan los márgenes superior, derecho, inferior e izquierdo de una caja.

```
H1 { margin-top: 2em }
```

**'margin'**

*Valor:* <margin-width>{1,4} | inherit  
*Inicial:* no definido para las propiedades resumidas  
*Se aplica a:* todos los elementos  
*Se hereda:* no  
*Porcentajes:* se refieren al ancho del bloque de contención  
*Medios:* visuales

La propiedad 'margin' es una propiedad resumida para establecer 'margin-top', 'margin-right', 'margin-bottom', y 'margin-left' en un mismo lugar de la hoja de estilo.

Si hay sólo un valor, se aplica a todos los lados. Si hay dos valores, los márgenes superior e inferior son determinados por el primer valor y los márgenes derecho e izquierdo son determinados por el segundo. Si hay tres valores, el superior es definido por el primer valor, el izquierdo y el derecho son definidos por el segundo, y el inferior es definido por el tercero. Si hay cuatro valores, ellos se aplican al superior, derecho, inferior e izquierdo, respectivamente.

```
BODY { margin: 2em } /* todos los márgenes en 2em */  
BODY { margin: 1em 2em } /* top & bottom = 1em, right & left = 2em */  
BODY { margin: 1em 2em 3em } /* top=1em, right=2em, bottom=3em, left=2em */
```

La última regla del ejemplo de arriba es equivalente al ejemplo de abajo:

```
BODY {
  margin-top: 1em;
  margin-right: 2em;
  margin-bottom: 3em;
  margin-left: 2em;           /* copiado del lado opuesto (right) */
}
```

8.3.1 Márgenes cerrados

En esta especificación, la expresión *márgenes cerrados* significa que los márgenes adyacentes (ningún área de relleno o de borde los separa) de dos o más cajas (que pueden estar una al lado de la otra o anidadas) se combinan para formar un solo margen.

En CSS2, los márgenes horizontales nunca se cierran.

Los márgenes verticales pueden cerrarse entre ciertas cajas:

- Dos o más márgenes verticales adyacentes de cajas de bloques en el flujo normal se cierran. El ancho del margen resultante es el máximo de los anchos de los márgenes adyacentes. En el caso de márgenes negativos, el máximo absoluto de los márgenes adyacentes negativos es restado del máximo de los márgenes adyacentes positivos. Si no hay ningún margen positivo, el máximo absoluto de los márgenes adyacentes negativos es restado de cero.
- Los márgenes verticales entre una caja flotante y cualquier otra caja no se cierran.
- Los márgenes de cajas con posiciones absoluta y relativa no se cierran.

Por favor, consulte los ejemplos de márgenes, rellenos y bordes por una ilustración de márgenes cerrados.

8.4 Propiedades del relleno: 'padding-top', 'padding-right', 'padding-bottom', 'padding-left' y 'padding'

Las propiedades del relleno especifican el ancho del área de relleno de una caja. La propiedad resumida 'padding' define el relleno para los cuatro lados mientras que las otras propiedades de relleno sólo definen sus respectivos lados.

Las propiedades definidas en esta sección se refieren al tipo de valor de <padding-width>, que puede tomar uno de los siguientes valores:

- <medida>  
Especifica una medida fija.
- <porcentaje>  
El porcentaje es calculado con respecto al ancho del bloque de contención de la caja generada, aún para 'padding-top' y 'padding-bottom'.

A diferencia de las propiedades del margen, los valores para los valores de relleno no pueden ser negativos. Al igual que las propiedades del margen, los valores de porcentaje para las propiedades del relleno se refieren al ancho del bloque de contención de la caja generada.

| 'padding-top', 'padding-right', 'padding-bottom', 'padding-left' |
|--|
| Valor: <padding-width>   inherit                                 |
| Inicial: 0   |
| Se aplica a: todos los elementos                                 |
| Se hereda: no  |
| Porcentajes: se refieren al ancho del bloque de contención       |
| Medios: visuales   |

Estas propiedades determinan el relleno superior, derecho, inferior e izquierdo de una caja.

```
BLOCKQUOTE { padding-top: 0.3em }
```

| 'padding'  |
|--|
| Valor: <padding-width>{1,4}   inherit                      |
| Inicial: no definido para las propiedades resumidas        |
| Se aplica a: todos los elementos                           |
| Se hereda: no  |
| Porcentajes: se refieren al ancho del bloque de contención |
| Medios: visuales   |

La propiedad 'padding' es una propiedad resumida para definir 'padding-top', 'padding-right', 'padding-bottom' y 'padding-left' en un mismo lugar en la hoja de estilo.

Si hay sólo un valor, se aplica a todos los lados. Si hay dos valores, los rellenos superior e inferior son determinados por el primer valor y los rellenos derecho e izquierdo son determinados por el segundo. Si hay tres

valores, el superior es definido por el primer valor, el izquierdo y el derecho son definidos por el segundo, y el inferior es definido por el tercero. Si hay cuatro valores, ellos se aplican al superior, derecho, inferior e izquierdo, respectivamente.

El color de la superficie o la imagen del área de relleno es especificado a través de la propiedad 'background':

```
H1 {
  background: white;
  padding: 1em 2em;
}
```

El ejemplo de arriba especifica un relleno vertical igual a '1em' ('padding-top' y 'padding-bottom') y un relleno horizontal de '2em' ('padding-right' y 'padding-left'). La unidad 'em' es *relativa* al tamaño de la fuente del elemento: '1em' es igual al tamaño de la fuente en uso.

## 8.5 Propiedades del borde

Las propiedades del borde especifica el ancho, color y estilo del *área del borde* de una caja. Estas propiedades se aplican a todos los elementos.

*Nota. Particularmente para HTML, las aplicaciones del usuario pueden procesar los bordes de ciertos elementos (ej., botones, menús, etc.) de manera diferente a los elementos "ordinarios".*

### 8.5.1 Ancho del borde: 'border-top-width', 'border-right-width', 'border-bottom-width', 'border-left-width' y 'border-width'

Las propiedades del ancho del borde especifican la medida del *área del borde*. Las propiedades definidas en esta sección se refieren al tipo de valor de **<border-width>**, que puede tomar uno de los siguientes valores:

- thin**  
Un borde fino.
- medium**  
Un borde mediano.
- thick**  
Un borde grueso.
- <medida>**  
El grosor del borde tiene un valor explícito. Las dimensiones del borde explícitas no pueden ser negativas.

La interpretación de los primeros tres valores depende de la aplicación del usuario. Sin embargo, se deben mantener las siguientes relaciones:

'thin' <= 'medium' <= 'thick'.

Además, estos grosores deben mantenerse constantes a lo largo de todo un documento.

| 'border-top-width', 'border-right-width', 'border-bottom-width', 'border-left-width' |                          |
|--|--------------------------|
| Valor:   | <border-width>   inherit |
| Inicial:   | medium                   |
| Se aplica a:   | todos los elementos      |
| Se hereda:   | no                       |
| Porcentajes:   | N/A                      |
| Medios:  | visuales                 |

Estas propiedades determinan al ancho de los bordes superior, derecho, inferior e izquierdo de una caja.

| 'border-width' |                                  |
|----------------|----------------------------------|
| Valor:         | <border-width>{1,4}   inherit    |
| Inicial:       | ver las propiedades individuales |
| Se aplica a:   | todos los elementos              |
| Se hereda:     | no                               |
| Porcentajes:   | N/A                              |
| Medios:        | visuales                         |

Esta propiedad es una propiedad resumida para especificar 'border-top-width', 'border-right-width', 'border-bottom-width' y 'border-left-width' en un mismo lugar en la hoja de estilo.

Si hay sólo un valor, se aplica a todos los lados. Si hay dos valores, los bordes superior e inferior son determinados por el primer valor y los bordes derecho e izquierdo son determinados por el segundo. Si hay tres valores, el superior es definido por el primer valor, el izquierdo y el derecho son definidos por el segundo, y el inferior es definido por el tercero. Si hay cuatro valores, ellos se aplican al superior, derecho, inferior e izquierdo, respectivamente.



En los ejemplos de abajo, los comentarios indican los grosores resultantes en los bordes superior, derecho, inferior e izquierdo:

```
H1 { border-width: thin } /* thin thin thin thin */
H1 { border-width: thin thick } /* thin thick thin thick */
H1 { border-width: thin thick medium } /* thin thick medium thick */
```

### 8.5.2 Color del borde: 'border-top-color', 'border-right-color', 'border-bottom-color', 'border-left-color' y 'border-color'

Las propiedades del color del borde especifican el color del borde de una caja.

**'border-top-color', 'border-right-color', 'border-bottom-color', 'border-left-color'**

*Valor:* <color> | inherit  
*Inicial:* el valor de la propiedad 'color'  
*Se aplica a:* todos los elementos  
*Se hereda:* no  
*Porcentajes:* N/A  
*Medios:* visuales

**'border-color'**

*Valor:* <color>{1,4} | transparent | inherit  
*Inicial:* ver las propiedades individuales  
*Se aplica a:* todos los elementos  
*Se hereda:* no  
*Porcentajes:* N/A  
*Medios:* visuales

La propiedad 'border-color' determina el color de los cuatro bordes. Los valores tienen los siguientes significados:

- <color>**  
Especifica un valor de color.
- transparent**  
El borde es transparente (no obstante puede tener grosor).

La propiedad 'border-color' puede tener de uno a cuatro valores, y los valores son aplicados a los distintos lados como en 'border-width'.

Si el color del borde de un elemento no es especificado con una propiedad del borde, las aplicaciones del usuario deben tomar el valor de la propiedad 'color' del elemento como el [valor computado](#) para el color del borde.

En este ejemplo, el borde será una línea negra sólida.

```
P {
  color: black;
  background: white;
  border: solid;
}
```

### 8.5.3 Estilo del borde: 'border-top-style', 'border-right-style', 'border-bottom-style', 'border-left-style' y 'border-style'

Las propiedades del estilo del borde especifican el estilo de la línea del borde de una caja (sólida, doble, punteada, etc.). Las propiedades definidas en esta sección se refieren al tipo de valor de **<border-style>**, que puede estar constituido por uno de los siguientes:

- none**  
Ningún borde. Este valor fuerza al valor computado de 'border-width' a '0'.
- hidden**  
Igual a 'none', excepto en términos de [resolución de conflictos de bordes](#) para los [elementos de tabla](#).
- dotted**  
El borde es una serie de puntos.
- dashed**  
El borde es una serie de pequeños segmentos de línea.
- solid**  
El borde es un único segmento de línea.
- double**  
El borde son dos líneas sólidas. La suma de las dos líneas y el espacio entre ellas es igual al valor de 'border-width'.
- groove**  
El borde luce como si estuviese tallado en el lienzo.

ridge

Lo opuesto a 'grove': el borde parece que estuviera sobresaliendo del lienzo.

inset

El borde hace que toda la caja luzca como si estuviera empotrada en el lienzo.

outset

Lo opuesto a 'inset': el borde hace que toda la caja parezca sobresalir del lienzo.

Todos los bordes son dibujados por encima del fondo de la caja. El color de los bordes con valores de 'groove', 'ridge', 'inset' y 'outset' depende de la propiedad 'color' del elemento.

Las aplicaciones de usuario HTML con conformidad pueden interpretar a 'dotted', 'dashed', 'double', 'groove', 'ridge', 'inset' y 'outset' como 'solid'.

'border-top-style', 'border-right-style', 'border-bottom-style', 'border-left-style'

Valor: <border-style> | inherit

Inicial: none

Se aplica a: todos los elementos

Se hereda: no

Porcentajes: N/A

Medios: visuales

'border-style'

Valor: <border-style>{1,4} | inherit

Inicial: ver las propiedades individuales

Se aplica a: todos los elementos

Se hereda: no

Porcentajes: N/A

Medios: visuales

La propiedad 'border-style' determina el estilo de los cuatro bordes. Puede tener de uno a cuatro valores, y los valores son distribuidos para los distintos lados como en 'border-width' más arriba.

```
#xy34 { border-style: solid dotted }
```

En el ejemplo de arriba, los bordes horizontales serán 'solid' y los bordes verticales serán 'dotted'.

Como el valor inicial del estilo de borde es 'none', ningún borde será visible a menos que se establezca el estilo de borde.

8.5.4 Propiedades resumidas del borde: 'border-top', 'border-bottom', 'border-right', 'border-left' y 'border'

'border-top', 'border-right', 'border-bottom', 'border-left'

Valor: [ <'border-top-width'> || <'border-style'> || <color> ] | inherit

Inicial: ver las propiedades individuales

Se aplica a: todos los elementos

Se hereda: no

Porcentajes: N/A

Medios: visuales

Esta es una propiedad resumida para definir el ancho, estilo y color del borde superior, derecho, inferior e izquierdo de una caja.

```
H1 { border-bottom: thick solid red }
```

La regla de arriba determinará el ancho, el estilo y el color del borde por **debajo** del elemento H1. Los valores omitidos son colocados en sus **valores iniciales**. Como la siguiente regla no especifica un color de borde, el borde tendrá el color especificado en la propiedad 'color':

```
H1 { border-bottom: thick solid }
```

'border'

Valor: [ <'border-width'> || <'border-style'> || <color> ] | inherit

Inicial: ver las propiedades individuales

Se aplica a: todos los elementos

Se hereda: no

Porcentajes: N/A

Medios: visuales

La propiedad 'border' es una propiedad resumida para colocar el mismo ancho, color y estilo a los cuatro bordes de una caja. A diferencia de las propiedades resumidas 'margin' y 'padding', la propiedad 'border' no puede definir diferentes valores para los cuatro bordes. Para eso, deben usarse una a más de las otras propiedades del borde.

Por ejemplo, la primer regla de abajo es equivalente al conjunto de cuatro reglas mostradas a continuación de ella.

```
P { border: solid red }
P {
  border-top: solid red;
  border-right: solid red;
  border-bottom: solid red;
  border-left: solid red
}
```

Como, hasta cierto punto, las propiedades tienen un funcionamiento que se superpone, el orden en que las reglas son especificadas es importante.

Considere este ejemplo:

```
BLOCKQUOTE {
  border-color: red;
  border-left: double;
  color: black
}
```

En el ejemplo de arriba, el color del borde izquierdo es negro, mientras que los otros bordes son rojos. Esto se debe a la definición del ancho, estilo y color para 'border-left'. Como el valor del color no está dado en la propiedad 'border-left', éste será tomado de la propiedad 'color'. El hecho de que la propiedad 'color' esté puesta después que la propiedad 'border-left' no es relevante.

# 9 Modelo de formato visual

## 9.1 Introducción al modelo de formato visual

Este capítulo y el siguiente describen el modelo de formato visual: cómo las aplicaciones del usuario procesan la [estructura del documento](#) para los [medios](#) visuales.

En este modelo de formato visual, cada elemento en la estructura del documento genera cero o más cajas de acuerdo al [modelo de caja](#). La composición de estas cajas se gobierna por:

- [dimensiones de la caja](#) y [tipo](#).
- [esquema de posicionamiento](#) (flujo normal, flotante y absoluto).
- relación entre los elementos de una [estructura del documento](#).
- información externa (ej., tamaño del acceso visual, dimensiones [intrínsecas](#) de las imágenes, etc.).

Las propiedades definidas en este capítulo y el siguiente se aplican a los [medios continuos](#) y a los [medios paginados](#). Sin embargo, el significado de las [propiedades del margen](#) varía cuando son aplicadas a los medios paginados (ver el [modelo de página](#) para más detalles).

El modelo de formato visual no especifica todos los aspectos del formato (ej., no especifica un algoritmo para el espaciado entre letras). Las [aplicaciones del usuario con conformidad](#) pueden tener distintos comportamientos con aquellos formatos de composición no cubiertos por esta especificación.

### 9.1.1 El acceso visual

Las aplicaciones del usuario para los [medios continuos](#) generalmente ofrecen a los usuarios un *acceso visual* (una ventana u otra área de visión en la pantalla) a través del cual los usuarios consultan un documento. Las aplicaciones del usuario pueden cambiar la composición del documento cuando el acceso visual es redimensionado (ver el [bloque de contención inicial](#)). Cuando el acceso visual es más pequeño que el bloque de contención inicial de un documento, la aplicación del usuario debe ofrecer un mecanismo de desplazamiento. A lo sumo hay un acceso visual por [lienzo](#), pero las aplicaciones del usuario pueden procesar más de un lienzo (ej., proporcionar diferentes vistas del mismo documento).

### 9.1.2 Bloques de contención

En CSS2, muchas posiciones y tamaños de cajas son calculados con respecto a los límites de una caja rectangular llamada *bloque de contención*. En general, las cajas generadas actúan como bloques de contención para las cajas descendientes; decimos que una caja "establece" el bloque de contención para sus descendientes. La frase "el bloque de contención de la caja" significa "el bloque de contención en el cuál la caja existe", no aquella que genera.

Cada caja tiene una posición dada con respecto a su bloque de contención, pero no está confinada a este bloque de contención; lo puede [desbordar](#).

La raíz de la [estructura del documento](#) genera una caja que sirve de *bloque de contención inicial* para la composición subsecuente.

El ancho del bloque de contención inicial puede ser especificado con la propiedad 'width' del elemento raíz. Si esta propiedad tiene el valor 'auto', la aplicación del usuario provee el ancho inicial (ej., la aplicación del usuario utiliza el ancho actual del [acceso visual](#)).

La altura del bloque de contención inicial puede ser especificada con la propiedad 'height' del elemento raíz. Si esta propiedad tiene el valor 'auto', la altura del bloque de contención aumentará para acomodar el contenido del documento.

El bloque de contención inicial no puede ser posicionado o flotar (es decir, las aplicaciones del usuario [ignoran](#) las propiedades 'position' y 'float' para el elemento raíz).

Los [detalles](#) del modo en que las dimensiones de un bloque de contención son calculadas se describen en el [capítulo siguiente](#).

## 9.2 Control de la generación de cajas

Las siguientes secciones describen los tipos de cajas que pueden ser generadas en CSS2. Un tipo de caja afecta, en parte, su comportamiento en el modelo de formato visual. La propiedad 'display', descrita abajo, especifica un tipo de caja.

### 9.2.1 Elementos a nivel de bloque y cajas de bloques

Los *elementos a nivel de bloque* son aquellos elementos del documento fuente que son tratados visualmente como bloques (ej., párrafos). Algunos valores de la propiedad 'display' conforman un elemento a nivel de bloque: 'block', 'list-item', 'compact' y 'run-in' (algunas veces; ver [compact boxes](#) y [run-in boxes](#)), y 'tabla'.

Los elementos a nivel de bloque generan una *caja de bloque principal* que sólo contiene *cajas de bloque*. La caja de bloque principal establece el **bloque de contención** para las cajas descendientes y el contenido generado y es también la caja involucrada en cada esquema de posicionamiento. Las cajas de bloque principales participan de un **contexto de formato de bloque**.

Algunos elementos a nivel de bloque generan cajas adicionales fuera de la caja principal: los elementos 'list-item' y aquellos con **marcadores**. Estas cajas adicionales son situadas con respecto a la caja principal.

Cajas de bloque anónimas

En un documento como éste:

```
<DIV>
  Un texto
  <P>Más texto
</DIV>
```

(y asumiendo que DIV y P tienen ambos 'display: block'), DIV parece tener un contenido de línea y un contenido de bloque. Para facilitar el formato, asumimos que existe una *caja de bloque anónima* alrededor de "Un texto".

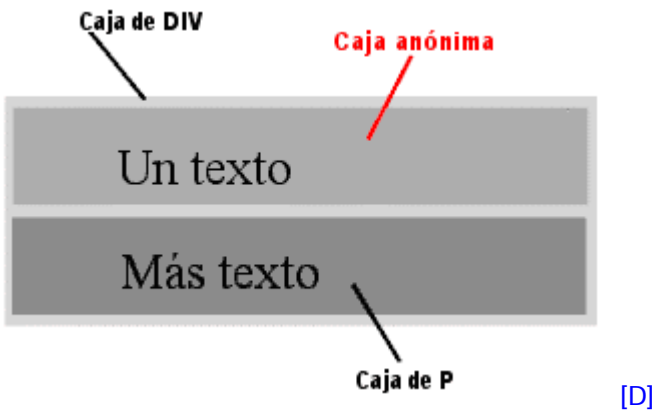


Diagrama mostrando las tres cajas, una de las cuales es anónima, para el ejemplo de arriba.

En otras palabras: si la caja de bloque (como las generadas por DIV, arriba) tiene otra caja de bloque en su interior (como la de P, arriba), entonces la forzamos a contener *sólo* cajas de bloque en su interior, envolviendo a las cajas de línea en una caja de bloque anónima.

Este modelo se aplicaría en el próximo ejemplo si las siguientes reglas:

```
/* Nota: las AU HTML pueden no respetar estas reglas */
BODY { display: inline }
P     { display: block }
```

fueran usadas con este documento HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HEAD>
<TITLE>Texto anónimo interrumpido por un bloque</TITLE>
</HEAD>
<BODY>
  Este es el texto anónimo antes de P.
  <P>Este es el contenido de P.</P>
  Este es el texto anónimo después de P.
</BODY>
```

El elemento BODY contiene un segmento (C1) de texto anónimo seguido por un elemento a nivel de bloque seguido por otro segmento (C2) de texto anónimo. Las cajas resultantes serían una caja de bloque anónima para BODY, conteniendo una caja de bloque anónima alrededor de C1, la caja de bloque para P y otra caja de bloque anónima alrededor de C2.

Las propiedades de las cajas anónimas son heredadas de las cajas no-anónimas que las encierran (en el ejemplo: la de DIV). Las propiedades no heredadas tiene su valor inicial. Por ejemplo, la fuente de la caja anónima es heredada de DIV, pero los márgenes serán igual a 0.

## 9.2.2 Elementos a nivel de línea y cajas de línea

Los *elementos a nivel de línea* son aquellos elementos del documento fuente que no forman nuevos bloques de contenido; el contenido es distribuido en líneas (ej., fragmentos de texto enfatizado dentro de un párrafo, imágenes en línea, etc.). Algunos valores de la propiedad 'display' conforman un elemento de línea: 'inline', 'inline-table', 'compact' y 'run-in' (algunas veces; ver cajas [compact](#) y [run-in](#)). Los elementos a nivel de línea generan cajas de línea.

Las cajas de línea pueden participar de varios contextos de formato:

- Dentro de una caja de bloque, las cajas de línea participan de un [contexto de formato de línea](#).
- Una caja de línea del tipo [compact](#) se da en una posición en el margen de una caja de bloque.
- Las cajas del tipo [marcador](#) también se dan en posiciones fuera de una caja de bloque.

### Cajas de línea anónimas

En un documento como éste:

```
<P>Algún <EM>texto</em> enfatizado</P>
```

P genera una caja de bloque, con algunas cajas de línea dentro de ella. La caja para "texto" en una caja de línea generada por un elemento de línea (EM), pero las otras cajas ("Algún" y "enfatizado") son cajas de línea generadas por un elemento a nivel de bloque (P). Los últimos son denominados cajas de línea anónimas porque no tienen un elemento a nivel de línea asociado.

Tales cajas de línea anónimas heredan las propiedades hereditarias de su caja de bloque padre. Las propiedades no hereditarias tienen su valor inicial. En el ejemplo, el color de las cajas de línea anónimas es heredado de P, pero el fondo es transparente.

Si queda claro en el contexto a qué tipo de caja anónima se refiere, las cajas de línea anónimas y las cajas de bloque anónimas son denominadas simplemente cajas anónimas en esta especificación.

Existen otros tipos de cajas anónimas que surgen en el formato de las [tablas](#).

## 9.2.3 Cajas compact

Una *caja compact* se comporta del siguiente modo:

- Si una caja de [bloque](#) (que no flota y no está [posicionada absolutamente](#)) sigue a la caja compact, la caja compact es procesada como una caja de línea de una línea. El ancho de la caja resultante es comparada con uno de los márgenes laterales de la caja de bloque. La elección del margen izquierdo o derecho es determinada por la 'dirección' especificada para el elemento que produce el [bloque de contención](#) de la caja compact y la caja siguiente. Si el ancho de la caja de línea es menor o igual al margen, la caja de línea se da en una posición en el margen como se describe inmediatamente abajo.
- De lo contrario, la caja compact se convierte en una caja de bloque.

La caja compact se da en una posición en el margen como sigue: está fuera (a la izquierda o a la derecha) de la primer [caja de línea](#) del bloque, pero afecta el cálculo de la [altura de la caja de línea](#). La propiedad 'vertical-align' de la caja compact determina la posición vertical de la caja compact con relación a esa caja de línea. La posición horizontal de la caja compact es siempre en el margen de la caja de bloque.

Un elemento que no puede ser compuesto en una línea no puede ser colocado en el margen del bloque siguiente. Por ejemplo, un elemento 'compact' en HTML que contiene un elemento BR siempre será compuesto como una caja de bloque (asumiendo el estilo predeterminado de BR, que inserta una nueva línea). Para posicionar textos formados por varias líneas en el margen, la propiedad 'float' es a menudo más conveniente.

El siguiente ejemplo ilustra una caja compact.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Un ejemplo de caja compact</TITLE>
    <STYLE type="text/css">
      DT { display: compact }
      DD { margin-left: 4em }
    </STYLE>
  </HEAD>
  <BODY>
    <DL>
      <DT>corta
        <DD><P>La descripción va aquí.
      <DT>demasiado larga para el margen
```

```

        <DD><P>La descripción va aquí.
    </DL>
</BODY>
</HTML>
```

Este ejemplo podría estructurarse como:

```

corta      La descripción va aquí.

demasiado larga para el margen
    La descripción va aquí.
```

La propiedad 'text-align' puede ser usada para alinear los elementos compact dentro del margen: contra el borde izquierdo del margen ('left'), contra el borde derecho ('right') o centrado en el margen ('center'). El valor 'justify' no se aplica y es considerado como 'left' o 'right', dependiendo de la 'dirección' del elemento a nivel de bloque entre cuyos márgenes el elemento compact es procesado. ('left' si la dirección es 'ltr', 'right' si es 'rtl'.)

Por favor consulte la sección sobre [contenido generado](#) por información acerca de cómo las cajas compact interactúan con el contenido generado.

### 9.2.4 Cajas run-in

Una *caja run-in box* se comporta del siguiente modo:

- Si una caja de [bloque](#) (que no flota y no está [posicionada absolutamente](#)) sigue a la caja run-in, la caja run-in se convierte en la primer caja de línea de la caja de bloque.
- De lo contrario, la caja run-in se convierte en una caja de bloque.

Una caja 'run-in' es útil para los encabezados run-in, como en este ejemplo:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Un ejemplo de caja run-in</TITLE>
    <STYLE type="text/css">
      H3 { display: run-in }
    </STYLE>
  </HEAD>
  <BODY>
    <H3>Un encabezado run-in.</H3>
    <P>Y un párrafo de texto que lo sigue.
  </BODY>
</HTML>
```

Este ejemplo podría ser procesado como:

```

Un encabezado run-in. Y un
párrafo de texto que
lo sigue.
```

Las propiedades del elemento run-in son heredados de su padre en la estructura del documento, no de la caja de bloque de la cual visualmente se vuelve parte.

Por favor consulte la sección sobre [contenido generado](#) por información acerca de cómo las cajas run-in interactúan con el contenido generado.

### 9.2.5 La propiedad 'display'

|                  |   |
|------------------|---|
| <b>'display'</b> |   |
| Valor:           | inline   block   list-item   run-in   compact   marker   table   inline-table   table-row-group   table-header-group   table-footer-group   table-row   table-column-group   table-column   table-cell   table-caption   none   inherit |
| Inicial:         | inline  |
| Se aplica a:     | todos los elementos   |
| Se hereda:       | no  |
| Porcentajes:     | N/A   |
| Medios:          | todos   |

Los valores de esta propiedad tienen los siguientes significados:



**block**

Este valor provoca que un elemento genere una caja de bloque principal.

**inline**

Este valor provoca que un elemento genere una o más cajas de línea.

**list-item**

Este valor provoca que un elemento (ej., LI en HTML) genere una caja de bloque principal y una caja de línea list-item. Por información acerca de las listas y ejemplos del formato de listas, por favor consulte la sección sobre [listas](#).

**marker**

Este valor declara que el [contenido generado](#) antes o después de una caja será un marcador. Este valor sólo debe usarse con los [pseudo-elementos :before y :after](#) vinculados a elementos a nivel de bloque. En otros caso, el valor es interpretado como 'inline'. Por favor consulte la sección sobre [marcadores](#) para más información.

**none**

Este valor provoca que un elemento no genere **ninguna** caja en la [estructura del formato](#) (es decir, el elemento no tiene ningún efecto sobre la composición). Los elementos descendientes tampoco generan cajas; este comportamiento **no puede** alterarse poniendo la propiedad 'display' a los descendientes.

Por favor observe que un valor 'none' para display no crea una caja invisible; no crea ninguna caja en absoluto. CSS incluye mecanismos que permiten que un elemento genere cajas en la estructura del formato que afecten la composición pero que ellas mismas sean invisibles. Por favor consulte la sección sobre [visibilidad](#) para más detalles.

**run-in y compact**

Estos valores crean cajas de bloque o de línea, según el contexto. Las propiedades se aplican a las cajas run-in y compact en base a su condición final (a nivel de línea o a nivel de bloque). Por ejemplo, la propiedad 'white-space' sólo se aplica si la caja se convierte en una caja de bloque.

**table, inline-table, table-row-group, table-column, table-column-group, table-header-group, table-footer-group, table-row, table-cell y table-caption**

Estos valores provocan que un elemento se comporte como un elemento tabla (sujeto a las restricciones descritas en el capítulo sobre [tablas](#)).

Observe que aunque el [valor inicial](#) de 'display' es 'inline', las reglas de la [hoja de estilo predeterminada](#) de la aplicación del usuario puede [sustituir](#) este valor. Ver la [hoja de estilo de ejemplo](#) para HTML 4.0 en el apéndice.

Aquí hay algunos ejemplos de la propiedad 'display':

```
P { display: block }
EM { display: inline }
LI { display: list-item }
IMG { display: none } /* No muestra las imágenes */
```

Las aplicaciones del usuario HTML con [conformidad](#) pueden [ignorar](#) la propiedad 'display'.

9.3 Esquemas de posicionamiento

En CSS2, una caja puede situarse según tres *esquemas de posicionamiento*:

- 1. [Flujo normal](#). En CSS2, el flujo normal incluye el [formato de bloque](#) de cajas de [bloque](#), el [fomato de línea](#) de cajas de [línea](#), el [posicionamiento relativo](#) de cajas de bloque o de línea, y el posicionamiento de cajas [compact](#) y [run-in](#).
- 2. [Flotantes](#). En el modelo flotante, un caja se sitúa primero según el flujo normal, luego se saca del flujo y se mueve a la izquierda o a la derecha tanto como sea posible. El contenido puede fluir a lo largo del costado del flotante.
- 3. [Posicionamiento absoluto](#). En el modelo de posicionamiento absoluto, una caja es quitada completamente del flujo normal (no tiene ningún impacto sobre los hermanos siguientes) y se le asigna una posición con respecto al bloque de contención.

*Nota. Los esquemas de posicionamiento de CSS2 ayudan a que los autores hagan sus documento más accesibles permitiéndoles evitar los artilugios (ej., imágenes invisibles) usados para los efectos de composición.*

9.3.1 Elección de un esquema de posicionamiento: la propiedad 'position'

Las propiedades 'position' y 'float' determinan qué algoritmo de posicionamiento CSS2 se usa para calcular la posición de una caja.

**'position'**

|              |  |
|--------------|--|
| Valor:       | static   relative   absolute   fixed   inherit     |
| Inicial:     | static   |
| Se aplica a: | todos los elementos, pero no al contenido generado |
| Se hereda:   | no   |
| Porcentajes: | N/A  |
| Medios:      | visuales   |

Los valores de esta propiedad tiene los siguientes significados:

**static**

La caja es una caja normal, situada de acuerdo al [flujo normal](#). Las propiedades 'top', 'right', 'bottom' y 'left' no se aplican.

**relative**

La posición de la caja se calcula de acuerdo al [flujo normal](#) (ésta se llama la posición en el flujo normal). Luego la caja se desplaza de modo [relativo](#) a su posición normal. Cuando una caja B es posicionada relativamente, la posición de la caja siguiente se calcula como si B no se hubiera desplazado.

**absolute**

La posición de la caja (y posiblemente el tamaño) se especifica con las propiedades 'left', 'right', 'top', y 'bottom'. Estas propiedades especifican los desplazamientos con respecto al [bloque de contención](#) de la caja. Las cajas posicionadas absolutamente son quitadas del flujo normal. Esto significa que ellas no tienen ningún impacto sobre la composición de los hermanos siguientes. También, aunque las cajas [posicionadas absolutamente](#) tiene márgenes, no se [cierran](#) con ningún otro margen.

**fixed**

La posición de la caja se calcula de acuerdo al modelo 'absolute', pero además, la caja se [fija](#) con respecto a alguna referencia. En el caso de los [medios continuos](#), la caja se fija con respecto al [acceso visual](#) (y no se mueve cuando se realiza un desplazamiento). En el caso de los [medios paginados](#), la caja se fija con respecto a la página, aún cuando esa página se vea a través de un [acceso visual](#) (en el caso de la previsualización de un impreso, por ejemplo). Los autores pueden preferir especificar 'fixed' de un modo dependiente de los medios. Por ejemplo, un autor puede desear que una caja permanezca en la parte superior del [acceso visual](#) en la pantalla, pero no al comienzo de cada página impresa. Las dos especificaciones pueden ser separadas utilizando una [regla @media](#), como en:

```
@media screen {
  H1#first { position: fixed }
}
@media print {
  H1#first { position: static }
}
```

9.3.2 Desplazamiento de las cajas: 'top', 'right', 'bottom', 'left'

Se dice que un elemento está *posicionado* si su propiedad 'position' tiene un valor distinto a 'static'. Los elementos posicionados generan cajas posicionadas, situadas de acuerdo a cuatro propiedades:

**'top'**

- Valor: <medida> | <porcentaje> | auto | inherit
- Inicial: auto
- Se aplica a: los elementos posicionados
- Se hereda: no
- Porcentajes: referidos a la altura del bloque de contención
- Medios: visuales

Esta propiedad especifica cuán lejos se desplaza el margen superior del contenido de la caja por debajo del borde superior del [bloque de contención](#) de la caja.

**'right'**

- Valor: <medida> | <porcentaje> | auto | inherit
- Inicial: auto
- Se aplica a: los elementos posicionados
- Se hereda: no
- Porcentajes: referidos al ancho del bloque de contención
- Medios: visuales

Esta propiedad especifica cuán lejos se desplaza el margen derecho del contenido de la caja hacia la izquierda del borde derecho del [bloque de contención](#) de la caja.

**'bottom'**

- Valor: <medida> | <porcentaje> | auto | inherit
- Inicial: auto
- Se aplica a: los elementos posicionados
- Se hereda: no
- Porcentajes: referidos a la altura del bloque de contención
- Medios: visuales

Esta propiedad especifica cuán lejos es desplazado el margen inferior del contenido de la caja por sobre la base del [bloque de contención](#) de la caja.

**'left'**

*Valor:* <medida> | <porcentaje> | auto | inherit  
*Inicial:* auto  
*Se aplica a:* los elementos posicionados  
*Se hereda:* no  
*Porcentajes:* referidos al ancho del bloque de contención  
*Medios:* visuales

Esta propiedad especifica cuán lejos se desplaza el margen izquierdo del contenido de la caja hacia la derecha del borde izquierdo del [bloque de contención](#) de la caja.

Los valores para las cuatro propiedades tienen los siguientes significados:

**<medida>**

El desplazamiento es una distancia fija desde el borde de referencia.

**<porcentaje>**

El desplazamiento es un porcentaje del ancho (para 'left' o 'right') o de la altura (para 'top' y 'bottom') del bloque de contención. Para 'top' y 'bottom', si la altura del bloque de contención no está especificada explícitamente (es decir, depende de la altura del contenido), el valor del porcentaje es interpretado como 'auto'.

**auto**

El efecto de este valor depende de cuáles de las propiedades relacionadas tienen también el valor 'auto'. Ver la sección sobre el [ancho](#) y la [altura](#) de los elementos no reemplazados [posicionados absolutamente](#), para más detalles.

Para las cajas [posicionadas absolutamente](#), el desplazamiento es con respecto al [bloque de contención](#) de la caja. Para las cajas posicionadas relativamente, el desplazamiento es con respecto al borde externo de la propia caja (es decir, a la caja se le da una posición dentro del flujo normal, entonces se desplaza de esa posición de acuerdo a estas propiedades).

## 9.4 Flujo normal

Las cajas dentro del flujo normal pertenecen al contexto del formato, que puede ser de bloque o de línea, pero no ambos simultáneamente. Las cajas de [bloque](#) participan de un contexto de [formato de bloque](#). Las [cajas de línea](#) participan en un contexto de [formato de línea](#).

### 9.4.1 Contexto de formato de bloque

En un contexto de formato de bloque, las cajas se colocan una después de otra, verticalmente, comenzando desde lo alto de un bloque de contención. La distancia vertical entre dos cajas hermanas es determinada por las propiedades 'margin'. Los márgenes verticales entre cajas de bloque adyacentes en un contexto de formato de bloque se [cierran](#).

En un contexto de formato de bloque, cada borde izquierdo externo de la caja toca el borde izquierdo del bloque de contención (para la composición de derecha a izquierda, los bordes derechos se tocan). Esto es así aún en presencia de flotantes (aunque el área del *contenido* de una caja puede encogerse debido a los flotantes)

Para información acerca de los saltos de página en los medios paginados, por favor consulte la sección sobre [saltos de página permitidos](#).

### 9.4.2 Contexto de formato de línea

En un contexto de formato de línea, las cajas son colocadas horizontalmente, una después de otra, comenzando desde lo alto de un bloque de contención. Los márgenes horizontales, bordes y relleno son respetados entre estas cajas. Las cajas puede ser alineadas verticalmente de maneras diferentes: pueden alinearse por su parte superior o inferior, o por las líneas de base del texto en su interior. El área rectangular que contiene las cajas que forman una línea es llamada una *caja de línea*.

El ancho de una caja de línea es determinado por un [bloque de contención](#). La altura de una caja de línea es determinada por las reglas establecidas en la sección sobre [cálculo de la altura de la línea](#). Una caja de línea es siempre suficientemente alta para todas las cajas que contiene. Sin embargo, puede ser mayor que la caja más alta que contiene (si, por ejemplo, las cajas son alineadas de manera que sus líneas de base queden en línea). Cuando la altura de una caja B es menor que la altura de la caja de línea que la contiene, la alineación vertical de B dentro de la caja de línea es determinada por la propiedad 'vertical-align'.

Cuando varias cajas de línea no pueden ajustarse horizontalmente dentro de una sola caja de línea, son distribuidas entre dos o más cajas de línea apiladas verticalmente. De este modo, un párrafo es una pila vertical de cajas de líneas. Las cajas de línea son apiladas sin ninguna separación vertical y nunca se superponen.

En general, el borde izquierdo de una caja de línea toca el borde izquierdo de su bloque de contención y el borde derecho toca el borde derecho de su bloque de contención. Sin embargo, las cajas flotantes pueden interponerse entre el borde del bloque de contención y el borde de la caja de línea. Así, aunque las cajas de línea en el mismo contexto de formato de línea generalmente tienen igual ancho (el del bloque de contención), pueden variar en su ancho si el espacio horizontal disponible es reducido por los [flotantes](#). Las cajas de línea en el mismo contexto de formato de línea generalmente varían en altura (ej., una línea podría contener una imagen muy alta mientras que las otras contienen sólo texto).

Cuando el ancho total de las cajas de línea en una línea es menor que el ancho de la caja de línea que las contiene, su distribución horizontal dentro de la caja de línea es determinada por la propiedad 'text-align'. Si esa propiedad tiene el valor 'justify', la aplicación del usuario puede estirar también las cajas de línea.

Puesto que una caja de línea no puede exceder el ancho de una caja de línea, las cajas de línea largas son divididas en varias cajas y éstas distribuidas en varias cajas de línea. Cuando una caja de línea es dividida, los márgenes, bordes y relleno no tienen ningún efecto visual donde se produce la división. El formato de los márgenes, bordes y relleno puede no ser completamente definido si la división se produce dentro de una incrustación bidireccional.

Las cajas de línea pueden también dividirse en varias cajas *dentro de la misma caja de línea* debido al [procesamiento del texto bidireccional](#).

Aquí hay un ejemplo de construcción de una caja de línea. El siguiente párrafo (creado por el elemento HTML a nivel de bloque P) contiene texto anónimo mezclado con los elementos EM y STRONG:

```
<P>Algunas <EM>palabras con énfasis</EM> aparecen
<STRONG>en esta</STRONG> oración, querida.</P>
```

El elemento P genera una caja de bloque que contiene cinco cajas de línea, tres de las cuáles son anónimas:

- Anónimo: "Algunas"
- EM: "palabras con énfasis"
- Anónimo: "aparecen"
- STRONG: "en esta"
- Anónimo: "oración, querida."

Para dar formato al párrafo, la aplicación del usuario fluye las cinco cajas dentro de cajas de línea. En este ejemplo, la caja generada por el elemento P establece el bloque de contención para las cajas de línea. Si el bloque de contención es suficientemente ancho, todas las cajas de línea encajarán en una sola caja de línea:

```
Algunas palabras con énfasis aparecen en esta oración, querida.
```

Si no, las cajas de línea serán divididas y distribuidas en varias cajas de línea. El párrafo anterior podría dividirse como sigue:

```
Algunas palabras con énfasis aparecen
en esta oración, querida.
```

o así:

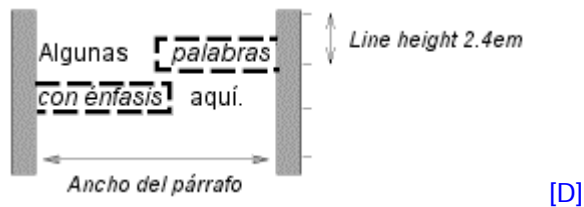
```
Algunas palabras
con énfasis aparecen
en esta oración,
querida.
```

En el ejemplo anterior, la caja EM se dividió en dos cajas EM (llamémoslas "split1" y "split2"). Los márgenes, bordes y relleno, o las decoraciones del texto no tiene efecto visible después de split1 o antes de split2.

Considere el ejemplo siguiente:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Ejemplo de flujo en línea sobre varias líneas</TITLE>
    <STYLE type="text/css">
      EM {
        padding: 2px;
        margin: 1em;
        border-width: medium;
        border-style: dashed;
        line-height: 2.4em;
      }
    </STYLE>
  </HEAD>
  <BODY>
    <P>Algunas <EM>palabras con énfasis</EM> aquí.</P>
  </BODY>
</HTML>
```

Dependiendo del ancho de P, las cajas pueden distribuirse como sigue:



[D]

- El margen se inserta antes de "palabras" y después de "énfasis".
- El relleno se inserta antes, sobre y debajo de "palabras", y después, sobre y debajo de "con énfasis". Un borde con líneas cortadas es procesado en los tres lados en cada caso.

### 9.4.3 Posicionamiento relativo

Una vez que la caja es situada de acuerdo al [flujo normal](#), puede moverse con relación a su posición. A esto se le llama *posicionamiento relativo*. Desplazando una caja (B1) de esta manera no conlleva ningún efecto sobre la caja (B2) que le sigue: a B2 se le da una posición como si B1 no hubiera sido desplazada y B2 no es reposicionada después de que el desplazamiento de B1 ha sido aplicado. Esto implica que el posicionamiento relativo puede ocasionar que las cajas se superpongan.

Las cajas posicionadas relativamente mantienen su tamaño de flujo normal, incluyendo los saltos de línea y el espacio originalmente reservado para ellos. Una caja posicionada relativamente establece un nuevo [bloque de contención](#) para los hijos con flujo normal y los descendientes posicionados.

Una caja posicionada relativamente es generada cuando la propiedad 'position' de un elemento contiene el valor 'relative'. El desplazamiento es especificado con las propiedades 'top', 'bottom', 'left' y 'right'.

El movimiento dinámico de cajas posicionadas relativamente puede producir efectos de animación en entornos de scripts (ver también la propiedad 'visibility'). El posicionamiento relativo también puede usarse como una forma general de producir subíndices y superíndices excepto que la altura de la línea no es automáticamente ajustada para tomar en consideración el posicionamiento. Ver la descripción del [cálculo de la altura de la línea](#) para más información.

Ejemplos de posicionamiento relativo se proporcionan en la sección sobre [comparación entre flujo normal, flotantes y posicionamiento absoluto](#).

## 9.5 Flotantes

Un flotante es una caja que es desplazada a la izquierda o a la derecha en la línea actual. La característica más interesante de un flotante (o "flotador" o caja "flotante") es que el contenido puede fluir por su costado (o estar impedido de hacerlo mediante la propiedad 'clear'). El contenido fluye por el costado derecho de una caja flotante a la izquierda y por el lado izquierdo de una caja flotante a la derecha. La siguiente es una introducción al posicionamiento flotante y al flujo del contenido; las [reglas](#) precisas que gobiernan el comportamiento flotante se dan en la descripción de la propiedad 'float'.

Una caja flotante debe tener un ancho explícito (asignado a través de la propiedad 'width', o un ancho [intrínseco](#) en el caso de los [elementos reemplazados](#)). Cualquier caja flotante se convierte en una [caja de bloque](#) que se desplaza hacia la izquierda o derecha hasta que su borde externo toca el borde del bloque de contención o el borde externo de otro flotante. La parte superior de la caja flotante se alinea con la parte superior de la caja de línea actual (o la parte inferior de la caja de bloque precedente si no existe ninguna caja de línea). Si no hay suficiente espacio horizontal en la línea actual para el flotante, es desplazado hacia abajo, línea por línea, hasta una línea que tenga espacio para él.

Puesto que un flotante no está en el flujo, las cajas de bloque no posicionadas creadas antes y después de la caja flotante fluyen verticalmente como si el flotante no existiera. Sin embargo, las cajas de línea creadas al lado del flotante son acortadas para darle espacio a la caja flotante. Cualquier contenido en la línea actual antes de una caja flotante se vuelve a hacer fluir en la primer línea disponible al otro lado del flotante.

Varios flotantes pueden ser adyacentes, y este modelo también se aplica a flotantes adyacentes en la misma línea.

La siguiente regla hace flotar todas las cajas IMG con `class="icon"` a la izquierda (y pone el margen izquierdo en '0'):

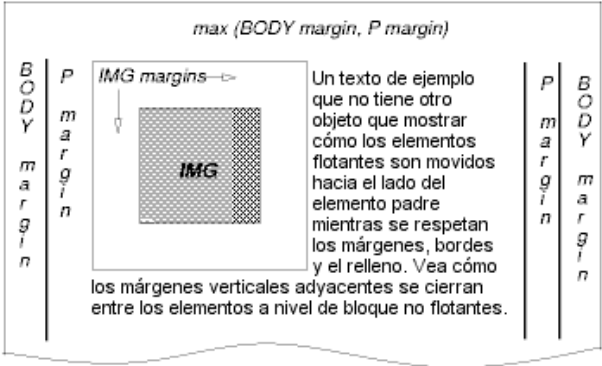
```
IMG.icon {
  float: left;
  margin-left: 0;
}
```

Considere la siguiente fuente HTML y hoja de estilo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
```

```
<HTML>
  <HEAD>
    <TITLE>Ejemplo de flotante</TITLE>
    <STYLE type="text/css">
      IMG { float: left }
      BODY, P, IMG { margin: 2em }
    </STYLE>
  </HEAD>
  <BODY>
    <P><IMG src=img.gif alt="Esta imagen ilustrará un flotante">
      Un texto de ejemplo que no tiene otro...
    </P>
  </BODY>
</HTML>
```

La caja IMG flota hacia la izquierda. El contenido que le sigue es compuesto a la derecha del flotante, comenzando en la misma línea que el flotante. Las cajas de línea a la derecha del flotante son acortadas debido a la presencia del flotante, pero retoman su ancho "normal" (el del bloque de contención establecido por el elemento P) después del flotante. Este documento podría componerse como:



[D]

El formato habría sido exactamente el mismo si el documento hubiera sido:

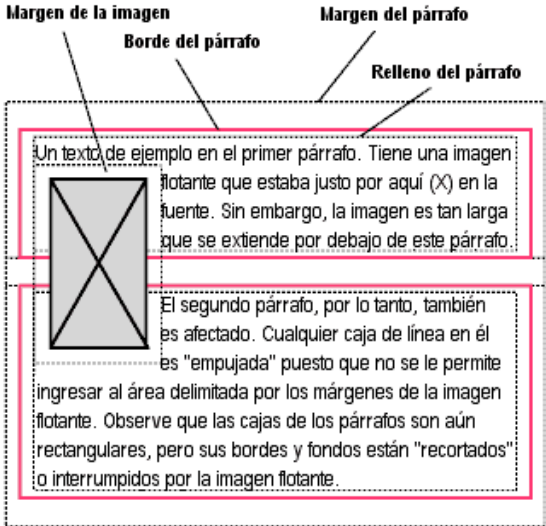
```
<BODY>
  <P>Un texto de ejemplo
  <IMG src=img.gif alt="Esta imagen ilustrará un flotante">
    que no tiene otro...
</BODY>
```

debido a que el contenido a la izquierda del flotante es desplazado por el flotante y vuelto a fluir por su lado derecho.

Los márgenes de las cajas flotantes nunca se **cierran** con los márgenes de las cajas adyacentes. De este modo, en el ejemplo anterior, los márgenes verticales no se **cierran** entre la caja de P y la caja flotante de IMG.

Un flotante puede superponerse con otras cajas en el flujo normal (ej., cuando una caja de flujo normal al lado de un flotante tiene márgenes negativos). Cuando una caja de línea se superpone con un flotante, el contenido, fondo y bordes de la caja de línea son procesados delante del flotante. Cuando una caja de bloque se superpone, el fondo y los bordes de la caja de bloque son procesados detrás del flotante y son solamente visibles cuando la caja es transparente. El contenido de la caja de bloque es procesado delante del flotante.

Aquí hay otra ilustración que muestra lo que sucede cuando un flotante se superpone con los bordes de elementos en el flujo normal.



[D]



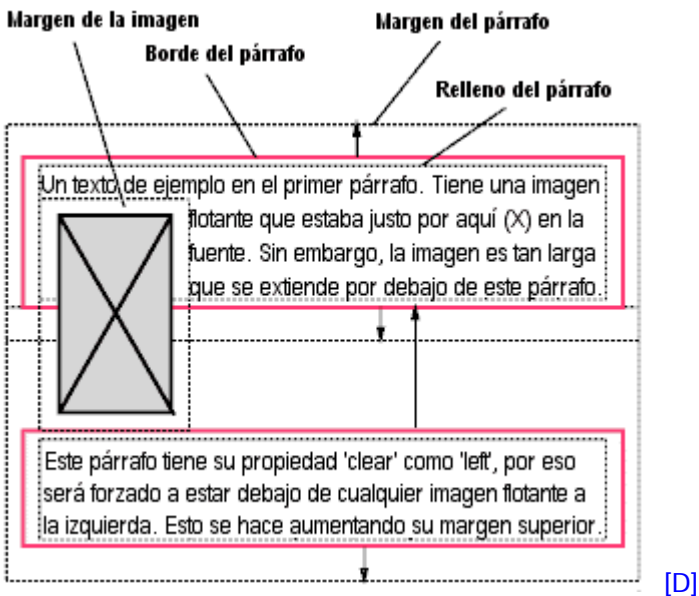
Una imagen flotante oculta los bordes de las cajas de bloque con los que se superpone.

El ejemplo siguiente ilustra el uso de la propiedad 'clear' para impedir que el contenido fluya al costado de un flotante.

Asumiendo una regla como ésta:

```
P { clear: left }
```

el formato podría parecerse a esto:



Ambos párrafos tienen especificado 'clear: left', lo que provoca que el segundo párrafo sea "empujado" a una posición debajo del flotante --su margen superior se expande para conseguir esto (ver la propiedad 'clear')--.

9.5.1 Posicionamiento del flotante: la propiedad 'float'

'float'

- Valor: left | right | none | inherit
- Inicial: none
- Se aplica a: todos menos los elementos posicionados y el contenido generado
- Se hereda: no
- Porcentajes: N/A
- Medios: visuales

Esta propiedad especifica si una caja debe flotar a la izquierda, derecha o no debe flotar en absoluto. Puede especificarse para los elementos que generan cajas que no están posicionadas absolutamente. Los valores de esta propiedad tiene los siguientes significados:

left

El elemento genera una caja de bloque que flota a la izquierda. El contenido fluye sobre el costado derecho de la caja, comenzando en la parte superior (sujeto a la propiedad 'clear'). La propiedad 'display' es ignorada, a menos que tenga el valor 'none'.

right

Igual que 'left', pero el contenido fluye sobre el costado izquierdo de la caja, comenzando en la parte superior.

none

La caja no es flotante.

Aquí están las reglas precisas que gobiernan el comportamiento de los flotantes:

- El borde exterior izquierdo de una caja flotante a la izquierda no puede estar a la izquierda del borde izquierdo de su bloque de contención. Una regla análoga se establece para los elementos flotantes a la derecha.



- 2. Si la caja actual es flotante a la izquierda y hay cualquier otra caja flotante a la izquierda generada por elementos que aparecen antes en el documento fuente, entonces para cada una de tales cajas anteriores, o el **borde exterior** izquierdo de la caja actual debe estar a la derecha del **borde exterior** derecho de la caja anterior, o su parte superior deber estar más abajo que la parte inferior de la caja anterior. Reglas análogas se establecen para las cajas flotantes a la derecha.
- 3. El **borde exterior** derecho de una caja flotante a la izquierda no puede estar a la derecha del **borde exterior** izquierdo de ninguna caja flotante a la derecha que se encuentra a su derecha. Reglas análogas se establecen para los elementos flotantes a la derecha.
- 4. El **borde exterior** superior de una caja flotante no puede estar a mayor altura que la parte superior de su **bloque de contención**.
- 5. El **borde exterior** superior de una caja flotante no puede estar a mayor altura que el borde exterior superior de ninguna caja de **bloque** o **flotante** generada por un elemento que aparece antes en el documento fuente.
- 6. El **borde exterior** superior de la caja flotante de un elemento no puede estar a mayor altura que la parte superior de ninguna **caja de línea** que contiene una caja generada por un elemento que aparece antes en el documento fuente.
- 7. Una caja flotante a la izquierda que tiene otra caja flotante a la izquierda a su izquierda no puede tener su borde exterior derecho a la derecha de borde derecho de su bloque de contención. (En síntesis: un flotante izquierdo no puede estar pegado al borde derecho a menos que ya se encuentre tan a la izquierda como le sea posible). Una regla análoga se sostiene para los elementos flotantes a la derecha.
- 8. Una caja flotante debe colocarse tan alta como sea posible.
- 9. Una caja flotante a la izquierda debe ponerse tan a la izquierda como sea posible, una caja flotante a la derecha tan a la derecha como sea posible. Una posición más alta es preferible a una que se encuentre más a la izquierda/derecha.

9.5.2 Control del flujo al costado del flotante: la propiedad 'clear'

'clear'

Valor: none | left | right | both | inherit  
Inicial: none  
Se aplica a: los elementos a nivel de bloque  
Se hereda: no  
Porcentajes: N/A  
Medios: visuales

Esta propiedad indica cuál de los lados de la(s) caja(s) de un elemento *no* puede quedar adyacente a una caja flotante anterior. (Puede suceder que el propio elemento tenga descendientes flotantes; la propiedad 'clear' no tiene ningún efecto sobre estos.)

Esta propiedad sólo puede especificarse para elementos a **nivel de bloque** (incluyendo los flotantes). Para las cajas **compact** y **run-in**, esta propiedad se aplica a la caja de bloque final al que la caja compact o run-in pertenece.

Los valores tienen los siguientes significados cuando se aplican a cajas de bloque no flotantes:

- left**  
El margen superior de la caja generada se aumenta lo suficiente para que el borde exterior superior quede debajo del borde exterior inferior de cualquier caja flotante a la izquierda que sea el resultado de elementos que aparecen antes en el documento fuente.
- right**  
El margen superior de la caja generada se aumenta lo suficiente para que el borde exterior superior quede debajo del borde exterior inferior de cualquier caja flotante a la derecha que sea el resultado de elementos que aparecen antes en el documento fuente.
- both**  
La caja generada se mueve debajo de todas las cajas flotantes que aparecen antes en el documento fuente.
- none**  
Ninguna restricción a la posición de la caja con respecto a los flotantes.

Cuando la propiedad se determina para elementos flotantes, se produce una modificación en las **reglas** para el posicionamiento del flotante. Una restricción adicional (#10) es agregada:

- El **borde exterior** superior del flotante debe quedar debajo del borde exterior inferior de todas las cajas anteriores flotantes a la izquierda (en el caso de 'clear: left'), o de todas las cajas anteriores flotantes a la derecha (en el caso de 'clear: right'), o ambas ('clear: both').

9.6 Posicionamiento absoluto

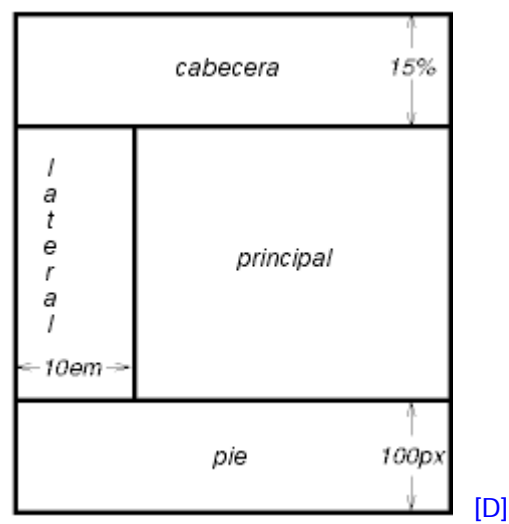
En el modelo de posicionamiento absoluto, una caja es explícitamente desplazada con respecto a su bloque de contención. Es quitada del flujo normal completamente (no tiene ningún impacto sobre los hermanos siguientes). Una caja absolutamente posicionada establece un nuevo bloque de contención para los hijos en el flujo normal y los descendientes posicionados. Sin embargo, el contenido de un elemento absolutamente posicionado no fluye alrededor de ninguna otra caja. Ellos pueden o no tapar el contenido de otra caja, dependiendo del **nivel de pila** de las cajas solapadas.

Las referencias en esta especificación a un *elemento absolutamente posicionado* (o su caja) implica que la propiedad 'position' del elemento tiene el valor 'absolute' o 'fixed'.

### 9.6.1 Posicionamiento fijo

El posicionamiento fijo es una subcategoría del posicionamiento absoluto. La única diferencia es que para una caja posicionada de modo fijo, el bloque de contención es establecido por el [acceso visual](#). Para los [medios contínuos](#), las cajas fijas no se mueven cuando el documento es desplazado. En este aspecto, son similares a las [imágenes fijas de fondo](#). Para los [medios paginados](#), las cajas con posiciones fijas son repetidas en cada página. Esto resulta útil para poner, por ejemplo, una firma al pie de cada página.

Los autores pueden usar el posicionamiento fijo para crear presentaciones al estilo de los marcos. Considere la siguiente composición con marcos:



Esto podría lograrse con el siguiente documento HTML y sus reglas de estilo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Un documento con marcos con CSS2</TITLE>
    <STYLE type="text/css">
      BODY { height: 8.5in } /* Requerido para los porcentajes de altura
usados abajo */
      #cabecera {
        position: fixed;
        width: 100%;
        height: 15%;
        top: 0;
        right: 0;
        bottom: auto;
        left: 0;
      }
      #lateral {
        position: fixed;
        width: 10em;
        height: auto;
        top: 15%;
        right: auto;
        bottom: 100px;
        left: 0;
      }
      #principal {
        position: fixed;
        width: auto;
        height: auto;
        top: 15%;
        right: 0;
        bottom: 100px;
        left: 10em;
      }
      #pie {
        position: fixed;
        width: 100%;
        height: 100px;
        top: auto;
        right: 0;
        bottom: 0;
      }
    </STYLE>
  </HEAD>
  <BODY>
    <div id="cabecera">
      <div id="lateral">
        <div id="principal">
          <div id="pie">
            </div>
          </div>
        </div>
      </div>
    </div>
  </BODY>
</HTML>
```

```
        left: 0;
    }
</STYLE>
</HEAD>
<BODY>
    <DIV id="cabecera"> ... </DIV>
    <DIV id="lateral"> ... </DIV>
    <DIV id="principal"> ... </DIV>
    <DIV id="pie"> ... </DIV>
</BODY>
</HTML>
```

## 9.7 Relaciones entre 'display', 'position' y 'float'

Las tres propiedades que afectan a la generación de cajas y a la composición --'display', 'position' y 'float'-- interactúan del siguiente modo:

1. Si 'display' tiene el valor 'none', las aplicaciones del usuario deben **ignorar** 'position' y 'float'. En este caso, el elemento no genera ninguna caja.
2. De otro modo, si 'position' tiene como valor 'absolute' o 'fixed', 'display' es definido como 'block' y 'float' como 'none'. La posición de la caja será determinada por las propiedades 'top', 'right', 'bottom' y 'left' y el bloque de contención de la caja.
3. De otro modo, si 'float' tiene un valor distinto a 'none', 'display' es definida como 'block' y la caja es flotante.
4. De otro modo, las restantes propiedades 'display' se aplican como están especificadas.

***Nota.** CSS2 no especifica un comportamiento para la composición cuando los valores para estas propiedades son cambiados mediante un lenguaje de scripts. Por ejemplo, ¿qué sucede cuando un elemento que tiene 'width: auto' es reposicionado? ¿El contenido vuelve a volcarse en el flujo o mantiene su formato original? La respuesta está fuera del alcance de este documento y tal comportamiento probablemente difiera en las implementaciones iniciales de CSS2.*

## 9.8 Comparación entre flujo normal, flotantes y posicionamiento absoluto

Para ilustrar la diferencia entre flujo normal, posicionamiento relativo, flotantes y posicionamiento absoluto, proporcionamos una serie de ejemplos basados en el siguiente fragmento HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Comparación entre esquemas de posicionamiento</TITLE>
  </HEAD>
  <BODY>
    <P>El comienzo del contenido.
      <SPAN id="externo"> Este es el tema externo.
      <SPAN id="interno"> Tema interno.</SPAN>
      Fin del tema interno.</SPAN>
      El final del contenido.
    </P>
  </BODY>
</HTML>
```

En este documento, asumimos las siguientes reglas:

```
BODY { display: block; line-height: 200%;
        width: 400px; height: 400px }
P     { display: block }
SPAN  { display: inline }
```

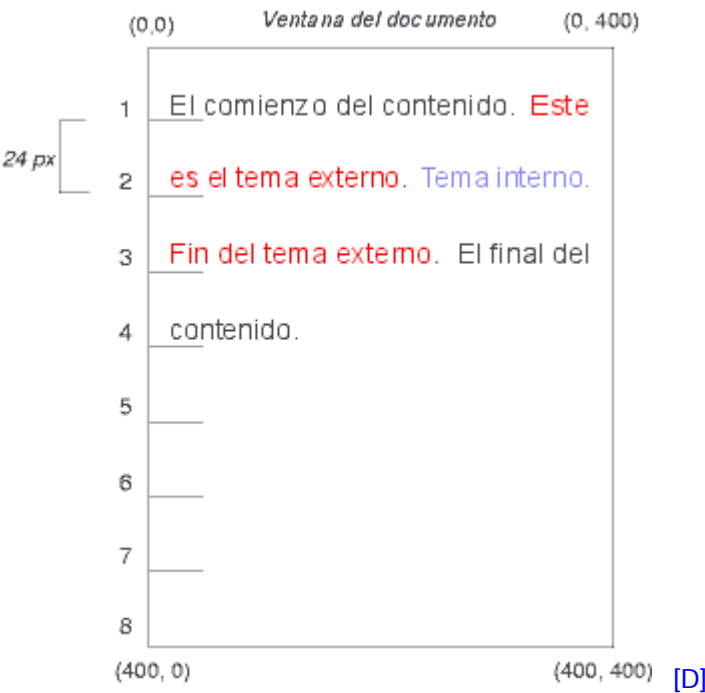
La posición final de las cajas generadas por los elementos *externo* e *interno* varía en cada ejemplo. En cada ilustración, los números a la izquierda de la ilustración indican la posición en el **flujo normal** de las líneas con doble espacio (para mayor claridad). (Nota: las ilustraciones usan diferentes escalas horizontales y verticales.)

### 9.8.1 Flujo normal

Considere las siguientes declaraciones CSS para *externo* e *interno* que no alteran el **flujo normal** de las cajas:

```
#externo { color: red }
#interno { color: blue }
```

El elemento P posee todo su contenido a nivel de línea: [texto de línea anónimo](#) y dos elementos SPAN. Por lo tanto, todo el contenido será colocado en un contexto de formato de línea, con un bloque de contención establecido por el elemento P, produciendo algo similar a:



### 9.8.2 Posicionamiento relativo

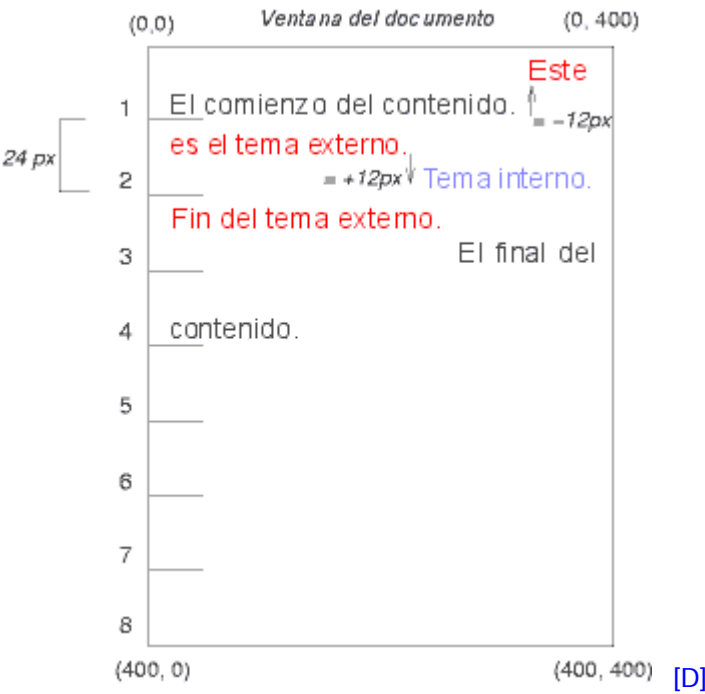
Para ver el efecto del [posicionamiento relativo](#), especificamos:

```
#externo { position: relative; top: -12px; color: red }
#interno { position: relative; top: 12px; color: blue }
```

El texto fluye normalmente hasta el elemento *externo*. El texto *externo* fluye entonces dentro de la posición y dimensiones de su flujo normal al final de la línea 1. Luego, la cajas de línea que contienen el texto (distribuido en tres líneas) son movidas como una unidad en '-12px' (hacia arriba).

El contenido de *interno*, como hijo de *externo*, debería fluir inmediatamente después de las palabras "es el tema externo" (en la línea 1.5). Sin embargo, el propio contenido *interno* está desplazado con respecto al contenido *externo* en '12px' (hacia abajo), regresando a su posición original en la línea 2.

Observe que el contenido que sigue a *externo* no es afectado por el posicionamiento relativo de *externo*.



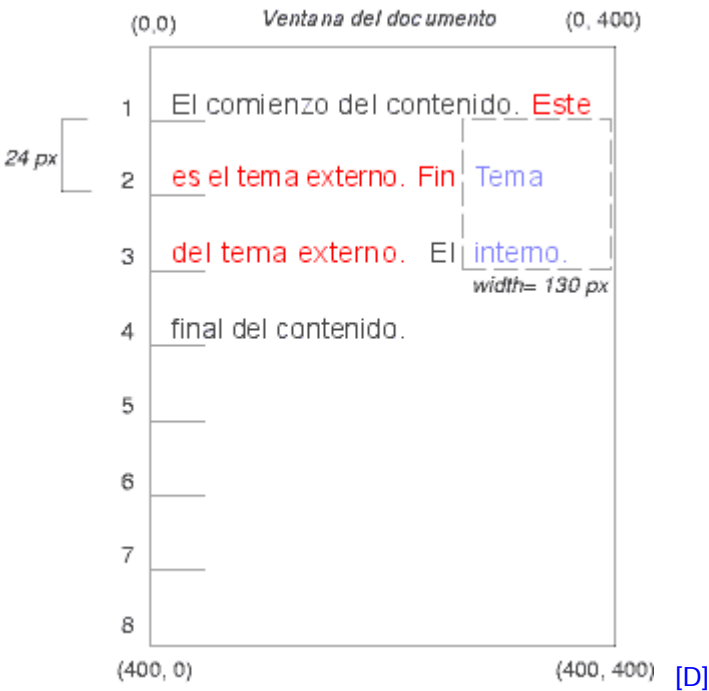
Observe que si el desplazamiento de *externo* hubiera sido de '-24px', el texto de *externo* y el texto del contenido se habrían encimado.

### 9.8.3 Flotar una caja

Ahora considere el efecto de **flotar** el texto del elemento *interno* a la derecha por medio de las siguientes reglas:

```
#outer { color: red }
#inner { float: right; width: 130px; color: blue }
```

El texto fluye normalmente hasta la caja de *interno*, la que es empujada fuera del flujo y flotada sobre el margen derecho (su propiedad 'width' ha sido asignada explícitamente). Las cajas de línea sobre la izquierda son acortadas y el texto del documento restante fluye dentro de ellas.



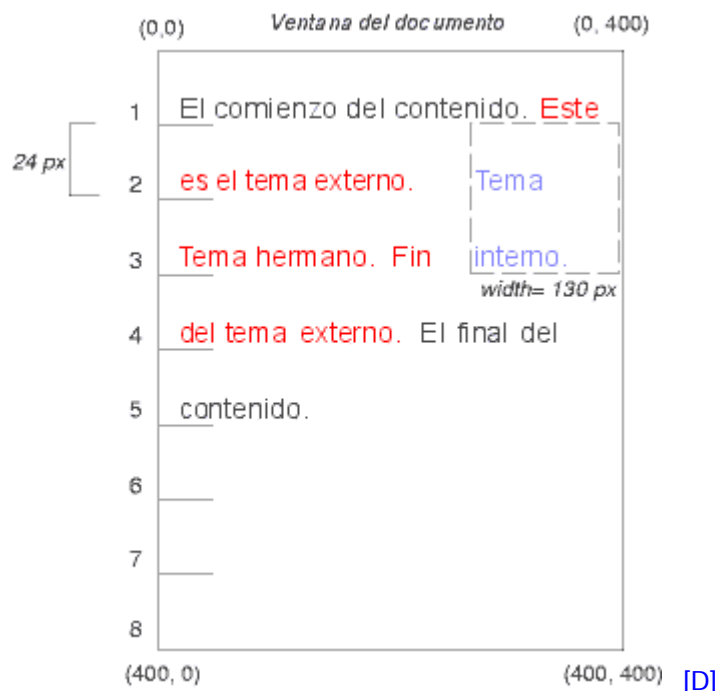
Para mostrar el efecto de la propiedad 'clear' agregamos un elemento *hermano* al ejemplo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Comparación entre esquemas de posicionamiento</TITLE>
  </HEAD>
  <BODY>
    <P>El comienzo del contenido.
      <SPAN id=externo> Este es el tema externo.
      <SPAN id=interno> Tema interno.</SPAN>
      <SPAN id=hermano> Tema hermano.</SPAN>
      Fin del tema interno.</SPAN>
      El final del contenido.
    </P>
  </BODY>
</HTML>
```

Las siguientes reglas:

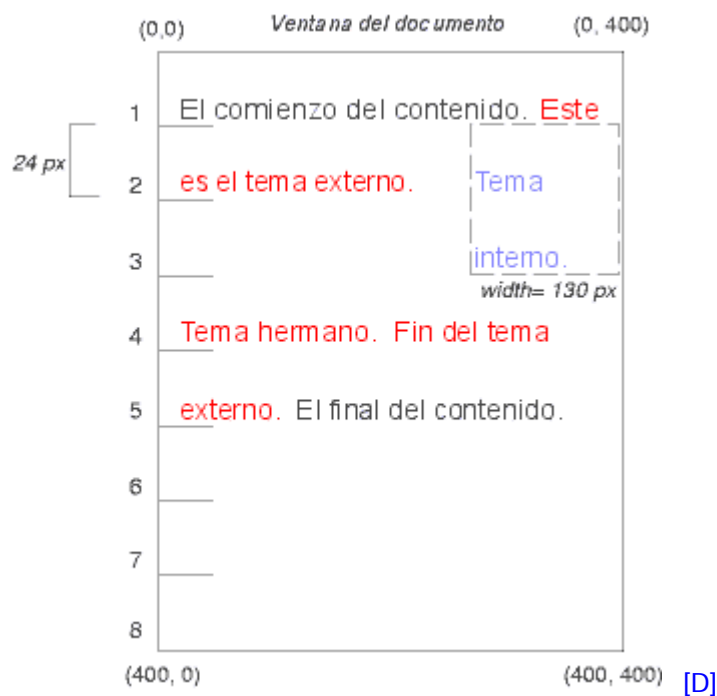
```
#interno { float: right; width: 130px; color: blue }
#hermano { color: red }
```

provocan que la caja de *interno* flote a la derecha como antes y el texto del documento restante fluye dentro del espacio vacante:



Sin embargo, si la propiedad 'clear' en el elemento *hermano* es puesta como 'right' (es decir, la caja *hermana* generada no aceptará una posición al lado de las cajas flotantes a su derecha), el contenido de *hermano* comienza a fluir debajo del flotante:

```
#interno { float: right; width: 130px; color: blue }
#hermano { clear: right; color: red }
```

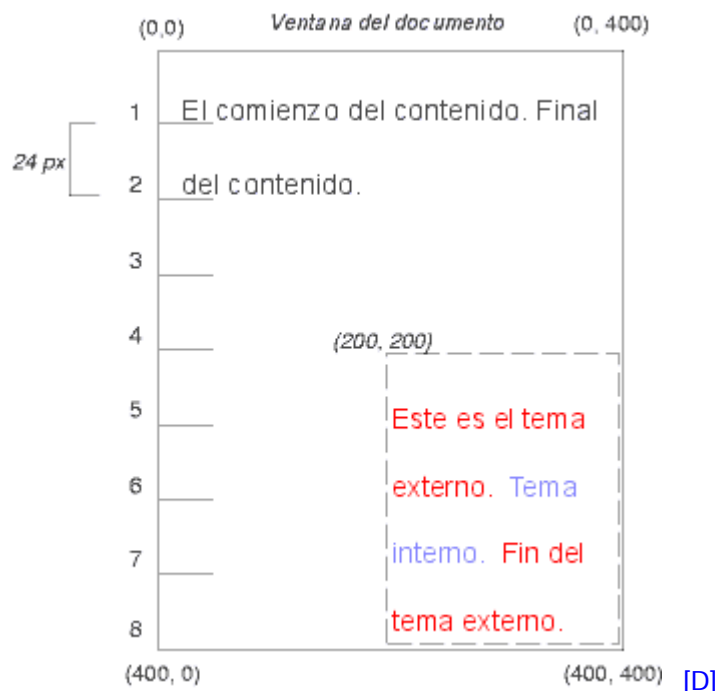


### 9.8.4 Posicionamiento absoluto

Finalmente, consideramos el efecto del **posicionamiento absoluto**. Considere las siguientes declaraciones CSS para *externo* e *interno*:

```
#esterno {
    position: absolute;
    top: 200px; left: 200px;
    width: 200px;
    color: red;
}
#interno { color: blue }
```

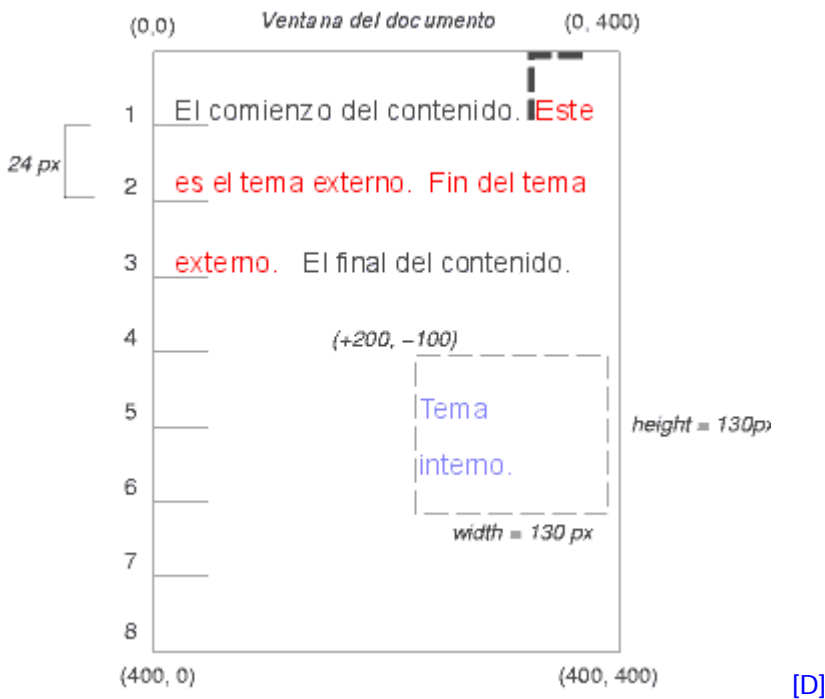
que provocan que el lado superior de la caja de *externo* sea posicionada con respecto a su bloque de contención. El bloque de contención para una caja posicionada se establece por el antepasado posicionado más cercano (o, si no existe ninguno, el [bloque de contención inicial](#), como en nuestro ejemplo). El lado superior de la caja de *externo* está '200px' debajo de la parte superior del bloque de contención y el lado izquierdo está a '200px' del lado izquierdo. La caja hija de *externo* fluye normalmente con respecto a su padre.



El siguiente ejemplo muestra una caja posicionada absolutamente que es hija de una caja relativamente posicionada. Aunque la caja padre *externo* no está realmente desplazada, poniendo su propiedad 'position' como 'relative' significa que su caja puede servir como bloque de contención para los descendientes posicionados. Como la caja de *externo* es una caja de línea que está dividida entre varias líneas, los bordes superior e izquierdo de la primer caja de línea (señalados con líneas cortadas gruesas en la ilustración de abajo) sirven como referencia para los desplazamientos de 'top' y 'left'.

```
#externo {
  position: relative;
  color: red
}
#interno {
  position: absolute;
  top: 200px; left: -100px;
  height: 130px; width: 130px;
  color: blue;
}
```

Esto da como resultado algo similar a lo siguiente:



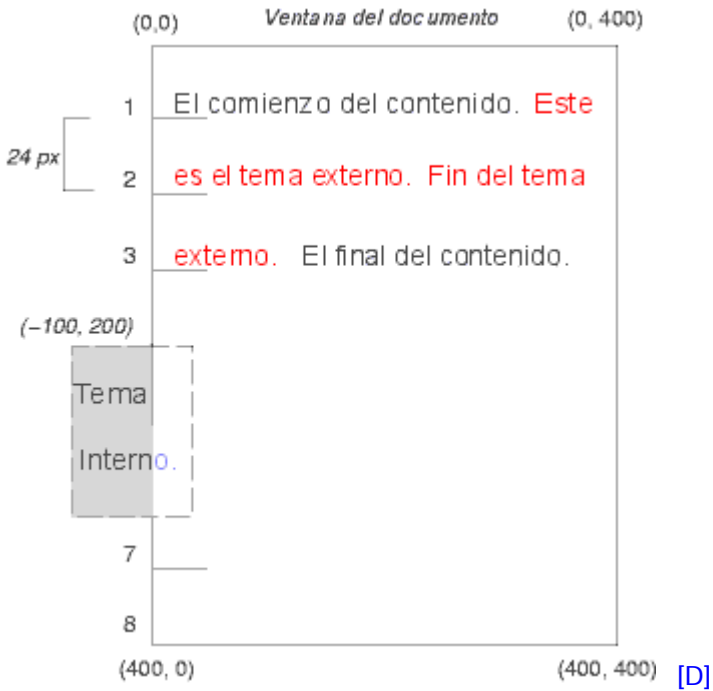
Si no posicionamos la caja de *externo*:

```
#externo { color: red }
#interno {
  position: absolute;
```



```
top: 200px; left: -100px;
height: 130px; width: 130px;
color: blue;
}
```

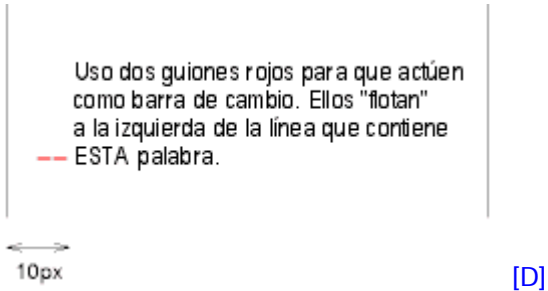
el bloque de contención para *interno* es ahora el **bloque de contención inicial** (en nuestro ejemplo). La siguiente ilustración muestra donde acabaría la caja de *interno* en este caso.



El posicionamiento relativo y absoluto puede utilizarse para implementar barras de cambio, como se muestra en el siguiente ejemplo. El documento que sigue:

```
<P style="position: relative; margin-right: 10px; left: 10px;">
Uso dos guiones rojos para que actúen como barra de cambio.
Ellos "flotan" a la izquierda de la línea que contiene ESTA
<SPAN style="position: absolute; top: auto; left: -1em; color: red;">--
</SPAN>
palabra.</P>
```

resultaría en algo como:



Primero, el párrafo (cuyos lados del bloque de contención son mostrados en la ilustración) fluye normalmente. Entonces es desplazado '10px' desde el borde izquierdo de su bloque de contención (así también, un margen derecho de '10px' ha sido reservado en anticipación al desplazamiento). Los dos guiones que actúan como barras de cambio son sacadas del flujo y posicionadas en la línea actual (debido a 'top: auto'), '-1em' desde el margen izquierdo de su bloque de contención (establecido por P en su posición final). El resultado es que la barra de cambio parece "flotar" a la izquierda de la línea actual.

## 9.9 Presentación por capas

En la siguiente sección, la expresión "en frente de" significa más cercano al usuario considerando como el usuario enfrente a la pantalla.

En CSS2, cada caja tiene una posición en tres dimensiones. Además de su posición horizontal y vertical, las cajas se ubican a lo largo de un eje "z" y son procesadas una encima de otra. Las posiciones en el eje Z son particularmente relevantes cuando las cajas se superponen visualmente. Esta sección trata de cómo las cajas pueden ser posicionadas a lo largo del eje z.

Cada caja pertenece a un *contexto de pila*. Cada caja en un contexto de pila dado tiene un entero como *nivel de pila*, el cuál es su posición en el eje z en relación a otras cajas en el mismo contexto de pila. Las cajas con niveles de pila mayores son siempre procesados en frente de las cajas con niveles de pila menores. Las cajas pueden tener niveles de pila negativos. Las cajas con el mismo nivel de pila en un contexto de pila son apiladas de abajo hacia arriba de acuerdo al orden en la estructura del documento.

El elemento **raíz** crea un *contexto de pila raíz*, pero otros elementos pueden establecer *contextos de pila locales*. Los contextos de pila son heredados. Un contexto de pila local es atómico; las cajas en otros contextos de pila no pueden aparecer entre ninguna de sus cajas.

Un elemento que establece un contexto de pila local genera una caja que tiene dos niveles de pila: uno para el contexto de pila que él crea (siempre '0') y uno para el contexto de pila al cual pertenece (dado por la propiedad 'z-index').

La caja de un elemento tiene el mismo nivel de pila que la caja de su padre a menos que se le dé un nivel de pila diferente con la propiedad 'z-index'.

### 9.9.1 Especificar el nivel de pila: la propiedad 'z-index'

#### 'z-index'

|              |                            |
|--------------|----------------------------|
| Valor:       | auto   <entero>   inherit  |
| Inicial:     | auto                       |
| Se aplica a: | los elementos posicionados |
| Se hereda:   | no                         |
| Porcentajes: | N/A                        |
| Medios:      | visuales                   |

Para una caja posicionada, la propiedad 'z-index' especifica:

1. El nivel de pila de la caja en el contexto de pila actual.
2. Si la caja establece un contexto de pila local.

Los valores tienen los siguientes significados:

#### <entero>

Este entero es el nivel de pila de la caja generada en el contexto de pila actual. La caja también establece un contexto de pila local en el cual su nivel de pila es '0'.

#### auto

El nivel de pila de la caja generada en el contexto de pila actual es el mismo que el de la caja de su padre. La caja no establece un nuevo contexto de pila local.

En el siguiente ejemplo, los niveles de pila de las cajas (nombradas con sus atributos "id") son: "texto2"=0, "imagen"=1, "texto3"=2 y "texto1"=3. El nivel de pila de "texto2" es heredado de la caja raíz. Los otros son especificados con la propiedad 'z-index'.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Posiciones en el orden Z</TITLE>
    <STYLE type="text/css">
      .pila {
        position: absolute;
        left: 2in;
        top: 2in;
        width: 3in;
        height: 3in;
      }
    </STYLE>
  </HEAD>
  <BODY>
    <P>
      <IMG id="imagen" class="pila"
        src="mariposa.gif" alt="La imagen de una mariposa"
        style="z-index: 1">

      <DIV id="texto1" class="pila"
        style="z-index: 3">
        Este texto quedará arriba de la imagen de la mariposa.
      </DIV>

      <DIV id="texto2">
        Este texto quedará abajo de todo.
      </DIV>
```

```
<DIV id="texto3" class="pila"
  style="z-index: 2">
  Este texto quedará abajo del texto1, pero encima de la imagen de la
  mariposa.
</DIV>
</BODY>
</HTML>
```

Este ejemplo demuestra la noción de *transparencia*. El comportamiento predeterminado de una caja es permitir que las cajas que quedan detrás sean visibles a través de las áreas transparentes en su contenido. En el ejemplo, cada caja se superpone transparentemente con las cajas debajo de ellas. Este comportamiento puede ser anulado usando una las [propiedades del fondo](#) existentes.

## 9.10 Dirección del texto: las propiedades 'direction' y 'unicode-bidi'

Los caracteres en ciertas escrituras se escriben de derecha a izquierda. En algunos documentos, en particular aquellos escritos en Arabe o en Hebreo, y en algunos contextos de lenguaje mixto, el texto de un único (mostrado visualmente) bloque puede aparecer con direccionalidad mixta. Este fenómeno se llama *bidireccionalidad*, o "bidi" para abreviar.

El estándar Unicode ([\[UNICODE\]](#), sección 3.11) define un complejo algoritmo para determinar la direccionalidad correcta del texto. El algoritmo se compone de una parte implícita basada en las propiedades de los caracteres, como así también de controles explícitos para las incrustaciones y sustituciones. CSS2 confía en este algoritmo para lograr un adecuado procesamiento bidireccional. Las propiedades 'direction' y 'unicode-bidi' permiten a los autores especificar cómo los elementos y atributos del lenguaje de un documento se vinculan con este algoritmo.

Si un documento contiene caracteres de derecha a izquierda, y si la aplicación del usuario muestra esos caracteres (con los signos apropiados, no con sustitutos arbitrarios como un signo de interrogación, un código hexadecimal, una caja negra, etc.), la aplicación del usuario debe aplicar el algoritmo bidireccional. Este requisito aparentemente parcial refleja el hecho de que, aunque no todos los documentos en Hebreo o Arabe contienen texto de direccionalidad mixta, tales documentos son mucho más propensos a contener texto de izquierda a derecha (ej., números, texto en otros lenguajes) que los documentos escritos en lenguajes de izquierda a derecha.

Debido a que la direccionalidad de un texto depende de la estructura y semántica del lenguaje del documento, estas propiedades deben en la mayoría de los casos ser usadas por los diseñadores de descripciones del tipo de documento (DTD), o autores de documentos especiales. Si una hoja de estilo predeterminada especifica estas propiedades, los autores y usuarios no deben especificar reglas para sustituirlas. Una excepción típica sería sustituir el comportamiento bidi en una aplicación del usuario si esa aplicación del usuario translitera el Yiddish (usualmente escrito con letras del hebreo) con caracteres latinos a pedido del usuario.

La especificación HTML 4.0 ([\[HTML40\]](#), sección 8.2) define el comportamiento de la bidireccionalidad en los elementos HTML. Las aplicaciones del usuario HTML [con conformidad](#) pueden por lo tanto ignorar las propiedades 'direction' y 'unicode-bidi' en las hojas de estilo de los autores y usuarios. Las reglas de hojas de estilo que conseguirían el comportamiento bidi especificado en [\[HTML40\]](#) están dadas en [el ejemplo de hoja de estilo](#). La especificación HTML 4.0 también contiene más información acerca del tema de la bidireccionalidad.

### 'direction'

- Valor: ltr | rtl | inherit
- Inicial: ltr
- Se aplica a: todos los elementos, pero ver el texto
- Se hereda: si
- Porcentajes: N/A
- Medios: visuales

Esta propiedad especifica la dirección básica de escritura de los bloques y la dirección de las incrustaciones y sustituciones (ver 'unicode-bidi') para el algoritmo bidireccional de Unicode. Además, especifica la dirección de la composición de las columnas en una [tabla](#), la dirección del [desbordamiento](#) horizontal y la posición de una última línea incompleta de un bloque en el caso de 'text-align: justify'.

Los valores para esta propiedad tiene los siguientes significados:

- ltr  
Dirección de izquierda a derecha (Left-To-Right).
- rtl  
Dirección de derecha a izquierda (Right-To-Left).

Para que la propiedad 'direction' tenga algún efecto en los elementos a nivel de línea, el valor de la propiedad 'unicode-bidi' debe ser 'embed' u 'override'.

**Nota.** La propiedad 'direction', cuando se especifica para las columnas de una tabla, no es heredado por las celdas de la columna ya que las columnas no existen en la estructura del documento. Debido a eso, CSS no puede capturar fácilmente las reglas de herencia del atributo "dir" descritas en [\[HTML40\]](#), sección 11.3.2.1.

'unicode-bidi'

Valor: normal | embed | bidi-override | inherit  
Inicial: normal  
Se aplica a: todos los elementos, pero ver más adelante  
Se hereda: no  
Porcentajes: N/A  
Medios: visuales

Los valores para esta propiedad tiene los siguientes significados:

normal

El elemento no abre un nivel adicional de incrustación con respecto al algoritmo bidireccional. Para los elementos a nivel de línea, el reordenamiento implícito funciona a través de todos los límites del elemento.

embed

Si el elemento es a nivel de línea, este valor abre un nivel adicional de incrustación con respecto al algoritmo bidireccional. La dirección de este nivel de incrustación está dada por la propiedad 'direction'. Dentro del elemento, el reordenamiento está dado implícitamente. Esto corresponde a agregar LRE (U+202A; para 'direction: ltr') o RLE (U+202B; para 'direction: rtl') al comienzo del elemento y PDF (U+202C) al final del elemento.

bidi-override

Si el elemento es a nivel de línea o es un elemento a nivel de bloque que sólo contiene elementos a nivel de línea, crea una sustitución. Esto significa que dentro del elemento, el reordenamiento es estrictamente en secuencia de acuerdo a la propiedad 'direction'; la parte implícita del algoritmo bidireccional se ignora. Esto corresponde a agregar LRO (U+202D; para 'direction: ltr') o RLO (U+202E; para 'direction: rtl') al comienzo del elemento y PDF (U+202C) al final del elemento.

El orden final de los caracteres en cada elemento a nivel de bloque es el mismo que si los códigos de control bidi se hubieran agregado como se describe arriba, las marcas se hubieran quitado y la secuencia de caracteres resultante hubiera pasado a una implementación del algoritmo bidireccional de Unicode para un texto puro que produce los mismos saltos de línea que el texto con estilo. En este proceso, la entidades no textuales como las imágenes son tratadas como caracteres neutros, a menos que su propiedad 'unicode-bidi' tenga un valor distinto a 'normal', en cuyo caso son tratadas como caracteres gruesos en la 'direction' especificada para el elemento.

Por favor observe que para conseguir un flujo de las cajas de línea con dirección uniforme (ya sea completamente de izquierda a derecha o completamente de derecha a izquierda), pueden crearse más cajas de línea (incluyendo cajas de línea anónimas) y algunas cajas de línea pueden tener que ser divididas y reordenadas antes de fluir.

Debido a que el algoritmo de Unicode tiene un límite de 15 niveles de incrustación, se debe cuidar de no usar 'unicode-bidi' con un valor distinto a 'normal' a menos que sea apropiado. En particular, un valor de 'inherit' debería usarse con extrema precaución. Sin embargo, para los elementos que son, en general, pensados para ser mostrados como bloques, se prefiere una definición de 'unicode-bidi: embed' para mantener el elemento junto en caso de que la visualización cambie a nivel de línea (ver el ejemplo de abajo).

El siguiente ejemplo muestra un documento XML con texto bidireccional. Ilustra un importante principio de diseño: los diseñadores de DTD deben tomar en cuenta la bidireccionalidad en el propio lenguaje (elementos y atributos) y en cualquier hoja de estilo que acompañe. Las hojas de estilo deben ser diseñadas de manera tal que las reglas bidi queden separadas de las otras reglas de estilo. Las reglas bidi no deben ser sustituidas por otras hojas de estilo de modo que el comportamiento de la bidireccionalidad del lenguaje del documento o la DTD sea preservada.

En este ejemplo, las letras en minúsculas se usan para los caracteres de izquierda a derecha heredados y las mayúsculas representan los caracteres de derecha a izquierda heredados:

```
<HEBREO>
  <PAR>HEBREO1 HEBREO2 español3 HEBREO4 HEBREO5</PAR>
  <PAR>HEBREO6 <ENF>HEBREO7</ENF> HEBREO8</PAR>
</HEBREO>
<ESPAÑOL>
  <PAR>español9 español10 español11 HEBREO12 HEBREO13</PAR>
  <PAR>español14 español15 español16</PAR>
  <PAR>español17 <HE-CITA>HEBREO18 español19 HEBREO20</HE-CITA></PAR>
</ESPAÑOL>
```

Como esto es XML, la hoja de estilo es la responsable de definir la dirección de escritura. Esta es la hoja de estilo:

```
/* Reglas para bidi */
HEBREO, HE-CITA {direction: rtl; unicode-bidi: embed}
ESPAÑOL         {direction: ltr; unicode-bidi: embed}

/* Reglas para la presentación */
HEBREO, ESPAÑOL, PAR {display: block}
ENF                  {font-weight: bold}
```

El elemento HEBREO es un bloque con una dirección básica de derecha a izquierda, el elemento ESPAÑOL es un bloque con una dirección básica de izquierda a derecha. Los PAR son bloques que heredan la dirección básica de sus padres. Así, los primeros dos PAR se leen comenzando desde arriba a la derecha, los tres finales se leen comenzando desde arriba a la izquierda. Por favor, observe que HEBREO y ESPAÑOL son elegidos como nombres de los elementos solamente por ser más explícitos, los nombres de los elementos pueden proporcionar la estructura sin referencia al idioma.

El elemento ENF es a nivel de línea y como su valor para 'unicode-bidi' es 'normal' (el valor inicial), no tiene ningún efecto en el ordenamiento del texto. El elemento HE-CITA, por otra parte, crea una incrustación.

El formato de este texto podría parecerse a esto si la línea es larga:

```

50ERBEH 40ERBEH español3 20ERBEH 10ERBEH
80ERBEH 70ERBEH 60ERBEH
español9 español10 español11 130ERBEH 120ERBEH
español14 español15 español16
español17 200ERBEH español19 180ERBEH
```

Observe que la incrustación HE-CITA provoca que HEBREO18 quede a la derecha de español19.

Si las líneas tienen que ser cortadas, sería más como esto:

```

20ERBEH 10ERBEH
-EH 40ERBEH español3
50ERB

-EH 70ERBEH 60ERBEH
80ERB

español9 español10 es-
pañol11 120ERBEH
130ERBEH

español14 español15
español16

español17 180ERBEH
200ERBEH español19
```

Como HEBREO18 debe leerse antes que español19, está en la línea encima de español19. Sólo cortando la línea larga de formato anterior no hubiera funcionado. Vea también que la primer sílaba de español19 podría caber en la línea anterior, pero el silabeo de las palabras de izquierda a derecha en un contexto de derecha a izquierda, y viceversa, es normalmente suprimido para evitar tener de mostrar un guión en el medio de la línea.

# 10 Detalles del modelo de formato visual

## 10.1 Definición de "bloque de contención"

La posición y tamaño de la(s) caja(s) de un elemento a veces son computadas en relación a cierto rectángulo, llamado el *bloque de contención* del elemento. El bloque de contención de un elemento es definido del siguiente modo:

1. El bloque de contención (llamado el bloque de contención inicial) en el cuál reside el **elemento raíz** es escogido por la aplicación del usuario.
2. Para los demás elementos, a menos que el elemento esté **posicionado absolutamente**, el bloque de contención está formado por el borde del contenido de la caja a nivel de bloque que es el antepasado más cercano.
3. Si el elemento tiene 'position: fixed', el bloque de contención es establecido por el **acceso visual**.
4. Si el elemento tiene 'position: absolute', el bloque de contención es establecido por el antepasado más cercano con una propiedad 'position' distinta a 'static', del siguiente modo:
  1. En el caso de que el antepasado sea a nivel de bloque, el bloque de contención estará formado por el **borde del relleno** del antepasado.
  2. En el caso de que el antepasado sea a nivel de línea, el bloque de contención dependerá de la propiedad 'direction' del antepasado:
    1. Si 'direction' es 'ltr', la parte superior izquierda del bloque de contención es la parte superior izquierda del borde del contenido de la primer caja generada por el antepasado, y la parte inferior derecha es la inferior derecha del borde del contenido de la última caja del antepasado.
    2. Si 'direction' es 'rtl', la parte superior derecha del bloque de contención es la parte superior derecha del borde del contenido de la primer caja generada por el antepasado, y la parte inferior izquierda es la inferior izquierda del borde del contenido de la última caja del antepasado.

Si no hay tal antepasado, el borde de la caja del contenido del elemento raíz establece el bloque de contención.

Sin posicionamiento, los bloques de contención (B.C.) en el siguiente documento:

```
<HTML>
  <HEAD>
    <TITLE>Ilustración de bloques de contención</TITLE>
  </HEAD>
  <BODY id="body">
    <DIV id="div1">
      <P id="p1">Este es texto en el primer párrafo...</P>
      <P id="p2">Este es texto <EM id="em1"> en el
      <STRONG id="strong1">segundo</STRONG> párrafo.</EM></P>
    </DIV>
  </BODY>
</HTML>
```

son establecidos como sigue:

| Para la caja generada por | B.C. es establecido por         |
|---------------------------|---------------------------------|
| body                      | B.C. inicial (depende de la AU) |
| div1                      | body                            |
| p1                        | div1                            |
| p2                        | div1                            |
| em1                       | p2                              |
| strong1                   | p2                              |

Si posicionamos "div1":

```
#div1 { position: absolute; left: 50px; top: 50px }
```

su bloque de contención no es más "body"; se convierte en el bloque de contención inicial (porque no hay otras cajas antepasadas posicionadas).

Si también posicionamos "em1":

```
#div1 { position: absolute; left: 50px; top: 50px }
#em1  { position: absolute; left: 100px; top: 100px }
```

la tabla de bloques de contención se convierte en:

| Para la caja generada por | B.C. es establecido por |
|---------------------------|-------------------------|
| body                      | B.C. inicial            |
| div1                      | B.C. inicial            |
| p1                        | div1                    |
| p2                        | div1                    |
| em1                       | div1                    |
| strong1                   | em1                     |

Posicionando "em1", su bloque de contención se convierte en la caja del antepasado posicionado más cercano (es decir, lo generado por "div1").

## 10.2 Ancho del contenido: la propiedad 'width'

### 'width'

- Valor: <medida> | <porcentaje> | auto | inherit
- Inicial: auto
- Se aplica a: todos los elementos excepto los elementos de línea no reemplazados, las filas de las tablas y los grupos de filas
- Se hereda: no
- Porcentajes: referidos al ancho del bloque de contención
- Medios: visuales

Esta propiedad especifica el [ancho del contenido](#) de las cajas generadas por elementos a nivel de bloque y [reemplazados](#).

Esta propiedad no se aplica a los elementos [a nivel de línea](#) no reemplazados. El ancho de las cajas de los elementos de línea no reemplazados es el de su contenido generado (*antes* de cualquier desplazamiento relativo de los hijos). Recuerde que esas cajas a nivel de línea fluyen en [cajas de línea](#). El ancho de las cajas de línea está dado por su [bloque de contención](#), pero puede ser acortado por la presencia de [flotantes](#).

El ancho de la caja de un elemento reemplazado es [intrínseco](#) y puede ser modificado por la aplicación del usuario si el valor de esta propiedad es distinto a 'auto'.

Los valores tienen los siguientes significados:

- <medida>**  
Especifica un ancho fijo.
- <porcentaje>**  
Especifica un ancho según un porcentaje. El porcentaje es calculado con respecto al ancho del *Specifies a percentage width. The percentage is* [bloque de contención](#) de la caja generada.
- auto**  
El ancho depende de los valores de otras propiedadess. Ver las secciones de abajo.

Los valores negativos para 'width' son ilegítimos.

Por ejemplo, la siguiente regla fija el ancho del contenido de los párrafos en 100 pixels:

```
P { width: 100px }
```

## 10.3 Computando anchos y márgenes

Los valores computados de las propiedades 'width', 'margin-left', 'margin-right', 'left' y 'right' de un elemento dependen del tipo de caja generada y entre sí. En principio, los valores computados son los mismos valores especificados, con 'auto' reemplazado por algún valor apropiado, pero existen excepciones. Las siguientes situaciones necesitan distinguirse:

1. elementos no reemplazados, de línea
2. elementos reemplazados, de línea
3. elementos no reemplazados en flujo normal, a nivel de bloque
4. elementos reemplazados en flujo normal, a nivel de bloque
5. elementos no reemplazados, flotantes
6. elementos reemplazados, flotantes
7. elementos no reemplazados, posicionados absolutamente
8. elementos reemplazados, posicionados absolutamente

Los puntos 1-6 incluyen posicionamiento relativo.



### 10.3.1 Elementos no reemplazados, de línea

La propiedad 'width' no se aplica. Un valor 'auto' especificado para 'left', 'right', 'margin-left' o 'margin-right' se convierte en un valor computado de '0'.

### 10.3.2 Elementos reemplazados, de línea

Un valor 'auto' especificado para 'left', 'right', 'margin-left' o 'margin-right' se convierte en un valor computado de '0'. Un valor 'auto' especificado para 'width' da el ancho [intrínseco](#) del elemento como valor computado.

### 10.3.3 Elementos no reemplazados en flujo normal, a nivel de bloque

Si 'left' o 'right' es dado como 'auto', su valor computado es 0. Las siguientes exigencias deben mantenerse entre las propiedades:

$$\text{'margin-left' + 'border-left-width' + 'padding-left' + 'width' + 'padding-right' + 'border-right-width' + 'margin-right'} = \text{ancho del } \text{bloque de contención}$$

(Si el estilo de borde es 'none', usa '0' como ancho de borde.) Si todos los anteriores tiene un valor especificado distinto a 'auto', se dice que los valores están "sobre-exigidos" y uno de los valores computados tendrá que ser diferente de su valor especificado. Si la propiedad 'direction' tiene el valor 'ltr', el valor especificado de 'margin-right' es ignorado y el valor es computado de modo que la igualdad resulte verdadera. Si el valor de 'direction' es 'rtl', esto a su vez sucede con 'margin-left'.

Si hay exactamente un valor especificado como 'auto', su valor computado proviene de la igualdad.

Si 'width' es puesto como 'auto', cualquier otro valor 'auto' se convierte en '0' y 'width' proviene de la igualdad resultante.

Si ambos, 'margin-left' y 'margin-right' son 'auto', sus valores computados son iguales.

### 10.3.4 Elementos reemplazados en flujo normal, a nivel de bloque

Si 'left' o 'right' es 'auto', su valor computado es 0. Si 'width' es especificado como 'auto', su valor es el ancho [intrínseco](#) del elemento. Si uno de los márgenes es 'auto', su valor computado está dado por las [exigencias](#) de más arriba. Además, si ambos márgenes son 'auto', sus valores computados son iguales.

### 10.3.5 Elementos no reemplazados, flotantes

Si 'left', 'right', 'width', 'margin-left' o 'margin-right' es especificado como 'auto', su valor computado es '0'.

### 10.3.6 Elementos reemplazados, flotantes

Si 'left', 'right', 'margin-left' o 'margin-right' es especificado como 'auto', su valor computado es '0'. Si 'width' es 'auto', su valor es el ancho [intrínseco](#) del elemento.

### 10.3.7 Elementos no reemplazados, posicionados absolutamente

La exigencia que determina los valores computados de estos elementos es:

$$\text{'left' + 'margin-left' + 'border-left-width' + 'padding-left' + 'width' + 'padding-right' + 'border-right-width' + 'margin-right' + 'right'} = \text{ancho del } \text{bloque de contención}$$

(Si el estilo de borde es 'none', se usa '0' como ancho de borde.) El cumplimiento de esta exigencia se alcanza a través de un número de sustituciones en el siguiente orden:

1. Si 'left' tiene el valor 'auto' mientras que 'direction' es 'ltr', se reemplaza 'auto' con la distancia desde el borde izquierdo del bloque de contención hasta el margen izquierdo de una hipotética caja que sería la primer caja del elemento si su propiedad 'position' hubiera sido 'static'. (Pero en vez de computar realmente esa caja, las aplicaciones del usuario son libres de suponer su probable posición.) El valor es negativo si la hipotética caja está a la izquierda del bloque de contención.
2. Si 'right' tiene el valor 'auto' mientras que 'direction' es 'rtl', se reemplaza 'auto' con la distancia desde el borde derecho del bloque de contención hasta el borde del margen derecho de la misma hipotética caja de arriba. El valor es positivo si la hipotética caja está a la izquierda del borde del bloque de contención.
3. Si 'width' es 'auto', se reemplaza cualquier 'auto' que reste para 'left' o 'right' con '0'.
4. Si 'left', 'right' o 'width' es (también) 'auto', se reemplaza cualquier 'auto' en 'margin-left' o 'margin-right' con '0'.
5. Si en este punto ambos, 'margin-left' y 'margin-right' son también 'auto', se resuelve la ecuación bajo la exigencia extra de que los dos márgenes deben tener valores iguales.
6. Si en este punto queda un sólo 'auto', se resuelve la ecuación para ese valor.
7. Si en este punto los valores están sobre-exigidos, se ignora el valor para 'left' (en el caso de que 'direction' sea 'rtl') o 'right' (en el caso de que 'direction' sea 'ltr') y se resuelve ese valor.

### 10.3.8 Elementos reemplazados, posicionados absolutamente

Esta situación es similar a la previa, excepto que el elemento tiene un ancho [intrínseco](#). La secuencia de sustituciones es ahora:

1. Si 'width' es 'auto', lo sustituye el ancho [intrínseco](#) del elemento.
2. Si 'left' tiene el valor 'auto' mientras que 'direction' es 'ltr', se reemplaza 'auto' con la distancia desde el borde izquierdo del bloque de contención hasta el borde del margen izquierdo de una hipotética caja que sería la primer caja del elemento si su propiedad 'position' hubiera sido 'static'. (Pero en vez de computar realmente esa caja, las aplicaciones del usuario son libres de suponer su probable posición.) El valor es negativo si la hipotética caja está a la izquierda del bloque de contención.
3. Si 'right' tiene el valor 'auto' mientras que 'direction' es 'rtl', se reemplaza 'auto' con la distancia desde el borde derecho del bloque de contención hasta el borde del margen derecho de la misma hipotética caja de arriba. El valor es positivo si la hipotética caja está a la izquierda del borde del bloque de contención.
4. Si 'left' o 'right' es 'auto', se reemplaza cualquier 'auto' para 'margin-left' o 'margin-right' con '0'.
5. Si en este punto ambos, 'margin-left' y 'margin-right' son también 'auto', se resuelve la ecuación bajo la exigencia extra de que los dos márgenes deben tener valores iguales.
6. Si en este punto queda un sólo 'auto', se resuelve la ecuación para ese valor.
7. Si en este punto los valores están sobre-exigidos, se ignora el valor para 'left' (en el caso de que 'direction' sea 'rtl') o 'right' (en el caso de que 'direction' sea 'ltr') y se resuelve ese valor.

### 10.4 Anchos mínimos y máximos: 'min-width' y 'max-width'

#### 'min-width'

*Valor:* <medida> | <porcentaje> | inherit  
*Inicial:* depende de la AU  
*Se aplica a:* todos los elementos excepto los elementos de línea no reemplazados y los elementos tabla  
*Se hereda:* no  
*Porcentajes:* referidos al ancho del bloque de contención  
*Medios:* visuales

#### 'max-width'

*Valor:* <medida> | <porcentaje> | none | inherit  
*Inicial:* none  
*Se aplica a:* todos los elementos excepto los elementos de línea no reemplazados y los elementos tabla  
*Se hereda:* no  
*Porcentajes:* referidos al ancho del bloque de contención  
*Medios:* visuales

Estas dos propiedades permiten a los autores restringir los anchos de las cajas a cierto rango. Los valores tienen los siguientes significados:

- <medida>  
Especifica un ancho mimimo o máximo fijo computado.
- <porcentaje>  
Especifica un porcentaje para determinar el valor computado. El porcentaje es calculado con respecto al ancho del [bloque de contención](#) de la caja generada.
- none  
(Solo en 'max-width') Ninguna limitación en el ancho de la caja.

El siguiente algoritmo describe cómo influyen las dos propiedades el [valor computado](#) de la propiedad 'width':

1. El ancho es computado (sin 'min-width' y 'max-width') siguiendo las reglas que se encuentran en ["Computando anchos y márgenes"](#) más arriba.
2. Si el valor computado de 'min-width' es mayor que el valor de 'max-width', a 'max-width' se le da el valor de 'min-width'.
3. Si el ancho computado es mayor que 'max-width', las reglas de [arriba](#) se aplican nuevamente, pero esta vez usando el valor de 'max-width' como el valor especificado para 'width'.
4. Si el ancho computado es menor que 'min-width', las reglas de [arriba](#) se aplican nuevamente, pero esta vez usando el valor de 'min-width' como el valor especificado para 'width'.

La aplicación del usuario puede definir un valor mínimo no negativo para la propiedad 'min-width', el que puede variar de elemento a elemento y aún depender de otras propiedades. Si 'min-width' queda bajo ese límite, ya sea porque fue definido explícitamente o porque es 'auto' y las reglas de arriba lo hacen demasiado pequeño, la aplicación del usuario puede usar el valor mínimo como valor computado.

### 10.5 Altura del contenido: la propiedad 'height'

#### 'height'

*Valor:* <medida> | <porcentaje> | auto | inherit  
*Inicial:* auto  
*Se aplica a:* todos los elementos excepto los elementos de línea no reemplazados, las columnas de la tabla y los grupos de columnas  
*Se hereda:* no

Porcentajes: ver el texto  
Medios: visuales

Esta propiedad especifica la [altura del contenido](#) de las cajas generadas por los elementos a nivel de bloque y [reemplazados](#).

Esta propiedad no se aplica a los elementos [a nivel de línea](#) no reemplazados. La altura de las cajas de los elementos de línea no reemplazados está dada por el valor (posiblemente heredado) 'line-height' del elemento.

Los valores tienen los siguientes significados:

- <medida>  
Especifica una altura fija.
- <porcentaje>  
Especifica una altura en porcentaje. El porcentaje es calculado con respecto a la altura del [bloque de contención](#) de la caja generada. Si la altura del bloque de contención no se especifica explícitamente (es decir, depende de la altura del contenido), el valor es interpretado como 'auto'.
- auto  
La altura depende de los valores de otras propiedades. Ver más abajo.

Los valores negativos para 'height' son ilegítimos.

Por ejemplo, la regla siguiente fija la altura de los párrafos en 100 pixels:

```
P { height: 100px }
```

Los párrafos que requieren más de 100 pixels de altura [desbordarán](#) de acuerdo a la propiedad 'overflow'.

## 10.6 Computando alturas y márgenes

Para computar los valores de 'top', 'margin-top', 'height', 'margin-bottom' y 'bottom' debe hacerse una distinción entre los distintos tipos de cajas:

1. elementos no reemplazados, de línea
2. elementos reemplazados, de línea
3. elementos no reemplazados en flujo normal, a nivel de bloque
4. elementos reemplazados en flujo normal, a nivel de bloque
5. elementos no reemplazados, flotantes
6. elementos reemplazados, flotantes
7. elementos no reemplazados, posicionados absolutamente
8. elementos reemplazados, posicionados absolutamente

Los puntos 1-6 incluyen posicionamiento relativo.

### 10.6.1 Elementos no reemplazados, de línea

Si 'top', 'bottom', 'margin-top' o 'margin-bottom' es 'auto', su valor computado es 0. La propiedad 'height' no se aplica, pero la altura de la caja está dada por la propiedad 'line-height'.

### 10.6.2 Elementos reemplazados, de línea, elementos reemplazados en flujo normal, a nivel de bloque y elementos reemplazados, flotantes

Si 'top', 'bottom', 'margin-top' o 'margin-bottom' es 'auto', su valor computado es 0. Si 'height' es 'auto', el valor computado es la altura [intrínseca](#).

### 10.6.3 Elementos no reemplazados en flujo normal, a nivel de bloque y elementos reemplazados, flotantes

Si 'top', 'bottom', 'margin-top', or 'margin-bottom' es 'auto', su valor computado es 0. Si 'height' es 'auto', la altura depende si el elemento tiene algún hijo a nivel de bloque y si tiene relleno o bordes. Si sólo tiene hijos a nivel de línea, la altura es la distancia desde la parte superior de la caja de línea que se encuentra más arriba hasta la base de la caja de línea que se encuentra más abajo. Si tiene hijos a nivel de bloque, es la distancia desde el límite del borde superior de la caja a nivel de bloque hija que se halla más arriba, hasta el límite del borde inferior de la caja a nivel de bloque hija que se halla más abajo. Sin embargo, si el elemento tiene un relleno superior distinto a cero y/o un borde superior, entonces el contenido comienza en el límite del margen superior de la hija ubicada más arriba. De manera similar, si el elemento tiene un relleno inferior distinto a cero y/o un borde inferior, entonces el contenido termina en el límite del margen inferior de la hija ubicada más abajo. Solamente los hijos en el flujo normal son tomados en cuenta (es decir, las cajas flotantes y las cajas posicionadas absolutamente son ignoradas, y las cajas posicionadas relativamente son consideradas sin sus desplazamientos). Observe que la caja hija puede ser una [caja anónima](#).

### 10.6.4 Elementos no reemplazados, posicionados absolutamente

Para los elementos absolutamente posicionados, las dimensiones verticales deben satisfacer esta condición:

$$'top' + 'margin-top' + 'border-top-width' + 'padding-top' + 'height' + 'padding-bottom' + 'border-bottom-width' + 'margin-bottom' + 'bottom' = \text{altura del bloque de contención}$$

(Si el estilo de borde es 'none', se usa '0' como ancho de borde.) La solución a esta condición se alcanza mediante algunas sustituciones en el siguiente orden:

1. Si 'top' tiene el valor de 'auto' se reemplaza con la distancia desde el borde superior del bloque de contención hasta el límite del margen superior de una hipotética caja que sería la primer caja del elemento si su propiedad 'position' hubiera sido 'static'. (Pero en vez de computar realmente esa caja, las aplicaciones del usuario son libres de suponer su probable posición.) El valor es negativo si la hipotética caja está arriba del bloque de contención.
2. Si tanto 'height' como 'bottom' son 'auto', se reemplaza 'bottom' por 0.
3. Si 'bottom' o 'height' es (también) 'auto', se reemplaza cualquier 'auto' en 'margin-top' o 'margin-bottom' por '0'.
4. Si en este punto tanto 'margin-top' como 'margin-bottom' son también 'auto', se resuelve la ecuación bajo la condición adicional de que los dos márgenes deben tener igual valor.
5. Si en este punto queda un solo 'auto', se resuelve la ecuación para ese valor.
6. Si en este punto los valores están sobre-exigidos, se ignora el valor de 'bottom' y se resuelve ese valor.

### 10.6.5 Elementos reemplazados, posicionados absolutamente

Esta situación es similar a la anterior, excepto que el elemento tiene una altura [intrínseca](#). La secuencia de sustituciones es ahora:

1. Si 'height' es 'auto', lo sustituye la altura [intrínseca](#) del elemento.
2. Si 'top' tiene el valor 'auto', se lo reemplaza con la distancia desde el borde superior de la caja de contención hasta el límite del margen superior de una hipotética caja que resultaría la primer caja del elemento si su propiedad 'position' hubiera sido 'static'. (Pero en vez de computar realmente esa caja, las aplicaciones del usuario son libres de suponer su probable posición.) El valor es negativo si la hipotética caja está encima del bloque de contención.
3. Si 'bottom' es 'auto', se reemplaza cualquier 'auto' en 'margin-top' o 'margin-bottom' por '0'.
4. Si en este punto tanto 'margin-top' como 'margin-bottom' son también 'auto', se resuelve la ecuación bajo la condición adicional de que los dos márgenes deben tener igual valor.
5. Si en este punto queda un solo 'auto', se resuelve la ecuación para ese valor.
6. Si en este punto los valores están sobre-exigidos, se ignora el valor de 'bottom' y se resuelve ese valor.

## 10.7 Alturas mínimas y máximas: 'min-height' y 'max-height'

A veces resulta útil restringir la altura de los elementos a cierto rango. Las dos propiedades ofrecen esta función:

#### 'min-height'

Valor: <medida> | <porcentaje> | inherit  
Inicial: 0  
Se aplica a: todos los elementos excepto los elementos de línea no reemplazados y los elementos tabla  
Se hereda: no  
Porcentajes: referidos a la altura del bloque de contención  
Medios: visuales

#### 'max-height'

Value: <medida> | <porcentaje> | none | inherit  
Inicial: none  
Se aplica a: todos los elementos excepto los elementos de línea no reemplazados y los elementos tabla  
Se hereda: no  
Porcentajes: referidos a la altura del bloque de contención  
Medios: visuales

Estas dos propiedades permiten a los autores restringir la altura de las cajas a un cierto rango. Los valores tienen los siguientes significados:

#### <medida>

Especifica una mínima o máxima altura computada fija.

#### <porcentaje>

Especifica un porcentaje para determinar el valor computado. El porcentaje se calcula con respecto a la altura del [bloque de contención](#) de la caja generada. Si la altura del bloque de contención se está especificada explícitamente (es decir, depende de la altura del contenido), el valor de porcentaje se interpreta como 'auto'.

#### none

(Sólo en 'max-height') Ninguna limitación a la altura de la caja.

El siguiente algoritmo describe cómo las dos propiedades influyen en el [valor computado](#) de la propiedad 'height':

1. La altura es computada (sin 'min-height' y 'max-height') siguiendo las reglas sobre "[Computando anchos y márgenes](#)" de más arriba.

- 2. Si el valor computado de 'min-height' es mayor que el valor de 'max-height', a 'max-height' se le da el valor de 'min-height'.
- 3. Si la altura computada es mayor que 'max-height', las reglas anteriores se aplican nuevamente, pero esta vez usando el valor de 'max-height' como el valor especificado para 'height'.
- 4. Si la altura computada es mayor que 'min-height', las reglas anteriores se aplican nuevamente, pero esta vez usando el valor de 'min-height' como el valor especificado para 'height'.

## 10.8 Cálculo de la altura de la línea: las propiedades 'line-height' y 'vertical-align'

Como se describe en la sección sobre [contexto de formato de línea](#), las aplicaciones del usuario fluyen las cajas a nivel de línea en una pila vertical de [cajas de línea](#). La altura de una caja de línea se determina del siguiente modo:

- 1. La altura de cada caja a nivel de línea de la caja de línea es calculada (ver "[Computando anchos y márgenes](#)" y la propiedad 'line-height').
- 2. Las cajas a nivel de línea son alineadas verticalmente de acuerdo a su propiedad 'vertical-align'.
- 3. La altura de la caja de línea es la distancia entre la parte superior de la caja ubicada más arriba y la base de la caja ubicada más abajo.

Los elementos a nivel de línea vacíos generan cajas a nivel de línea vacías, pero estas cajas aún tiene márgenes, relleno, bordes y una altura de línea, y en consecuencia influyen en este cálculo tanto como los elementos con contenido.

Observe que si todas las cajas en la caja de línea están alineadas por sus bases, la caja de línea tendrá exactamente la altura de la caja más alta. Sin embargo, si las cajas están alineadas por una línea de base común, la parte superior e inferior de la caja de línea puede no tocar la parte superior e inferior de la caja más alta.

### 10.8.1 Interlineado y medio interlineado

Como la altura de una caja a nivel de línea puede ser distinta al tamaño de la fuente del texto de la caja (por ej., 'line-height' > 1em), puede haber espacio por encima y por debajo de los signos procesados. La diferencia entre el tamaño de la fuente y el valor computado de 'line-height' se llama *interlineado*. La mitad del interlineado se llama *medio interlineado*.

Las aplicaciones del usuario centran verticalmente los signos en una caja a nivel de línea, agregando medio interlineado por encima y por debajo. Por ejemplo, si un fragmento de texto es de '12pt' de alto y el valor de 'line-height' es '14pt', 2 puntos de espacio extra deberían ser agregados: 1pt encima y 1pt abajo de las letras. (Esto se aplica también a las cajas vacías, como si la caja vacía contuviera una letra infinitamente condensada.)

Cuando el valor de 'line-height' es menor que el tamaño de la fuente, la altura final de la caja a nivel de línea será menor que el tamaño de la fuente y los signos procesados serán "sangrados" fuera de la caja. Si una de estas cajas tocan el borde de una caja de línea, los signos procesados también serán "sangrados" dentro de la caja de línea adyacente.

Aunque los márgenes, bordes y relleno de los elementos no reemplazados no entran en el cálculo de la altura de la caja a nivel de línea (y así tampoco en el cálculo de la caja de línea), son también procesados alrededor de las cajas a nivel de línea. Esto significa que si la altura de una caja de línea es más corta que los [límites externos](#) de las cajas que contiene, el fondo y el color del relleno y los bordes pueden ser "sangrados" dentro de las cajas de línea adyacentes. Sin embargo, en este caso, algunas aplicaciones del usuario pueden usar la caja de línea para "cortar" las áreas del borde y el relleno (es decir, no procesarlas).

#### 'line-height'

Valor: normal | <número> | <medida> | <porcentaje> | inherit  
Inicial: normal  
Se aplica a: todos los elementos  
Se hereda: si  
Porcentajes: referidos al tamaño de la fuente del propio elemento  
Medios: visuales

Si la propiedad es puesta en un elemento [a nivel de bloque](#) cuyo contenido está compuesto de elementos [a nivel de línea](#), especifica la altura *mínima* de cada caja a nivel de línea generada.

Si la propiedad es puesta en un elemento [a nivel de línea](#), especifica la altura *exacta* de cada caja generada por el elemento (Excepto para los elementos a nivel de línea [reemplazados](#), donde la altura de la caja está dada por la propiedad 'height'.)

Los valores para esta propiedad tienen los siguientes significados:

#### normal

Le dice a las aplicaciones del usuario que pongan como [valor computado](#) un valor "razonable" en base al tamaño de la fuente del elemento. El valor tiene el mismo significado que <número>. Recomendamos un valor computado para 'normal' entre 1.0 y 1.2.

#### <medida>

La altura de la caja es puesta con esta medida. Los valores negativos son ilícitos.

<número>

El [valor computado](#) de la propiedad es este número multiplicado por el tamaño de la fuente del elemento. Los valores negativos son ilícitos. De cualquier modo, el número, no el [valor computado](#), es heredado.

<porcentaje>

El [valor computado](#) de la propiedad es este porcentaje multiplicado por el tamaño computado de la fuente del elemento. Los valores negativos son ilícitos.

Las tres reglas en el ejemplo de abajo dan como resultado la misma altura de línea:

```
DIV { line-height: 1.2; font-size: 10pt }      /* número */
DIV { line-height: 1.2em; font-size: 10pt }    /* medida */
DIV { line-height: 120%; font-size: 10pt }     /* porcentaje */
```

Cuando un elemento contiene texto que es procesado con más de una fuente, las aplicaciones del usuario deben determinar el valor de 'line-height' de acuerdo con el tamaño de fuente mayor.

Generalmente, cuando sólo hay un valor de 'line-height' para todas las cajas a nivel de línea en un párrafo (y ninguna imagen alta), lo anterior asegurará que las líneas de base de las líneas sucesivas tienen propiedades 'line-height' exactamente separadas. Esto es importante cuando columnas de texto con distintas fuentes tienen que ser alineadas, por ejemplo en una tabla.

Observe que los elementos reemplazados tiene una propiedad 'font-size' y una 'line-height', aún cuando no son usadas directamente para determinar la altura de la caja. La propiedad 'font-size' es, sin embargo, usada para definir las unidades 'em' y 'ex', y 'line-height' tiene un papel en la propiedad 'vertical-align'.

'vertical-align'

- Valor: baseline | sub | super | top | text-top | middle | bottom | text-bottom | <porcentaje> | <medida> | inherit
- Inicial: baseline
- Se aplica a: los elementos a nivel de línea y 'table-cell'
- Se hereda: no
- Porcentajes: referidos a 'line-height' del propio elemento
- Medios: visuales

Esta propiedad afecta el posicionamiento vertical dentro de una caja de línea de las cajas generadas por un elemento a nivel de línea. Los siguientes valores sólo tienen significado con respecto a un elemento a nivel de línea padre, o a un elemento a nivel de bloque padre, si ese elemento genera [cajas de línea anónimas](#); no tienen ningún efecto si no existe tal padre.

**Nota.** Los valores de esta propiedad tienen significados ligeramente distintos en el contexto de tablas. Por favor, consulte la sección sobre [algoritmos para la altura de la tabla](#) por más detalles.

baseline

Encuadra la línea de base de la caja con la línea de base de la caja padre. Si la caja no tiene una línea de base, alinea la base de la caja con la línea de base del padre.

middle

Alinea el punto medio vertical de la caja con la línea de base de la caja padre más la mitad de la altura de la x del padre.

sub

Baja la línea de base de la caja hasta la posición apropiada para los subíndices de la caja del padre. (Este valor no tiene ningún efecto en el tamaño de la fuente del texto del elemento.)

super

Eleva la línea de base de la caja hasta la posición adecuada por los superíndices de la caja del padre. (Este valor no tiene ningún efecto en el tamaño de la fuente del texto del elemento.)

text-top

Alinea la parte superior de la caja con la parte superior de la fuente del elemento padre.

text-bottom

Alinea la base de la caja con la base de la fuente del elemento padre.

<porcentaje>

Eleva (valor positivo) o baja (valor negativo) la caja en esta distancia (un porcentaje del valor 'line-height'). El valor '0%' significa lo mismo que 'baseline'.

<medida>

Eleva (valor positivo) o baja (valor negativo) la caja en esta distancia. El valor '0cm' significa lo mismo que 'baseline'.

Los valores restantes se refieren a la caja de línea en la cuál aparece la caja generada:

top

Alinea la parte superior de la caja con la parte superior de la caja de línea.

bottom

Alinea la base de la caja con la base de la caja de línea.



# 11 Efectos visuales

## 11.1 Desbordamiento y recorte

Generalmente, el contenido de una caja de bloque es confinado a los límites del contenido de la caja. En ciertos caso, una caja puede *desbordar*, significando que su contenido queda parcial o completamente fuera de la caja, por ej.:

- Una línea no puede ser cortada, provocando que la caja de línea sea más ancha que la caja de bloque.
- Una caja a nivel de bloque es demasiado ancha para el bloque de contención. Esto puede suceder cuando la propiedad 'width' de un elemento tiene un valor que provoca que la caja de bloque generada se salga por los lados del bloque de contención.
- La altura de un elemento excede la altura explícitamente asignada al bloque de contención (es decir, la altura del bloque de contención está determinada por la propiedad 'height', no por la altura del contenido).
- Una caja está [absolutamente posicionada](#).
- Tiene [márgenes negativos](#).

Siempre que ocurra el desbordamiento, la propiedad 'overflow' especifica cómo (y si) una caja es recortada. La propiedad 'clip' especifica el tamaño y la forma de la zona de recorte. Especificar una zona de recorte pequeña puede provocar el recorte de contenidos en otro caso visibles.

### 11.1.1 Desbordamiento: la propiedad 'overflow'

#### 'overflow'

|              |  |
|--------------|--|
| Valor:       | visible   hidden   scroll   auto   inherit     |
| Inicial:     | visible  |
| Se aplica a: | los elementos a nivel de bloque y reemplazados |
| Se hereda:   | no   |
| Porcentajes: | N/A  |
| Medios:      | visual   |

Esta propiedad especifica si el contenido de un elemento a nivel de bloque es recortado cuando desborda la caja del elemento (que está actuando como un bloque de contención para el contenido). Los valores tienen los siguientes significados:

#### visible

Este valor indica que el contenido no es recortado, es decir, puede ser procesado fuera de la caja de bloque.

#### hidden

Este valor indica que el contenido es recortado y que ningún mecanismo de desplazamiento deberá proporcionarse para ver el contenido fuera de la zona de recorte; los usuarios no tendrán acceso al contenido recortado. El tamaño y forma de la zona de recorte son especificados por la propiedad 'clip'.

#### scroll

Este valor indica que el contenido es recortado y que si la aplicación del usuario utiliza un mecanismo de desplazamiento que sea visible en la pantalla (tal como una barra de desplazamiento), ese mecanismo deberá ser visualizado en una caja tenga o no ésta parte de su contenido recortado. Esto evita cualquier problema con las barras de desplazamiento apareciendo y desapareciendo en un entorno dinámico. Cuando este valor es especificado y el medio al que está dirigido es 'print' o 'projection', el contenido que desborda deberá imprimirse.

#### auto

El comportamiento del valor 'auto' depende de la aplicación del usuario, pero debe provocar que se proporcione un mecanismo de desplazamiento para las cajas desbordadas.

Aún si 'overflow' es puesto como 'visible', el contenido puede ser recortado a la ventana del documento de la AU por el entorno operativo nativo.

Considere el siguiente ejemplo de un bloque de cita (BLOCKQUOTE) que es demasiado grande para su bloque de contención (establecido por DIV). Aquí está el documento fuente:

```
<DIV class="cita">
<BLOCKQUOTE>
<P>No me gustó la obra, pero la ví
bajo condiciones adversas -el telón estaba levantado-.
<DIV class="atribuido-a">- Groucho Marx</DIV>
</BLOCKQUOTE>
</DIV>
```

Aquí está la hoja de estilo que controla el tamaño y estilo de las cajas generadas:



```
DIV.cita { width : 100px; height: 100px;
border: thin solid red;
}

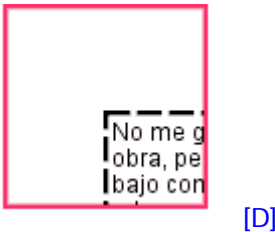
BLOCKQUOTE { width : 125px; height : 100px;
margin-top: 50px; margin-left: 50px;
border: thin dashed black
}

DIV.atribuido-a { text-align : right; }
```

El valor inicial de 'overflow' es 'visible', de modo que BLOCKQUOTE sera procesado sin recortar, algo como esto:



Poniendo 'overflow' como 'hidden' para el elemento DIV, por otro lado, provoca que BLOCKQUOTE sea recortado por el bloque de contención:



Un valor de 'scroll' le indicaría a las AU que soportan mecanismos de desplazamiento visibles que muestren uno para que los usuarios puedan acceder al contenido recortado.

11.1.2 Recorte: la propiedad 'clip'

Una zona de recorte define qué porción del contenido procesado de un elemento es visible. De manera predeterminada, la zona de recorte tiene el mismo tamaño y forma que la(s) caja(s) del elemento. Sin embargo, la zona de recorte puede ser modificada por la propiedad 'clip'.

|              |  |
|--------------|--|
| 'clip'       |  |
| Valor:       | <forma>   auto   inherit                       |
| Inicial:     | auto   |
| Se aplica a: | los elementos a nivel de bloque y reemplazados |
| Se hereda:   | no   |
| Porcentajes: | N/A  |
| Medios:      | visual   |

La propiedad 'clip' se aplica a elementos que tienen una propiedad 'overflow' con un valor diferente a 'visible'. Los valores tienen los siguientes significados:

auto

La zona de recorte tiene el mismo tamaño y ubicación que la(s) caja(s) del elemento.

<forma>

En CSS2, el único valor permitido para <forma> es: rect (<arriba> <derecha> <abajo> <izquierda>) donde <arriba>, <abajo> <derecha> e <izquierda> especifican los desplazamientos de los lados respectivos de la caja.

<arriba>, <derecha>, <abajo> e <izquierda> pueden tener un valor de <medida> o 'auto'. Las medidas negativas están permitidas. El valor 'auto' significa que un determinado borde de la zona de recorte será el mismo que el borde de la caja generada del elemento (es decir, 'auto' significa lo mismo que '0').

Cuando las coordenadas se redondean a coordenadas en pixeles, se debe tener cuidado de que ningún pixel permanezca visible cuando <izquierdo> + <derecho> es igual al ancho del elemento (o <arriba> + <abajo> es

igual a la altura del elemento), y recíprocamente que ningún pixel permanezca oculto cuando estos valores sean 0.

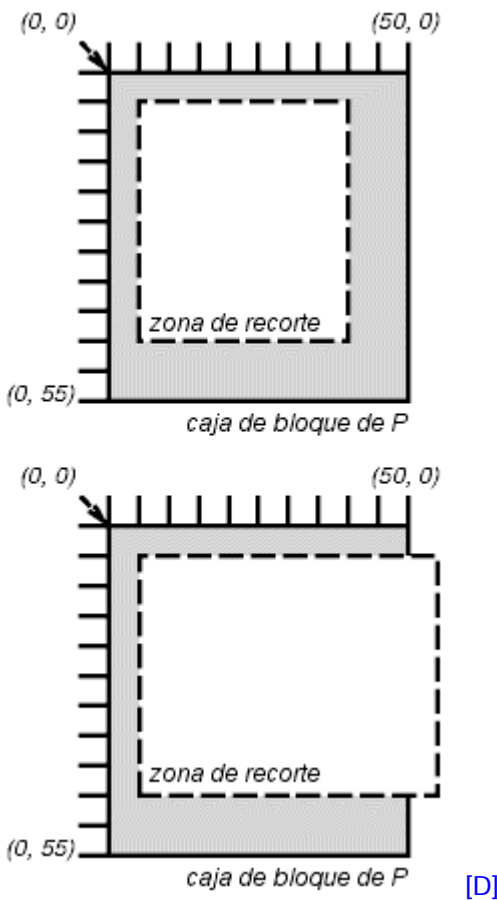
Los antepasados del elemento también pueden tener zonas de recorte (en el caso de que su propiedad 'overflow' no sea 'visible'); lo que se procesa es la intersección de las distintas zonas de recorte.

Si la zona de recorte excede los límites de la ventana del documento de la AU, el contenido puede ser recortado a esa ventana por el entorno operativo nativo.

Las dos reglas siguientes:

```
P { clip: rect(5px, 10px, 10px, 5px); }
P { clip: rect(5px, -5px, 10px, 5px); }
```

crearán las zonas de recorte rectangulares delimitadas por las líneas cortadas en las ilustraciones siguientes:



**Nota.** En CSS2, todas las zonas de recorte son rectangulares. Anticipamos futuras extensiones para permitir recortes no rectangulares.

## 11.2 Visibilidad: la propiedad 'visibility'

### 'visibility'

|              |                                       |
|--------------|---------------------------------------|
| Valor:       | visible   hidden   collapse   inherit |
| Inicial:     | inherit                               |
| Se aplica a: | todos los elementos                   |
| Se hereda:   | no                                    |
| Porcentajes: | N/A                                   |
| Medios:      | visual                                |

La propiedad 'visibility' especifica si las cajas generadas por un elemento son procesadas. Las cajas invisibles siguen afectando la composición (poner la propiedad 'display' como 'none' para suprimir completamente la generación de la caja). Los valores tienen los siguientes significados:

#### visible

La caja generada es visible.

#### hidden

La caja generada es invisible (totalmente transparente), pero sigue afectando la composición.

#### collapse

Favor de consultar la sección sobre [efectos dinámicos en filas y columnas](#) en las tablas. Si se usa en otros elementos que no sean filas o columnas, 'collapse' tiene el mismo significado que 'hidden'.

Esta propiedad puede ser usada conjuntamente con scripts para crear efectos dinámicos.

En el siguiente ejemplo, presionando cualquier botón del formulario se invoca una función de script definida por el usuario que provoca que la caja correspondiente se vuelva visible y la otra se oculte. Como estas cajas tiene el mismo tamaño y posición, el efecto es que una reemplaza a la otra. (El código del script está en un lenguaje de script hipotético. Puede o no tener algún efecto en una AU apta para CSS.)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
<HEAD>
<STYLE type="text/css">
<!--
    #contenedor1 { position: absolute;
                  top: 2in; left: 2in; width: 2in }
    #contenedor2 { position: absolute;
                  top: 2in; left: 2in; width: 2in;
                  visibility: hidden; }
-->
</STYLE>
</HEAD>
<BODY>
<P>Elija un sospechoso:</P>
<DIV id="contenedor1">
    <IMG alt="Al Capone"
         width="100" height="100"
         src="sospechosol.jpg">
    <P>Nombre: Al Capone</P>
    <P>Residencia: Chicago</P>
</DIV>

<DIV id="contenedor2">
    <IMG alt="Lucky Luciano"
         width="100" height="100"
         src="sospechoso2.jpg">
    <P>Nombre: Lucky Luciano</P>
    <P>Residencia: New York</P>
</DIV>

<FORM method="post"
       action="http://www.suspect.org/process-bums">
    <P>
    <INPUT name="Capone" type="button"
          value="Capone"
          onclick='show("contenedor1");hide("contenedor2")'>
    <INPUT name="Luciano" type="button"
          value="Luciano"
          onclick='show("contenedor2");hide("contenedor1")'>
    </FORM>
</BODY>
</HTML>
```

# 12 Contenido generado, numeración automática y listas

En ciertos casos, los autores pueden desear que las aplicaciones del usuario procesen contenidos que no vienen en la [estructura del documento](#). Un ejemplo familiar de esto es una lista numerada; el autor no quiere hacer la lista de números explícitamente, el o ella quiere que la aplicación del usuario la genere automáticamente. De modo similar, los autores pueden querer que la aplicación del usuario inserte la palabra "Figura" antes del encabezado de una imagen, o "Capítulo 7" antes del séptimo título del capítulo. Para audio o braille en particular, las aplicaciones del usuario deben ser capaces de insertar estas cadenas.

En CSS2, el contenido puede ser generado a través de distintos mecanismos:

- La propiedad 'content', junto con los pseudo-elementos :before y :after.
- Las propiedades sonoras 'cue-before', 'cue-after' (ver el capítulo sobre [hojas de estilo auditivas](#)). Cuando la propiedad 'content' se combina con las propiedades sonoras, éstas son procesadas en el siguiente orden: :before, 'cue-before', ('pause-before'), el contenido del elemento, ('pause-after'), 'cue-after' y :after.
- Elementos con un valor de 'list-item' para la propiedad 'display'.

Abajo describimos los mecanismos asociados con la propiedad 'content'.

## 12.1 Los pseudo-elementos :before y :after

Los autores especifican el estilo y la ubicación del contenido generado con los pseudo-elementos :before y :after. Como sus nombres lo indica, los pseudo-elementos :before (antes) y :after (después) especifica la ubicación del contenido antes o después del contenido de la [estructura del documento](#) de un elemento. La propiedad 'content', junto con estos pseudo-elementos, especifican lo que se inserta.

Por ejemplo, la siguiente regla inserta la cadena "Nota: " antes del contenido de cada elemento P cuyo atributo "class" tenga el valor "nota":

```
P.nota:before { content: "Nota: " }
```

El formato de los objetos (ej., cajas) generados por un elemento incluye el contenido generado. De modo que, por ejemplo, cambiando la hoja de estilo anterior a:

```
P.nota:before { content: "Nota: " }
P.nota        { border: solid green }
```

causaría que un borde verde sólido fuera procesado alrededor de todo el párrafo, incluyendo la cadena inicial.

Los pseudo-elementos :before y :after [heredan](#) cualquier propiedad hereditable del elemento en la estructura del documento al cuál están ligados.

Por ejemplo, las siguientes reglas insertan comillas de apertura antes de cada elemento Q. El color de la comillas será rojo, pero la fuente será la misma que la del resto del elemento Q:

```
Q:before {
  content: open-quote;
  color: red
}
```

En la declaración de un pseudo-elemento :before o :after, las propiedades no hereditables toman sus [valores iniciales](#).

De modo que, por ejemplo, debido a que el valor inicial de la propiedad 'display' es 'inline', las comillas en el ejemplo previo se insertan como una caja a nivel de línea (es decir, en la misma línea que el contenido de texto inicial del elemento). El siguiente ejemplo pone la propiedad 'display' como 'block', de modo que el texto insertado se convierte en un bloque:

```
BODY:after {
  content: "Fin";
  display: block;
  margin-top: 2em;
  text-align: center;
}
```

Observe que una aplicación del usuario auditiva leería la palabra "Fin" después que el resto del contenido de BODY.

Las aplicaciones del usuario deben ignorar las siguientes propiedades con los pseudo-elementos :before y :after: 'position', 'float', propiedades de [list](#) y propiedades de [table](#).

Los pseudo-elementos :before y :after permiten valores de la propiedad 'display' como sigue:

- Si el [sujeto](#) del selector es un elemento [a nivel de bloque](#), los valores permitidos son 'none', 'inline', 'block' y 'marker'. Si el valor de 'display' es cualquier otro valor, el pseudo-elemento se comportará como si el valor fuera 'block'.
- Si el [sujeto](#) del selector es un elemento [a nivel de línea](#), los valores permitidos son 'none' e 'inline'. Si el valor de 'display' es cualquier otro valor, el pseudo-elemento se comportará como si el valor fuese 'inline'.

*Nota. Otros valores podrán ser permitidos en futuros niveles de CSS.*

## 12.2 La propiedad 'content'

### 'content'

|              |  |
|--------------|--|
| Value:       | [ <cadena>   <uri>   <contador>   attr(X)   open-quote   close-quote   no-open-quote   no-close-quote ]+   inherit |
| Inicial:     | una cadena vacía   |
| Se aplica a: | los pseudo-elementos :before y :after  |
| Se hereda:   | no   |
| Porcentajes: | N/A  |
| Medios:      | todos  |

Esta propiedad se usa con los pseudo-elementos :before y :after para generar contenido en un documento. Los valores tienen los siguientes significados:

#### <cadena>

Contenido del texto (ver la sección sobre [cadenas](#)).

#### <uri>

El valor es un URI que señala un recurso externo. Si una aplicación del usuario no puede soportar el recurso debido a los [tipos de medios](#) que soporta, debe ignorar el recurso. **Nota.** CSS no ofrece ningún mecanismo para cambiar el tamaño del objeto incrustado o para proporcionar una descripción textual, tal como los atributos "alt" o "longdesc" hacen con las imágenes en HTML. Esto puede cambiar en futuros niveles de CSS.

#### <contador>

Los [contadores](#) pueden especificarse mediante dos funciones distintas: 'counter()' o 'counters()'. La sintaxis tiene dos formas: 'counter(nombre)' o 'counter(nombre, estilo)'. El texto generado es el valor del contador nombrado en este punto en la estructura del formato; su formato está con el [estilo](#) indicado ('decimal' por defecto). La última función también tiene dos formas: 'counters(nombre, cadena)' o 'counters(nombre, cadena, estilo)'. El texto generado es el valor de todos los contadores con el nombre dado en este punto en la estructura del formato, separados por la cadena especificada. El contador es procesado con el [estilo](#) indicado ('decimal' por defecto). Ver la sección sobre [contadores y numeración automática](#) para mayor información.

#### open-quote y close-quote

Estos valores son reemplazados con la correspondiente cadena de la propiedad 'quotes'.

#### no-open-quote y no-close-quote

No inserta nada (una cadena vacía), pero incrementa (disminuye) el nivel de anidamiento de las comillas.

#### attr(X)

Esta función devuelve como una cadena el valor del atributo X del sujeto del selector. La cadena no es analizada por el procesador de CSS. Si el sujeto del selector no tiene un atributo X, se devuelve una cadena vacía. La distinción entre mayúsculas y minúsculas en el nombre de los atributos depende del lenguaje del documento. **Nota.** En CSS2, no es posible referirse a valores de atributos de otros elementos del selector.

La propiedad 'display' controla si el contenido es ubicado en una caja de bloque, a nivel de línea o marcador.

Los autores deben poner las declaraciones de 'content' en reglas [@media](#) cuando el contenido es sensible al medio. Por ejemplo, un texto literal puede ser utilizado para cualquier grupo de medios, pero las imágenes sólo se aplican a los grupos de medios visual + bitmap, y los archivos de sonido sólo se aplican al grupo de medios aural.

La siguiente regla produce que un archivo de sonido sea ejecutado al final de una cita (ver la sección sobre [hojas de estilo auditivas](#) por mecanismos adicionales):

```
@media aural {
  BLOCKQUOTE:after { content: url("musica-bella.wav") }
}
```

La siguiente regla inserta el texto del atributo HTML "alt" antes de la imagen. Si la imagen no es mostrada, el lector todavía verá el texto "alt".

```
IMG:before { content: attr(alt) }
```

Los autores pueden incluir saltos de línea en el contenido generado escribiendo la secuencia de escape "\A" en una de las cadenas luego de la propiedad 'content'. Esto inserta un *corte de línea forzado*, similar al elemento BR en HTML. Ver "[Cadenas](#)" y "[Caracteres y mayúsculas/minúsculas](#)" para más información sobre la secuencia de escape "\A".

```
H1:before {
  display: block;
  text-align: center;
  content: "capítulo\A chapter\A chapitre"
}
```

El contenido generado no altera la estructura del documento. En particular, no retroalimenta al procesador del lenguaje del documento (ej., para otro análisis).

*Nota. En futuros niveles de CSS, la propiedad 'content' podrá aceptar valores adicionales, permitiéndole variar el estilo de parte del contenido generado, pero en CSS2, todo el contenido de los pseudo-elementos :before o :after tiene el mismo estilo.*

## 12.3 Interacción de :before y :after con los elementos 'compact' y 'run-in'

Los siguientes casos pueden ocurrir:

1. **Un elemento 'run-in' o 'compact' tiene un pseudo-elemento :before de tipo 'inline':** el pseudo-elemento es tomado en cuenta cuando el tamaño de la caja del elemento es computado (para 'compact') y es procesado dentro de la misma caja de bloque que el elemento.
2. **Un elemento 'run-in' o 'compact' tiene un pseudo-elemento :after de tipo 'inline':** Se aplican las reglas del punto anterior.
3. **Un elemento 'run-in' o 'compact' tiene un pseudo-elemento :before de tipo 'block':** el pseudo-elemento es estructurado como un bloque encima del elemento, y no toma parte en el cómputo del tamaño del elemento (para 'compact').
4. **Un elemento 'run-in' o 'compact' tiene un pseudo-elemento :after de tipo 'block':** tanto el elemento como su pseudo-elemento :after son estructurados como cajas de bloque. El elemento *no* se estructura como una caja a nivel de línea de su propio pseudo-elemento :after.
5. **El elemento que sigue a un elemento 'run-in' o 'compact' tiene un :before de tipo 'block':** la decisión sobre cómo estructurar el elemento 'run-in'/'compact' se hace con respecto a la caja de bloque que es el resultado del pseudo-elemento :before.
6. **El elemento que sigue a un elemento 'run-in' o 'compact' tiene un :before de tipo 'inline':** la decisión sobre cómo estructurar el elemento 'run-in'/'compact' depende del valor 'display' del elemento al que :before está ligado.

Aquí hay un ejemplo de un encabezado 'run-in' con un pseudo-elemento :after, seguido por un párrafo con un pseudo-elemento :before. Todos los pseudo-elementos son a nivel de línea (por defecto) en este ejemplo. Cuando la hoja de estilo:

```
H3 { display: run-in }
H3:after { content: ": " }
P:before { content: "... " }
```

se aplica a este documento fuente:

```
<H3>Centauros</H3>
<P>tienen pezuñas
<P>tienen una cola
```

El formato visual se parecerá a:

```
Centauros: ... tienen pezuñas
... tienen una cola
```

## 12.4 Comillas

En CSS2, los autores pueden especificar, de un modo sensible al estilo y dependiente del contexto, cómo las aplicaciones del usuario deben procesar las comillas. La propiedad 'quotes' especifica pares de comillas para cada nivel de citas incrustadas. La propiedad 'content' da acceso a esas comillas y provoca que sean insertadas antes o después de una cita.

## 12.4.1 Especificación de comillas con la propiedad 'quotes'

### 'quotes'

|              |                                       |
|--------------|---------------------------------------|
| Valor:       | [<cadena> <cadena>]+   none   inherit |
| Inicial:     | depende de la aplicación del usuario  |
| Se aplica a: | todos los elementos                   |
| Se hereda:   | si                                    |
| Porcentajes: | N/A                                   |
| Medios:      | visuales                              |

Esta propiedad especifica las comillas para cualquier cantidad de citas incrustadas. Los valores tienen los siguientes significados:

#### none

Los valores 'open-quote' y 'close-quote' de la propiedad 'content' no provocan ninguna comillas.

#### [<cadena> <cadena>]+

Los valores para los valores 'open-quote' y 'close-quote' de la propiedad 'content' son tomados de esta lista de pares de comillas (de apertura y cierre). El primer (más a la izquierda) par representa el nivel más externo de la cita, el segundo par el primer nivel de incrustación, etc. La aplicación del usuario debe aplicar el par de comillas adecuado de acuerdo al nivel de incrustación.

Por ejemplo, aplicando la siguiente hoja de estilo:

```
/* Especifica pares de comillas para dos niveles en dos idiomas */
Q:lang(en) { quotes: '""' '""' '""' '""' }
Q:lang(no) { quotes: "«" "»" "<" ">" }

/* Inserta las comillas antes y después del contenido del elemento Q */
Q:before { content: open-quote }
Q:after { content: close-quote }
```

al siguiente fragmento HTML:

```
<HTML lang="en">
  <HEAD>
    <TITLE>Quotes</TITLE>
  </HEAD>
  <BODY>
    <P><Q>Quote me!</Q>
  </BODY>
</HTML>
```

permitiría a una aplicación del usuario producir:

```
"Quote me!"
```

mientras que este fragmento HTML:

```
<HTML lang="no">
  <HEAD>
    <TITLE>Quotes</TITLE>
  </HEAD>
  <BODY>
    <P><Q>Trøndere gråter når <Q>Vinsjan på kaia</Q> blir deklamert.</Q>
  </BODY>
</HTML>
```

produciría:

```
«Trøndere gråter når <Vinsjan på kaia> blir deklamert.»
```

**Nota.** Mientras que las comillas especificadas por 'quotes' en los ejemplos anteriores están convenientemente colocadas en los teclados de computadora, las componedoras de texto de alta calidad requerirían diferentes caracteres de ISO 10646. La siguiente tabla informativa lista algunos de los caracteres de comillas de ISO 10646:



| Procesamiento aproximado | Código ISO 10646 (hex) | Descripción                                  |
|--------------------------|------------------------|--|
| "                        | 0022                   | COMILLAS [comillas dobles en ASCII]          |
| '                        | 0027                   | Apóstrofe [comilla simple en ASCII]          |
| <                        | 2039                   | COMILLA ANGULAR SIMPLE HACIA LA IZQUIERDA    |
| >                        | 203A                   | COMILLA ANGULAR SIMPLE HACIA LA DERECHA      |
| «                        | 00AB                   | COMILLAS ANGULARES DOBLES HACIA LA IZQUIERDA |
| »                        | 00BB                   | COMILLAS ANGULARES DOBLES HACIA LA DERECHA   |
| `                        | 2018                   | COMILLA APERTURA SIMPLE [simple alta-6]      |
| '                        | 2019                   | COMILLA CIERRE SIMPLE [simple alta-9]        |
| ``                       | 201C                   | COMILLAS APERTURA DOBLES [doble alta-6]      |
| "                        | 201D                   | COMILLAS CIERRE DOBLES [doble alta-9]        |
| ''                       | 201E                   | COMILLAS DOBLES BAJA-9 [doble baja-9]        |

### 12.4.2 Inserción de comillas con la propiedad 'content'

Las comillas son insertadas en los lugares correspondientes de un documento con los valores 'open-quote' y 'close-quote' de la propiedad 'content'. Cada aparición de 'open-quote' o 'close-quote' se reemplaza por una de las cadenas del valor de 'quotes', en base a la profundidad del anidado.

'Open-quote' se refiere a la primera de un par de comillas, 'close-quote' se refiere a la segunda. Qué par de comillas se usa depende del nivel de anidado de las comillas: el número de apariciones de 'open-quote' en todo el texto generado antes de la presente, menos el número de apariciones de 'close-quote'. Si la profundidad es 0, se usa el primer par, si la profundidad es 1, se usa el segundo par, etc. Si la profundidad es mayor que el número de pares, el último par se repite.

Observe que esta profundidad de las citas es independiente del anidado del documento fuente o de la estructura del formato.

Algunos estilos tipográficos requieren comillas de apertura que se repiten antes de cada párrafo de una cita que abarca varios párrafos, pero sólo el último párrafo termina con comillas de cierre. En CSS, esto puede lograrse insertando comillas de cierre "fantasmas". La palabra clave 'no-close-quote' disminuye el nivel de las citas, pero no inserta una comilla.

La siguiente hoja de estilo pone comillas de apertura en cada párrafo de un elemento BLOCKQUOTE, e inserta una sola comilla al final:

```
BLOCKQUOTE P:before      { content: open-quote }
BLOCKQUOTE P:after       { content: no-close-quote }
BLOCKQUOTE P.último:after { content: close-quote }
```

Esto cuenta con que el último párrafo sea marcado con una clase "último", debido a que no hay selectores que puedan equivaler al último hijo de un elemento.

Simétricamente, hay también una palabra clave 'no-open-quote', que no inserta nada, pero incrementa en uno la profundidad de las citas.

**Nota.** Si una cita está en un idioma distinto al del texto circundante, se acostumbra entrecomillar el texto con las comillas del idioma del texto circundante, no con las del idioma de la propia cita.

Por ejemplo, francés dentro de inglés:

```
The device of the order of the garter is "Honi soit qui mal y pense."
```

Inglés dentro de francés:

```
Il disait: « Il faut mettre l'action en < fast forward >.»
```

Una hoja de estilo como la siguiente pondrá la propiedad 'quotes' de manera tal que 'open-quote' y 'close-quote' funcionen correctamente para todos los elementos. Estas reglas son para documentos que contienen sólo inglés y francés, o ambos. Se necesita una regla para cada idioma adicional. Observe el uso del combinador hijo (">") para poner las citas en los elementos en base al idioma del texto circundante:

```
[LANG|=fr] > * { quotes: "«" "»" "\2039" "\203A" }
[LANG|=en] > * { quotes: "\201C" "\201D" "\2018" "\2019" }
```

Las comillas para el inglés se muestran aquí en una forma que la mayoría de las personas podrán teclear. Si puede escribirlas directamente, se parecerán a esto:

```
[LANG|=fr] > * { quotes: "«" "»" "<" ">" }
[LANG|=en] > * { quotes: "\"" "\"\" \"'\" \"/" }
```

## 12.5 Contadores y numeración automática

La numeración automática en CSS2 es controlada con dos propiedades, 'counter-increment' y 'counter-reset'. Los contadores definidos por estas propiedades se usan con las funciones counter() y counters() de la propiedad 'content'.

### 'counter-reset'

*Valor:* [ <identificador> <entero>? ]+ | none | inherit  
*Inicial:* none  
*Se aplica a:* todos los elementos  
*Se hereda:* no  
*Porcentajes:* N/A  
*Medios:* todos

### 'counter-increment'

*Valor:* [ <identificador> <entero>? ]+ | none | inherit  
*Inicial:* none  
*Se aplica a:* todos los elementos  
*Se hereda:* no  
*Porcentajes:* N/A  
*Medios:* todos

La propiedad 'counter-increment' acepta uno o más nombres de contadores (identificadores), cada uno seguido de manera opcional por un entero. El entero indica en cuanto se incrementará el contador con cada aparición del elemento. El incremento predeterminado es 1. Los enteros negativos o el cero son admitidos.

La propiedad 'counter-reset' también contiene una lista de uno o más nombres de contadores, cada uno seguido de manera opcional por un entero. El entero da el valor en que el contador es colocado con cada aparición del elemento. El valor por defecto es 0.

Si 'counter-increment' se refiere a un contador que no está en el [área de alcance \(ver abajo\)](#) de ningún 'counter-reset', se asume que el contador ha sido restablecido a 0 por el elemento raíz.

Este ejemplo muestra una forma de numerar los capítulos y secciones con "Capítulo 1", "1.1", "1.2", etc.

```
H1:before {
  content: "Capítulo " counter(capítulo) ". ";
  counter-increment: capítulo; /* Agrega 1 a capítulo */
  counter-reset: section; /* Pone la sección a 0 */
}
H2:before {
  content: counter(capítulo) "." counter(section) " ";
  counter-increment: section;
}
```

Si un elemento incrementa/restablece un contador y también lo usa (en la propiedad 'content' de su pseudo-elemento :before o :after), el contador se usa *después* de ser incrementado/restablecido.

Si un elemento restablece y también incrementa un contador, el contador es primero restablecido y luego incrementado.

La propiedad 'counter-reset' sigue las reglas sobre cascada. De modo que, debido a la cascada, la siguiente hoja de estilo:

```
H1 { counter-reset: section -1 }
H1 { counter-reset: imaginario 99 }
```

solamente restablecerá 'imaginario'. Para restablecer ambos contadores, estos tienen que ser especificados al mismo tiempo:

```
H1 { counter-reset: section -1 imaginario 99 }
```

### 12.5.1 Contadores anidados y área de alcance

Los contadores son "autoanidados", en el sentido de que la re-utilización de un contador en un elemento hijo crea una nueva instancia del contador. Esto es importante en situaciones como las listas en HTML, donde los

elementos puede ser anidados dentro de sí mismos hasta una profundidad arbitraria. Resultaría imposible definir a los contadores con un nombre único para cada nivel.

Así, lo siguiente basta para numerar los ítems de una lista anidada. El resultado es muy similar al de poner 'display:list-item' y 'list-style: inside' en el elemento LI:

```
OL { counter-reset: item }
LI { display: block }
LI:before { content: counter(item) ". "; counter-increment: item }
```

El autoanidado se basa en el principio de que cada elemento que tiene 'counter-reset' para un contador X, crea un nuevo contador X, el *área de alcance* del cual es el elemento, sus hermanos precedentes y todos los descendientes del elemento y sus hermanos precedentes.

En el ejemplo de arriba, un OL creará un contador y todos los hijos de OL se referirán a ese contador.

Si señalamos con item[n] la nº instancia del contador "item" y con "(" y ")" el principio y fin de un área de alcance, entonces el siguiente fragmento de HTML utilizará los contadores indicados. (Asumimos una hoja de estilo como la proporcionada en el ejemplo de arriba.)

```
<OL>                                <!-- (pone item[0] a 0          -->
  <LI>item                          <!-- incrementa item[0] (= 1)  -->
  <LI>item                          <!-- incrementa item[0] (= 2)  -->
    <OL>                            <!-- (pone item[1] a 0          -->
      <LI>item                      <!-- incrementa item[1] (= 1)  -->
      <LI>item                      <!-- incrementa item[1] (= 2)  -->
      <LI>item                      <!-- incrementa item[1] (= 3)  -->
        <OL>                      <!-- (pone item[2] a 0          -->
          <LI>item                 <!-- incrementa item[2] (= 1)  -->
        </OL>                    <!-- )                          -->
      <OL>                        <!-- (pone item[3] a 0          -->
        <LI>                      <!-- incrementa item[3] (= 1)  -->
      </OL>                      <!-- )                          -->
    <LI>item                      <!-- incrementa item[1] (= 4)  -->
  </OL>                          <!-- )                          -->
  <LI>item                      <!-- incrementa item[0] (= 3)  -->
  <LI>item                      <!-- incrementa item[0] (= 4)  -->
</OL>                            <!-- )                          -->
<OL>                            <!-- (restablece item[4] a 0     -->
  <LI>item                      <!-- incrementa item[4] (= 1)  -->
  <LI>item                      <!-- incrementa item[4] (= 2)  -->
</OL>                            <!-- )                          -->
```

La función 'counters()' genera una cadena compuesta por los valores de todos los contadores con el mismo nombre, separados por una cadena dada.

La siguiente hoja de estilo numera los ítems anidados de la lista como "1", "1.1", "1.1.1", etc.

```
OL { counter-reset: item }
LI { display: block }
LI:before { content: counters(item, "."); counter-increment: item }
```

### 12.5.2 Estilos de contadores

Por defecto, los contadores son compuestos con números decimales, pero todos los estilos disponibles para la propiedad 'list-style-type' están también disponibles para los contadores. La sintaxis es:

```
counter(nombre)
```

para el estilo predeterminado, o:

```
counter(nombre, 'list-style-type')
```

Todos los estilos están permitidos, incluyendo 'disc', 'circle', 'square' y 'none'.

```
H1:before      { content: counter(chno, upper-latin) ". " }
H2:before      { content: counter(section, upper-roman) " - " }
BLOCKQUOTE:after { content: " [" counter(bq, hebrew) "]" }
```

```
DIV.note:before { content: counter(notecntr, disc) " " }
P:before { content: counter(p, none) }
```

### 12.5.3 Contadores en elementos con 'display: none'

Un elemento que no es mostrado (con 'display' puesto como 'none') no puede incrementar o restablecer un contador.

Por ejemplo, con la siguiente hoja de estilo, los H2 de la clase "secreto" no incrementan a 'cont2'.

```
H2.secreto {counter-increment: cont2; display: none}
```

Los elementos con 'visibility' puesta como 'hidden', por otro lado, *sí* incrementan los contadores.

## 12.6 Marcadores y listas

La mayoría de los elementos a nivel de bloque en CSS generan una caja de bloque principal. En esta sección, trataremos dos mecanismos de CSS que provocan que un elemento genere dos cajas: una caja de bloque [principal](#) (para el contenido del elemento) y una caja de marcador separada (para decoraciones tales como viñetas, imágenes o números). La caja del marcador puede estar ubicada dentro o fuera de la caja principal. Al contrario del contenido de :before y :after, la caja del marcador no afecta la posición de la caja principal, cualquiera sea el esquema de posicionamiento.

El más general de los dos mecanismos es nuevo en CSS2 y se llama *marcadores*. El mecanismo más limitado involucra las propiedades [list](#) de CSS1. Las propiedades de las listas le dan resultados rápidos a los autores en muchas situaciones comunes de listas ordenadas y desordenadas. Sin embargo, los marcadores le brindan a los autores un control preciso sobre el contenido y posición del marcador. Los marcadores pueden ser usados con los [contadores](#) para crear un nuevo estilo de lista, para numerar notas al margen, y mucho más.

Por ejemplo, el siguiente ejemplo ilustra cómo los marcadores pueden ser usados para agregar puntos después de cada ítem de una lista numerada. Este programa HTML y esta hoja de estilo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Creando una lista con marcadores</TITLE>
    <STYLE type="text/css">
      LI:before {
        display: marker;
        content: counter(micontador, lower-roman) ".";
        counter-increment: micontador;
      }
    </STYLE>
  </HEAD>
  <BODY>
    <OL>
      <LI> Este es el primer ítem.
      <LI> Este es el segundo ítem.
      <LI> Este es el tercer ítem.
    </OL>
  </BODY>
</HTML>
```

producirían algo semejante a esto:

```
i. Este es el primer ítem.
ii. Este es el segundo ítem.
iii. Este es el tercer ítem.
```

Con los [selectores de descendientes](#) y los [selectores de hijos](#), es posible especificar distintos tipos de marcadores dependiendo de la profundidad de las listas incrustadas.

### 12.6.1 Marcadores: la propiedad 'marker-offset'

Los marcadores son creados colocando la propiedad 'display' como 'marker' dentro de un pseudo-elemento :before o :after. Mientras que el contenido 'block' e 'inline' de :before y :after es parte de la [caja principal](#) generada por el elemento, el contenido de 'marker' es compuesto en una caja de marcador independiente, fuera de la caja principal. Las cajas de marcadores son compuestas en una sola línea (es decir, una [caja de línea](#)), de modo que no son tan flexibles como los [flotantes](#). La caja del marcador sólo es creada si la propiedad 'content' para el pseudo-elemento realmente genera un contenido.

Las cajas de marcadores tienen relleno y bordes, pero no márgenes.

Para el pseudo-elemento `:before`, la línea de base del texto de la caja del marcador será alineada verticalmente con la línea de base del texto de la primer línea del contenido de la caja principal. Si la caja principal no contiene ningún texto, el límite externo superior de la caja del marcador será alineado con el límite externo superior de la caja principal. Para el pseudo-elemento `:after`, la línea de base del texto de la caja del marcador será alineada verticalmente con la línea de base del texto de la última línea del contenido de la caja principal. Si la caja principal no contiene ningún texto, el límite externo inferior de la caja del marcador será alineado con el límite externo inferior de la caja principal.

La altura de una caja de marcador está determinada por la propiedad `'line-height'`. La caja del marcador `:before` (`:after`) interviene en el cálculo de la altura de la primer (última) caja de línea de la caja principal. De este modo, los marcadores son alineados con la primer y última línea del contenido de un elemento, aún cuando las cajas de marcadores residen en distintas cajas de línea. Si no existe ninguna primer o última caja de línea en la caja principal, la caja del marcador establece su caja de línea sola.

La alineación vertical de una caja de marcador dentro de su caja de línea se especifica con la propiedad `'vertical-align'`.

Si el valor de la propiedad `'width'` es `'auto'`, el [ancho del contenido](#) de la caja de marcador es el del contenido, en caso contrario es el valor de `'width'`. Para los valores de `'width'` menores al ancho del contenido, la propiedad `'overflow'` especifica la conducta del desbordamiento. Las cajas de marcadores pueden superponerse con las cajas principales. Para los valores de `'width'` mayores que el ancho del contenido, la propiedad `'text-align'` determina la alineación horizontal del contenido en la caja del marcador.

La propiedad `'marker-offset'` especifica el desplazamiento horizontal entre una caja de marcador y la [caja principal](#) asociada. La distancia es tomada entre sus [límites de bordes](#) más cercanos. **Nota.** Si un marcador fluye a la derecha de un flotante en un contexto de formato de izquierda a derecha, la caja principal fluirá a lo largo del costado derecho del flotante, pero las cajas de marcadores aparecerán a la izquierda del flotante. Como el borde izquierdo de la caja principal queda a la izquierda del flotante (ver la descripción de [flotantes](#)), y las cajas de marcadores quedan fuera de los límites del borde de la caja principal, el marcador también quedará a la izquierda del flotante. Un comportamiento análogo se aplica en los formatos de derecha a izquierda cuando un marcador fluye a la izquierda de un flotante.

Cuando la propiedad `'display'` tiene el valor `'marker'` para el contenido generado por un elemento con `'display: list-item'`, una caja de marcador generada por `:before` reemplaza al marcador normal del ítem de la lista.

En el siguiente ejemplo, el contenido es centrado dentro de una caja de marcador de un ancho fijo. Este documento:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Alineación del contenido en la caja del marcador</TITLE>
    <STYLE type="text/css">
      LI:before {
        display: marker;
        content: "(" counter(counter) ";
        counter-increment: counter;
        width: 6em;
        text-align: center;
      }
    </STYLE>
  </HEAD>
  <BODY>
    <OL>
      <LI> Este es el primer ítem.
      <LI> Este es el segundo ítem.
      <LI> Este es el tercer ítem.
    </OL>
  </BODY>
</HTML>
```

produciría algo semejante a esto:

- (1) Este es el primer ítem.
- (2) Este es el segundo ítem.
- (3) Este es el tercer ítem.

El siguiente ejemplo crea marcadores antes y después de los ítems de una lista.

Este documento:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Marcadores antes y después de los ítems de una lista</TITLE>
    <STYLE type="text/css">
      @media screen, print {
        LI:before {
          display: marker;
          content: url("sonrisa.gif");
        }
        LI:after {
          display: marker;
          content: url("triste.gif");
        }
      }
    </STYLE>
  </HEAD>
  <BODY>
    <UL>
      <LI>el primer ítem de la lista viene primero
      <LI>el segundo ítem de la lista viene después
    </UL>
  </BODY>
</HTML>
```

produciría algo como esto (los dibujos hechos en ascii se usan aquí en lugar de las imágenes gif de las caritas):

```
:-) el primer ítem de la
    lista viene primero  :-(
:-) el segundo ítem de la
    lista viene después  :-(
```

El ejemplo siguiente utiliza marcadores para numerar las notas (párrafos).

El siguiente documento:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Marcadores para crear notas numeradas</TITLE>
    <STYLE type="text/css">
      P { margin-left: 12 em; }
      @media screen, print {
        P.Nota:before {
          display: marker;
          content: url("nota.gif")
                    "Nota " counter(contador-nota) ":";
          counter-increment: contador-nota;
          text-align: left;
          width: 10em;
        }
      }
    </STYLE>
  </HEAD>
  <BODY>
    <P>El primer párrafo de este documento.</P>
    <P CLASS="Nota">Este documento es muy corto.</P>
    <P>Este es el final.</P>
  </BODY>
</HTML>
```

produciría algo así:

```
El primer párrafo
de este documento.

Note 1: Este documento
        es muy corto.

Este es el final.
```

'marker-offset'

Valor: <medida> | auto | inherit  
Inicial: auto  
Se aplica a: los elementos con 'display: marker'  
Se hereda: no  
Porcentajes: N/A  
Medios: visuales

Esta propiedad especifica la distancia entre los [límites de bordes](#) más cercanos de una caja de marcador y su [caja principal](#) asociada. El desplazamiento puede ser una (<medida>) especificada por el usuario o elegido por la AU ('auto'). Las medidas pueden ser negativas, pero pueden existir limitaciones propias de la implementación.

El ejemplo siguiente ilustra cómo los marcadores pueden utilizarse para agregar puntos después de cada ítem de una lista numerada. Este programa HTML y esta hoja de estilo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    <TITLE>Ejemplo de marcadores</TITLE>
    <STYLE type="text/css">
      P { margin-left: 8em } /* Hace espacio para los contadores */
      LI:before {
        display: marker;
        marker-offset: 3em;
        content: counter(micontador, lower-roman) ".";
        counter-increment: micontador;
      }
    </STYLE>
  </HEAD>
  <BODY>
    <P> Este es un largo párrafo anterior ...
    <OL>
      <LI> Este es el primer ítem.
      <LI> Este es el segundo ítem.
      <LI> Este es el tercer ítem.
    </OL>
    <P> Este es un largo párrafo posterior ...
  </BODY>
</HTML>
```

produciría algo semejante a:

```

    Este es un largo
    párrafo anterior ...

    i.   Este es el primer ítem.
    ii.  Este es el segundo ítem.
    iii. Este es el tercer ítem.

    Este es un largo
    párrafo posterior ...
```

12.6.2 Listas: las propiedades 'list-style-type', 'list-style-image', 'list-style-position' y 'list-style'

Las *propiedades de lista* permiten un formato visual básico en las listas. Al igual que con los marcadores más generales, un elemento con 'display: list-item' genera una [caja principal](#) para el contenido del elemento y una caja de marcador opcional. Las otras propiedades de lista posibilitan a los autores especificar el tipo de marcador (imagen, signo o número) y su posición con respecto a la caja principal (afuera o dentro de ella antes del contenido). No permiten a los autores especificar distintos estilos (color, fuente, alineación, etc.) para los marcadores de listas o ajustar su posición con respecto a la caja principal.

Además, cuando un marcador M (creado con 'display: marker') se usa con el ítem de una lista creada por las propiedades de lista, M reemplaza al marcador estándar del ítem de la lista.

Con las propiedades de lista, las [propiedades del fondo](#) se aplican solamente a la caja principal; una caja de marcador 'outside' es transparente. Los marcadores ofrecen mayor control sobre el estilo de la caja del marcador.

'list-style-type'

Valor: disc | circle | square | decimal | decimal-leading-zero | lower-roman | upper-roman | lower-greek | lower-alpha | lower-latin | upper-alpha | upper-latin | hebrew | armenian | georgian | cjk-ideographic | hiragana | katakana | hiragana-iroha | katakana-iroha | none | inherit



*Inicial:* disc  
*Se aplica a:* los elementos con 'display: list-item'  
*Se hereda:* si  
*Porcentajes:* N/A  
*Medios:* visuales

Esta propiedad especifica la apariencia del marcador de los ítems de la lista si 'list-style-image' tiene el valor 'none' o si la imagen señalada por el URI no puede ser mostrada. El valor 'none' especifica ningún marcador, de lo contrario hay tres tipos de marcadores: signos, sistemas numéricos y sistemas alfabéticos. **Nota.** Las listas numeradas mejoran la accesibilidad del documento haciendo a las listas más fáciles de navegar.

Los signos son especificados con **disc**, **circle** y **square**. Su procesamiento exacto depende de la aplicación del usuario.

Los sistemas numéricos son especificados con:

- decimal**  
Números decimales, comenzando de 1.
- decimal-leading-zero**  
Números decimales completados con ceros iniciales (ej., 01, 02, 03, ..., 98, 99).
- lower-roman**  
Números romanos en minúsculas (i, ii, iii, iv, v, etc.).
- upper-roman**  
Números romanos en mayúsculas (I, II, III, IV, V, etc.).
- hebrew**  
Numeración hebrea tradicional.
- georgian**  
Numeración georgiana tradicional (an, ban, gan, ..., he, tan, in, in-an, ...).
- armenian**  
Numeración armenia tradicional.
- cjk-ideographic**  
Números ideográficos planos
- hiragana**  
a, i, u, e, o, ka, ki, ...
- katakana**  
A, I, U, E, O, KA, KI, ...
- hiragana-iroha**  
i, ro, ha, ni, ho, he, to, ...
- katakana-iroha**  
I, RO, HA, NI, HO, HE, TO, ...

Una aplicación del usuario que no reconoce un sistema numérico debe usar 'decimal'.

***Nota. Este documento no especifica el mecanismo exacto de cada sistema numérico (ej., como se calculan los números romanos). Una futura Nota del W3C puede proporcionar mayores aclaraciones.***

Los sistemas alfabéticos se especifican con:

- lower-latin** o **lower-alpha**  
Letras minúsculas en ascii (a, b, c, ... z).
- upper-latin** o **upper-alpha**  
Letras mayúsculas en ascii (A, B, C, ... Z).
- lower-greek**  
Minúsculas griegas clásicas alfa, beta, gamma, ... (•, •, •, ...)

Esta especificación no define cómo los sistemas alfabéticos se desenvuelven al final del alfabeto. Por ejemplo, luego de 26 ítems en una lista, el procesamiento de 'lower-latin' es indefinido. Por eso, para listas extensas, recomendamos que los autores especifiquen números reales.

Por ejemplo, el siguiente documento en HTML:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Numeración con latinas minúsculas</TITLE>
    <STYLE type="text/css">
      OL { list-style-type: lower-roman }
    </STYLE>
  </HEAD>
  <BODY>
    <OL>
      <LI> Este es el primer ítem.
      <LI> Este es el segundo ítem.
      <LI> Este es el tercer ítem.
    </OL>
```

```
</BODY>
</HTML>
```

puede producir algo así:

```
    i Este es el primer ítem.
   ii Este es el segundo ítem.
  iii Este es el tercer ítem.
```

Observe que la alineación de los marcadores de listas (aquí, justificados a la derecha) depende de la aplicación del usuario.

**Nota.** Futuras versiones de CSS podrán proporcionar mecanismos más completos para los estilos de numeración internacionales.

**'list-style-image'**

```
Valor:      <uri> | none | inherit
Inicial:    none
Se aplica a: los elementos con 'display: list-item'
Se hereda:  si
Porcentajes: N/A
Medios:     visuales
```

Esta propiedad define la imagen que será usada como marcador de ítems de una lista. Cuando la imagen está disponible, reemplazará el marcador definido con el marcador 'list-style-type'.

El siguiente ejemplo determina que el marcador al comienzo de cada ítem de una lista sea la imagen "elipse.png".

```
UL { list-style-image: url("http://png.com/ellipse.png") }
```

**'list-style-position'**

```
Valor:      inside | outside | inherit
Inicial:    outside
Se aplica a: los elementos con 'display: list-item'
Se hereda:  si
Porcentajes: N/A
Medios:     visuales
```

Esta propiedad especifica la posición de la caja del marcador en la caja de bloque principal. Los valores tienen los siguientes significados:

**outside**

La caja del marcador está fuera de la caja de bloque principal. **Nota.** CSS1 no especificaba la ubicación precisa de la caja de marcador y por razones de compatibilidad, CSS2 se mantiene igualmente ambiguo. Para un control más preciso de las cajas de marcadores, por favor utilice los marcadores.

**inside**

La caja del marcador está en la primer caja a nivel de línea en la caja de bloque principal, luego de la cuál fluye el contenido del elemento.

Por ejemplo:

```
<HTML>
  <HEAD>
    <TITLE>Comparación entre inside/outside</TITLE>
    <STYLE type="text/css">
      UL          { list-style: outside }
      UL.compact { list-style: inside }
    </STYLE>
  </HEAD>
  <BODY>
    <UL>
      <LI>el primer ítem de la lista viene primero
      <LI>el segundo ítem de la lista viene después
    </UL>

    <UL class="compact">
      <LI>el primer ítem de la lista viene primero
      <LI>el segundo ítem de la lista viene después
    </UL>
  </BODY>
</HTML>
```

El ejemplo de arriba puede ser compuesto como:

- el primer ítem de la lista viene primero
- el segundo ítem de la lista viene después

---

- el primer ítem de la lista viene primero
- el segundo ítem de la lista viene después

Los costados izquierdos de las cajas de los ítems de la lista no son afectados por la ubicación de los marcadores.

[D]

En un texto de derecha a izquierda, los marcadores habrían estado al costado derecho de la caja.

'list-style'

Valor:

[ <'list-style-type'> || <'list-style-position'> || <'list-style-image'> ] | inherit

Inicial:

no definido para las propiedades resumidas

Se aplica a:

los elementos con 'display: list-item'

Se hereda:

si

Porcentajes:

N/A

Medios:

visuales

La propiedad 'list-style' es una fórmula resumida para definir las tres propiedades 'list-style-type', 'list-style-image' y 'list-style-position' en un mismo lugar en la hoja de estilo.

```
UL { list-style: upper-roman inside } /* Cualquier UL */
UL > UL { list-style: circle outside } /* Cualquier UL hijo de un UL */
```

Aunque los autores pueden especificar la información sobre 'list-style' directamente en los elementos ítems de una lista (ej., LI en HTML), deben hacerlo con cuidado. Las siguientes reglas parecen similares, pero la primera declara un [selector de descendiente](#) y la segunda un (más específico) [selector de hijo](#).

```
OL.alfa LI { list-style: lower-alpha } /* Cualquier LI descendiente de un OL */
OL.alfa > LI { list-style: lower-alpha } /* Cualquier LI hijo de un OL */
```

Los autores que usan solamente el [selector de descendiente](#) pueden no lograr los resultados que esperan. Considere las reglas siguientes:

```
<HTML>
  <HEAD>
    <TITLE>AVISO: Resultados inesperados debido a la cascada</TITLE>
    <STYLE type="text/css">
      OL.alfa LI { list-style: lower-alpha }
      UL LI      { list-style: disc }
    </STYLE>
  </HEAD>
  <BODY>
    <OL class="alfa">
      <LI>nivel 1
      <UL>
        <LI>nivel 2
      </UL>
    </OL>
  </BODY>
</HTML>
```

El procesamiento deseado tendría los ítems de lista del nivel 1 con etiquetas 'lower-alpha' y los ítems del nivel 2 con etiquetas 'disc'. Sin embargo, el [orden de la cascada](#) provocará que la primer regla de estilo (que incluye información acerca de la clase específica) enmascare a la segunda. Las siguientes reglas solucionan el problema empleando en cambio un [selector de hijo](#):

```
OL.alfa > LI { list-style: lower-alpha }
UL LI { list-style: disc }
```

Otra solución sería especificar la información sobre 'list-style' solamente en los elementos tipo de lista:

```
OL.alfa { list-style: lower-alpha }
UL { list-style: disc }
```

La herencia transferirá los valores de 'list-style' de los elementos OL y UL a los elementos LI. Este es el camino recomendado para especificar la información acerca del estilo de lista.

Un valor URI puede combinarse con cualquier otro valor, como en:

```
UL { list-style: url("http://png.com/ellipse.png") disc }
```

En el ejemplo anterior, 'disc' será utilizado cuando la imagen no sea disponible.

Un valor de 'none' para la propiedad 'list-style' pone a 'list-style-type' y a 'list-style-image' como 'none':

```
UL { list-style: none }
```

El resultado es que ningún marcador de ítem de lista es mostrado.

# 13 Medios paginados

## 13.1 Introducción a los medios paginados

Los medios paginados (por ej., papel, transparencias, páginas que son mostradas en la pantalla de la computadora, etc.) se diferencian de los [medios continuos](#) en que el contenido del documento se divide en una o más páginas discretas. Para manejar los saltos de página, CSS2 extiende el [modelo de formato visual](#) del siguiente modo:

1. La [caja de página](#) extiende el [modelo de caja](#) para permitir a los autores especificar el tamaño de una página, sus márgenes, etc.
2. El *modelo de página* extiende el [modelo de formato visual](#) para considerar los [saltos de página](#).

El modelo de página de CSS2 especifica cómo es el formato de un documento dentro de un área rectangular --la [caja de página](#)-- que tiene una ancho y una altura finitos. La caja de página no necesariamente corresponde a la *hoja* donde el documento será procesado finalmente (papel, transparencia, pantalla, etc.). El modelo de página de CSS especifica el formato en la caja de página, pero es responsabilidad de la aplicación del usuario transferir la caja de página a la hoja. Algunas posibilidades de transferencia incluyen:

- Transferencia de una caja de página a una hoja (ej., impreso de una sola cara).
- Transferencia de dos cajas de página a ambos lados de la misma hoja (ej., impreso de doble faz).
- Transferencia de N (pequeñas) cajas de página a una hoja.
- Transferencia de una (gran) caja de página a N x M hojas (llamado "mosaico").
- Creación de imposiciones. Una imposición es un grupo de páginas impresas sobre una hoja, que, luego de plegadas y cortadas como un libro, aparecen en la secuencia correcta.
- Impresión de un documento en varias bandejas de salida.
- Producir la salida a un archivo.

Aunque CSS2 no especifica el modo en que las aplicaciones del usuario transfieren las cajas de página a las hojas, incluye ciertos mecanismos para informar a las aplicaciones del usuario acerca del [tamaño y orientación](#) de la hoja de destino.

## 13.2 Cajas de página: la regla @page

La *caja de página* es una región rectangular que contiene dos áreas:

- El *área de la página*. El área de la página incluye las cajas dispuestas en esa página. Los límites del área de la página actúa como el [bloque de contención](#) inicial de la disposición efectuada entre saltos de páginas.
- El área del margen, que rodea al área de la página.

**Nota.** En CSS2, las [propiedades del borde](#) y las [propiedades del relleno](#) no se aplican a las páginas; puede suceder en el futuro.

Los autores especifican las dimensiones, orientación, márgenes, etc. de una caja de página dentro de una regla @page. Una regla @page consiste en una palabra clave "@page", un selector de página (seguido con ningún espacio intermedio por una pseudo-clase de página opcional) y un bloque de declaraciones (hechas para el *contexto de página*).

El *selector de página* especifica a qué páginas se aplican las declaraciones. En CSS2, los selectores de páginas pueden indicar la primer página, todas las páginas izquierdas, todas las páginas derechas, o una página con un nombre específico.

Las dimensiones de la caja de la página son fijadas con la propiedad 'size'. Las dimensiones del área de la página son las dimensiones de la caja de página menos el área del margen.

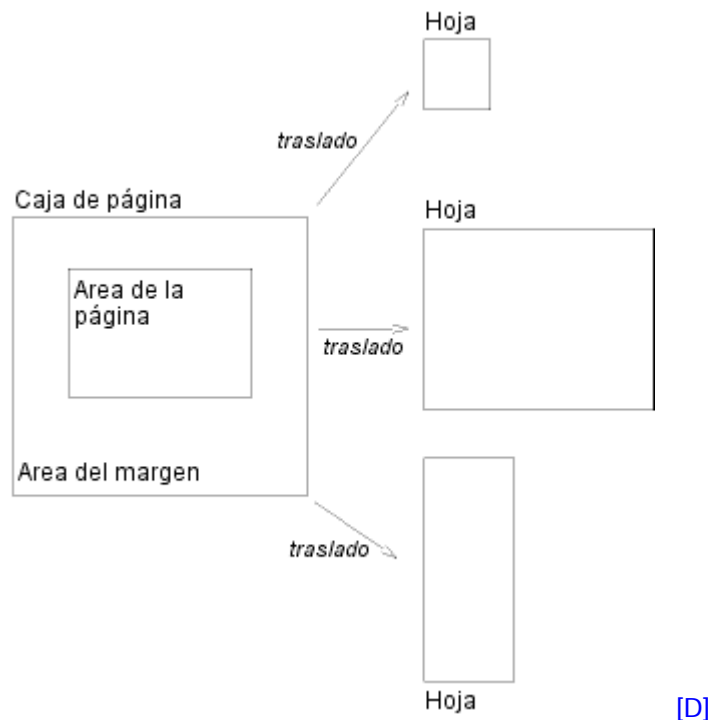
Por ejemplo, la siguiente regla @page fija el tamaño de la caja de página en 8.5 x 11 pulgadas y crea un margen de '2cm' en cada lado entre el borde de la caja de página y el área de la página:

```
@page { size 8.5in 11in; margin: 2cm }
```

La propiedad 'marks' en una regla @page especifica las marcas de corte y de registro para la caja de página.

### 13.2.1 Márgenes de la página

Las [propiedades del margen](#) ('margin-top', 'margin-right', 'margin-bottom', 'margin-left' y 'margin') se aplican en el [contexto de página](#). El siguiente diagrama ilustra la relación entre la hoja, la caja de página y los márgenes de la página:



El valor computado de los márgenes de la caja encima y abajo del área de la página es '0'.

El [contexto de página](#) no posee ninguna noción de fuentes, de modo que las unidades 'em' y 'ex' no están permitidas. Los valores de porcentajes en las propiedades del margen son relativas a las dimensiones de la [caja de página](#); para los márgenes izquierdo y derecho, se refieren al ancho de la caja de página mientras que para los márgenes superior e inferior, se refieren a la altura de la caja de página. Todas las demás unidades asociadas con las respectivas propiedades de CSS2 están permitidas.

Como consecuencia de valores negativos para el margen (ya sea en la caja de página o en los elementos) o del [posicionamiento absoluto](#) el contenido puede terminar fuera de la caja de página, pero este contenido puede ser "cortado" --por la aplicación del usuario, la impresora, o finalmente, por la guillotina--.

### 13.2.2 Tamaño de la página: la propiedad 'size'

#### 'size'

- Valor: <medida>{1,2} | auto | portrait | landscape | inherit
- Inicial: auto
- Se aplica a: el contexto de página
- Se hereda: N/A
- Porcentajes: N/A
- Medios: visuales, paginados

Esta propiedad especifica el tamaño y orientación de una caja de página.

El tamaño de una caja de página puede ser "absoluto" (tamaño fijo) o "relativo" (escalable, es decir, ajustado a los tamaños de hoja disponibles). Las cajas de página relativas permiten a las aplicaciones del usuario escalar un documento y hacer un uso óptimo del tamaño final.

Tres valores de la propiedad 'size' crean una caja de página relativa:

#### auto

La caja de página será puesta al tamaño y orientación de la hoja de destino.

#### landscape

Sustituye la orientación del destino. La caja de página es del mismo tamaño que el destino, y los lados más largos son horizontales.

#### portrait

Sustituye la orientación del destino. La caja de página es del mismo tamaño que el destino, y los lados más cortos son horizontales.

En el ejemplo siguiente, los bordes exteriores de una caja de página se alinearán con el destino. El valor de porcentaje en la propiedad 'margin' es relativo al tamaño del destino por lo que si las dimensiones de la hoja de destino son 21.0cm x 29.7cm (es decir, A4), los márgenes son 2.10cm y 2.97cm.

```
@page {
  size: auto;    /* auto es el valor inicial */
  margin: 10%;
}
```

Los valores de medida para la propiedad 'size' crean una caja de página absoluta. Si sólo se especifica un valor de medida, determina tanto el ancho como el alto de la caja de página (es decir, la caja es un cuadrado). Como la

caja de página es el [bloque de contención inicial](#), los valores de porcentaje no están permitidos para la propiedad 'size'.

Por ejemplo:

```
@page {
  size: 8.5in 11in; /* ancho altura */
}
```

El ejemplo de arriba determina que el ancho de la caja de página sea de 8.5 pulgadas y la altura de 11 pulgadas. La caja de página en este ejemplo requiere de un tamaño de hoja de destino de 8.5"x11" o mayor.

Las aplicaciones del usuario pueden permitir a los usuarios controlar el traslado de la caja de página a la hoja (por ej., rotando una caja de página absoluta que está siendo impresa).

Procesamiento de cajas de página que no se ajustan a una hoja de destino

Si una caja de página no se ajusta a las dimensiones de la hoja de destino, la aplicación del usuario puede decidir:

- Rotar 90° la caja de página si esto hace que la caja de página se ajuste.
- Escalar la página para ajustarla al destino.

La aplicación del usuario debe consultar al usuario antes de efectuar estas operaciones.

Ubicación de la caja de página en la hoja

Cuando la caja de página es menor que el tamaño del destino, la aplicación del usuario es libre de ubicar la caja de página en cualquier lugar de la hoja. Sin embargo, se recomienda que la caja de página sea centrada en la hoja porque así se alinearán las páginas de doble faz y se evitará la pérdida accidental de información que es impresa cerca del borde de la hoja.

13.2.3 Marcas de corte: la propiedad 'marks'

|              |                                    |
|--------------|------------------------------------|
| 'marks'      |                                    |
| Valor:       | [ crop    cross ]   none   inherit |
| Inicial:     | none                               |
| Se aplica a: | el contexto de página              |
| Se hereda:   | N/A                                |
| Porcentajes: | N/A                                |
| Medios:      | visuales, paginados                |

En las impresoras de alta calidad, a menudo se agregan marcas fuera de la caja de página. Esta propiedad especifica si se deben procesar las marcas de corte o marcas de registro o ambas justo fuera de los bordes de la [caja de página](#).

Las *crop marks* (marcas de corte) indican dónde debe cortarse la página. Las *cross marks* (marcas de registro, también llamadas marcas de registración) se usan para alinear las hojas.

Las marcas son visibles sólo en las cajas de página absolutas (ver la propiedad 'size'). En las cajas de página relativas, la caja de página será alineada con el destino y las marcas quedarán fuera del área de impresión.

El tamaño, estilo y posición de las marcas de registro depende de la aplicación del usuario.

13.2.4 Páginas izquierda, derecha y primera

Cuando se imprimen documento en doble faz, las [cajas de página](#) en páginas izquierdas o derechas deberán ser diferentes. Esto puede expresarse por medio de dos pseudo-clases de CSS que pueden definirse en el [contexto de página](#).

Todas las páginas son automáticamente clasificadas por las aplicaciones del usuario en la pseudo-clase :left o :right.

```
@page :left {
  margin-left: 4cm;
  margin-right: 3cm;
}

@page :right {
  margin-left: 3cm;
  margin-right: 4cm;
}
```



Si se hacen declaraciones diferenciadas para las páginas izquierda y derecha, la aplicación del usuario debe respetar estas declaraciones aún cuando la aplicación del usuario no traslade las cajas de página a hojas izquierda y derecha (ej., una impresora que sólo imprime una cara).

Los autores también pueden especificar un estilo para la primer página de un documento con la pseudo-clase `:first`:

```
@page { margin: 2cm } /* Todos los márgenes fijados en 2cm */

@page :first {
  margin-top: 10cm    /* Margen superior de la primer página de 10cm */
}
```

Si la primer página de un documento es `:left` o `:right` depende de la dirección de escritura principal del documento y está fuera del alcance de este documento. Sin embargo, para forzar una primer página `:left` o `:right`, los autores pueden insertar un salto de página antes de la primer caja generada (por ej., en HTML, especificando esto para el elemento BODY).

Las propiedades especificadas en una regla `@page :left` (o `:right`) reemplaza las especificadas en una regla `@page` que no tiene especificada una pseudo-clase. Las propiedades especificadas en una regla `@page :first` reemplaza las especificadas en las reglas `@page :left` (o `:right`).

**Nota.** Agregar declaraciones a las pseudo-clases `:left` o `:right` no influye si el documento sale impreso en una o en ambas caras (lo que está fuera del alcance de esta especificación).

**Nota.** Las futuras versiones de CSS pueden incluir otras pseudo-clases de página.

### 13.2.5 Contenido fuera de la caja de página

Al estructura el contenido en un modelo de página, parte del contenido puede acabar fuera de la caja de página. Por ejemplo, un elemento cuya propiedad `'white-space'` tiene el valor `'pre'` puede generar una caja que sea más ancha que la caja de página. También, cuando las cajas son posicionadas [absolutamente](#), pueden finalizar en ubicaciones "inconvenientes". Por ejemplo, las imágenes puede ubicarse sobre el borde de la caja de página o 100.000 pulgadas abajo de la caja de página.

Una especificación del formato exacto de tales elementos queda fuera del alcance de este documento. Sin embargo, recomendamos que los autores y las aplicaciones del usuario observen los siguientes principios generales concernientes al contenido fuera de la caja de página:

- El contenido debe ser admitido ligeramente fuera de la caja de página para permitir el "sangrado" de las páginas.
- Las aplicaciones del usuario deben evitar generar un gran número de cajas de página vacías para respetar la ubicación de los elementos (por ej., nadie quiere imprimir 100 páginas en blanco). Observe, sin embargo, que generar un pequeño número de cajas de página vacías puede ser necesario para respetar los valores `'left'` y `'right'` para `'page-break-before'` y `'page-break-after'`.
- Los autores no deben posicionar los elementos en ubicaciones inconvenientes sólo para evitar procesarlos. En cambio:
  - Para eliminar completamente la generación de la caja, poner la propiedad `'display'` como `'none'`.
  - Para hacer invisible una caja, usar la propiedad `'visibility'`.
- Las aplicaciones del usuario pueden manipular las cajas ubicadas fuera de la caja de página de varias maneras, incluyendo el descarte de las mismas o la creación de cajas de página para ellas al final del documento.

## 13.3 Saltos de página

Las siguientes secciones explican el formato de página en CSS2. Cinco propiedades indican dónde debe o puede la aplicación del usuario producir los saltos de página, y en qué página (izquierda o derecha) debe continuar el contenido subsecuente. Cada salto de página finaliza la composición de la [caja de página](#) actual y provoca que las partes que restan de la [estructura del documento](#) sean colocadas en una nueva caja de página.

### 13.3.1 Saltos antes/después de los elementos: 'page-break-before', 'page-break-after', 'page-break-inside'

#### 'page-break-before'

|              |  |
|--------------|--|
| Valor:       | auto   always   avoid   left   right   inherit |
| Inicial:     | auto   |
| Se aplica a: | los elementos a nivel de bloque                |
| Se hereda:   | no   |
| Porcentajes: | N/A  |
| Media:       | visual, paged                                  |

'page-break-after'

Valor: auto | always | avoid | left | right | inherit  
Inicial: auto  
Se aplica a: los elementos a nivel de bloque  
Se hereda: no  
Porcentajes: N/A  
Medios: visuales, paginados

'page-break-inside'

Valor: avoid | auto | inherit  
Inicial: auto  
Se aplica a: los elementos a nivel de bloque  
Se hereda: si  
Porcentajes: N/A  
Medios: visuales, paginados

Los valores para estas propiedades tienen los siguientes significados:

**auto**  
Ni fuerza ni prohíbe un salto de página antes de (después de, en) la caja generada.

**always**  
Siempre fuerza un salto de página antes (después) de la caja generada.

**avoid**  
Evita un salto de página antes de (después de, en) la caja generada.

**left**  
Fuerza uno o dos saltos de página antes (después) de la caja generada para que la siguiente página sea compuesta como una página izquierda.

**right**  
Fuerza uno o dos saltos de página antes (después) de la caja generada para que la siguiente página sea compuesta como una página derecha.

Una ubicación potencial de un salto de línea está típicamente bajo la influencia de la propiedad 'page-break-inside' del elemento padre, la propiedad 'page-break-after' del elemento precedente y la propiedad 'page-break-before' del elemento siguiente. Cuando estas propiedades tienen valores distintos que 'auto', los valores 'always', 'left' y 'right' tienen prioridad sobre 'avoid'. Ver la sección sobre [saltos de página permitidos](#) por las reglas exactas sobre cómo estas propiedades pueden forzar o suprimir un salto de página.

13.3.2 Uso de páginas nominadas: 'page'

'page'

Valor: <identificador> | auto  
Inicial: auto  
Se aplica a: los elementos a nivel de bloque  
Se hereda: si  
Porcentajes: N/A  
Medios: visuales, paginados

La propiedad 'page' puede utilizarse para especificar un tipo particular de página donde un elemento debe ser visualizado.

Este ejemplo pondrá a todas las tablas en páginas de la derecha apaisadas (llamadas "rotadas"):

```
@page rotadas {size: landscape}  
TABLE {page: rotadas; page-break-before: right}
```

La propiedad 'page' trabaja del siguiente modo: Si una caja de bloque con contenido a nivel de línea tiene una propiedad 'page' distinta a la caja de bloque con contenido a nivel de línea precedente, entonces se insertan uno o dos saltos de página entre ellas, y las cajas después del salto son procesadas en una caja de página del tipo nombrado. Ver ["Saltos de página forzados" más abajo](#).

En este ejemplo, las dos tablas son procesadas en páginas apaisadas (de hecho, en la misma página, si caben) y el tipo de página "angosta" no se usa en absoluto, a pesar de haber sido puestas en el DIV:

```
@page angosta {size: 9cm 18cm}  
@page rotada {size: landscape}  
DIV {page: angosta}  
TABLE {page: rotada}
```

con este documento:

```
<DIV>
<TABLE> . . . </TABLE>
<TABLE> . . . </TABLE>
</DIV>
```

### 13.3.3 Saltos dentro de los elementos: 'orphans', 'widows'

**'orphans'**

Valor: <entero> | inherit  
Inicial: 2  
Se aplica a: los elementos a nivel de bloque  
Se hereda: si  
Porcentajes: N/A  
Medios: visuales, paginados

**'widows'**

Valor: <entero> | inherit  
Inicial: 2  
Se aplica a: los elementos a nivel de bloque  
Se hereda: si  
Porcentajes: N/A  
Medios: visuales, paginados

La propiedad 'orphans' especifica el número mínimo de líneas de un párrafo que deben dejarse al final de una página. La propiedad 'widows' especifica el número mínimo de líneas de un párrafo que deben dejarse al comienzo de una página. Los ejemplos de cómo se usan para controlar los saltos de página se dan abajo.

**Nota de la traducción:**  
"Orphans" y "widows" significan literalmente "huérfanas" y "viudas" (en femenino porque se refieren a las líneas de un párrafo) y en castellano estas mismas palabras se utilizan en igual sentido.

Por información acerca del formato de los párrafos, por favor consulte la sección sobre [cajas de línea](#).

### 13.3.4 Saltos de página permitidos

En el flujo normal, los saltos de página pueden producirse en los siguientes lugares:

1. En el margen vertical entre cajas de bloque. Cuando un salto de página se produce aquí, los [valores computados](#) de las propiedades 'margin-top' y 'margin-bottom' son fijados en '0'.
2. Entre [cajas de línea](#) dentro de una caja de [bloque](#).

Estos saltos están sujetos a las siguientes reglas:

- **Regla A:** Provocar un salto en (1) está permitido sólo si las propiedades 'page-break-after' y 'page-break-before' de todas las cajas generadas de los elementos que se juntan en este margen lo permiten, lo cuál sucede cuando al menos uno de ellos tiene el valor 'always', 'left' o 'right', o cuando todos son auto'.
- **Regla B:** Sin embargo, si todos son 'auto' y el antepasado común más cercano de todos los elementos tiene un valor para 'page-break-inside' de 'avoid', entonces el salto aquí no está permitido.
- **Regla C:** Provocar un salto en (2) está permitido sólo si el número de [cajas de línea](#) entre el salto y el comienzo de la caja de bloque que las encierra es el valor 'orphans' o más, y el número de cajas de línea entre el salto y el final de la caja es el valor de 'widows' o más.
- **Regla D:** Además, provocar un salto en (2) está permitido sólo si la propiedad 'page-break-inside' es 'auto'.

Si lo anterior no proporciona suficientes puntos de ruptura para mantener al contenido fluyendo entre las cajas de página, entonces las reglas B y D se abandonan para encontrar puntos adicionales para los saltos.

Si eso aún no conduce a una cantidad de puntos de ruptura suficientes, las reglas A y B son también abandonadas, para encontrar todavía más puntos para los saltos.

Los saltos de página no pueden producirse dentro de las cajas que están [posicionadas absolutamente](#).

### 13.3.5 Saltos de página forzados

Un salto de página *debe* producirse en (1) if, entre las propiedades 'page-break-after' y 'page-break-before' de todas las cajas generadas de los elementos que se juntan en este margen, hay al menos una con el valor de 'always', 'left' o 'right'.

Un salto de página debe producirse en (1) si la última caja de línea encima de este margen y la primera debajo del mismo no tienen el mismo valor para 'page'.

### 13.3.6 Los "mejores" saltos de página

CSS2 *no* define cuál de todos el conjunto de saltos de página permitidos debe usarse; CSS2 no prohíbe que la aplicación del usuario realice el salto en cada posible punto de corte, o que no lo haga en absoluto. Pero CSS2 recomienda que las aplicaciones del usuario observen las siguientes reglas heurísticas (en tanto reconoce que estas son a veces contradictorias):

- Producir los menores cortes posibles.
- Hacer que todas las páginas que no terminan con cortes forzados parezcan tener aproximadamente la misma altura.
- Evitar el corte dentro un bloque que tiene un borde.
- Evitar el corte dentro de una tabla.
- Evitar el corte dentro de un elemento flotante.

Suponga, por ejemplo, que la hoja de estilo contiene 'orphans : 4', 'widows : 2', y hay 20 líneas (cajas de línea) disponibles al final de la página actual:

- Si un párrafo al final de la página actual contiene 20 líneas o menos, deber colocarse en la página actual.
- Si el párrafo contiene 21 o 22 líneas, la segunda parte del párrafo no debe violar la restricción 'widows', y entonces la segunda parte debe contener exactamente dos líneas.
- Si el párrafo contiene 23 líneas o más, la primer parte debe contener 20 líneas y la segunda parte las líneas restantes.

Ahora suponga que 'orphans' es '10', 'widows' es '20', y hay 8 líneas disponibles al final de la página actual:

- Si un párrafo al final de la página actual contiene 8 líneas o menos, debe colocarse en la página actual.
- Si el párrafo contiene 9 líneas o más, no puede ser cortado (esto violaría la restricción impuesta a las líneas huérfanas), así que debe moverse como un bloque a la página siguiente.

## 13.4 Cascada en el contexto de página

Las declaraciones en el contexto de página obedecen el orden de cascada tanto como las declaraciones normales de CSS2.

Considere el siguiente ejemplo:

```
@page {
  margin-left: 3cm;
}

@page :left {
  margin-left: 4cm;
}
```

Debido a la mayor especificidad del selector de pseudo-clase, el margen izquierdo en las páginas de la izquierda será de '4cm' y todas las demás páginas (es decir, las páginas de la derecha) tendrán un margen izquierdo de '3cm'.

# 14 Colores y fondos

Las propiedades de CSS permiten a los autores especificar el color de fondo y el aspecto del fondo de un elemento. Los fondos pueden ser colores o imágenes. Las propiedades del fondo permiten a los autores ubicar una imagen de fondo, repetirla y declarar si debe quedar fija con respecto al [acceso visual](#) o desplazada junto con el documento.

Ver la sección sobre [unidades de color](#) por la sintaxis de los valores de color válidos.

## 14.1 Color del primer plano: la propiedad 'color'

### 'color'

|              |                                      |
|--------------|--------------------------------------|
| Valor:       | <color>   inherit                    |
| Inicial:     | depende de la aplicación del usuario |
| Se aplica a: | todos los elementos                  |
| Se hereda:   | si                                   |
| Porcentajes: | N/A                                  |
| Medios:      | visuales                             |

Esta propiedad describe el color del primer plano del contenido de texto de un elemento. Existen distintas maneras de especificar rojo:

```
EM { color: red }           /* nombre de color predefinido */
EM { color: rgb(255,0,0) }  /* RGB rango 0-255 */
```

## 14.2 El fondo

Los autores pueden especificar el fondo de un elemento (es decir, su superficie de procesamiento) ya sea como un color o como una imagen. En términos del [modelo de caja](#), "background" se refiere al fondo del [contenido](#) y las áreas de [relleno](#) (padding). El color y estilo del borde se ponen con las [propiedades del borde](#). Los márgenes son siempre transparentes de modo que el fondo de la caja padre siempre se ve a través.

Las propiedades del fondo no se heredan, pero el fondo de la caja padre por defecto se verá a través debido al valor inicial 'transparent' para 'background-color'.

El fondo de una caja generada por el elemento raíz cubre todo el [lienzo](#).

Para los documentos en HTML, sin embargo, recomendamos que los autores especifiquen el fondo del elemento BODY en vez del elemento HTML. Las aplicaciones del usuario deben observar las siguientes reglas de precedencia para llenar el fondo: si el valor de la propiedad 'background' del elemento HTML es distinto a 'transparent' entonces se lo usa, de otro modo se usa el valor de la propiedad 'background' del elemento BODY. Si el valor resultante es 'transparent', el procesamiento es indefinido.

De acuerdo con estas reglas, el lienzo subyacente del siguiente documento HTML tendrá un fondo "mármol".

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Poniendo el fondo del lienzo</TITLE>
    <STYLE type="text/css">
      BODY { background: url("http://style.com/mármol.png") }
    </STYLE>
  </HEAD>
  <BODY>
    <P>Mi fondo es de mármol.
  </BODY>
</HTML>
```

### 14.2.1 Propiedades del fondo: 'background-color', 'background-image', 'background-repeat', 'background-attachment', 'background-position' y 'background'

#### 'background-color'

|              |                                 |
|--------------|---------------------------------|
| Valor:       | <color>   transparent   inherit |
| Inicial:     | transparent                     |
| Se aplica a: | todos los elementos             |
| Se hereda:   | no                              |
| Porcentajes: | N/A                             |
| Medios:      | visuales                        |

Esta propiedad determina del color de fondo de un elemento, ya sea un valor de <color> o la palabra clave 'transparent', para hacer que los colores subyacentes se vean a través.

```
H1 { background-color: #F00 }
```

'background-image'

- Valor: <uri> | none | inherit
- Inicial: none
- Se aplica a: todos los elementos
- Se hereda: no
- Porcentajes: N/A
- Medios: visuales

Esta propiedad determina la imagen de fondo de un elemento. Cuando ponen una imagen de fondo, los autores deben también especificar un color de fondo que será usado cuando la imagen no esté disponible. Cuando la imagen está disponible, es procesada encima del color de fondo. (Así, el color es visible a través de las zonas transparentes de la imagen.)

Los valores para esta propiedad son un <uri>, para especificar la imagen, o 'none', cuando no se utiliza ninguna imagen.

```
BODY { background-image: url("mármol.gif") }
P { background-image: none }
```

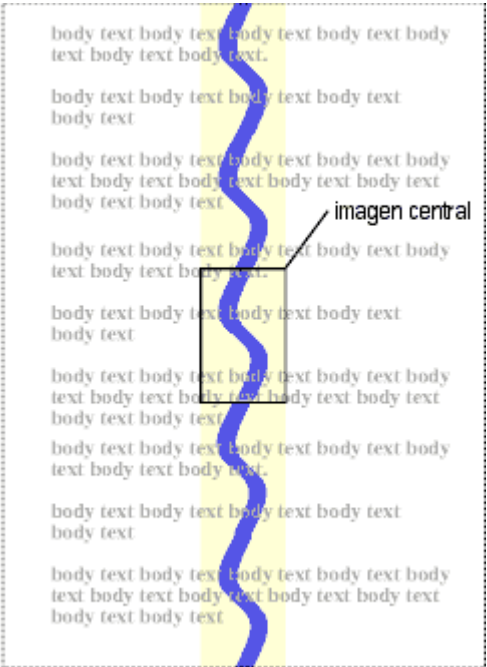
'background-repeat'

- Valor: repeat | repeat-x | repeat-y | no-repeat | inherit
- Inicial: repeat
- Se aplica a: todos los elementos
- Se hereda: no
- Porcentajes: N/A
- Medios: visuales

Si se especifica una imagen de fondo, esta propiedad especifica si la imagen es repetida (mosaico), y de qué modo. Un mosaico completo cubre las áreas de [contenido](#) y [relleno](#) de una caja. Los valores tienen los siguientes significados:

- repeat**  
La imagen es repetida tanto horizontal como verticalmente.
- repeat-x**  
La imagen es repetida sólo horizontalmente.
- repeat-y**  
La imagen es repetida sólo verticalmente.
- no-repeat**  
La imagen no se repite: sólo una copia de la imagen es visualizada.

```
BODY {
  background: white url("pendiente.gif");
  background-repeat: repeat-y;
  background-position: center;
}
```



[D]

Una copia de la imagen está centrada, y otras copias están colocadas por encima y por debajo de la misma para formar una banda vertical por detrás del elemento.

'background-attachment'

Valor: scroll | fixed | inherit  
Inicial: scroll  
Se aplica a: todos los elementos  
Se hereda: no  
Porcentajes: N/A  
Medios: visuales

Si se especifica una imagen de fondo, esta propiedad especifica si es fija con respecto al [acceso visual](#) ('fixed') o se desplaza junto con el documento ('scroll').

Incluso cuando la imagen sea fija, sólo será visible cuando esté en el fondo o el área de relleno del elemento. De este modo, a menos que la imagen forme un mosaico ('background-repeat: repeat'), puede ser invisible.

Este ejemplo crea una banda vertical infinita que permanece "pegada" al acceso visual cuando el elemento es desplazado.

```
BODY {  
  background: red url("pendiente.gif");  
  background-repeat: repeat-y;  
  background-attachment: fixed;  
}
```

Las aplicaciones del usuario pueden tratar 'fixed' como 'scroll'. Sin embargo, se recomienda que interpreten 'fixed' correctamente, la menos para los elementos HTML y BODY, porque no hay un modo de que el autor proporcione una imagen exclusivamente para aquellos navegadores que soportan 'fixed'. Ver la sección sobre [conformidad](#) para más detalles.

'background-position'

Valor: [ [<porcentaje> | <medida> ]{1,2} || [ [top | center | bottom] || [left | center | right] ] ] | inherit  
Inicial: 0% 0%  
Se aplica a: los elementos reemplazados y a nivel de bloque  
Se hereda: no  
Porcentajes: referidos al tamaño de la propia caja  
Medios: visuales

Si se ha especificado una imagen de fondo, esta propiedad especifica su posición inicial. Los valores tienen los siguientes significados:

- <porcentaje> <porcentaje>  
Con un par de valores de '0% 0%', la esquina superior izquierda de la imagen es alineada con la esquina superior izquierda del [límite del relleno](#) de la caja. Un par de valores de '100% 100%' coloca la esquina inferior derecha de la imagen en la esquina inferior derecha del área del relleno. Con un par de valores de '14% 84%', el punto ubicado a un 14% del costado y a un 84% hacia abajo en la imagen será colocado en el punto ubicado a un 14% del costado y a un 84% hacia abajo en el área de relleno.
- <medida> <medida>  
Con un par de valores de '2cm 2cm', la esquina superior izquierda de la imagen es ubicada 2cm a la derecha y 2cm abajo de la esquina superior izquierda del área de relleno.
- top left y left top  
Igual '0% 0%'.
- top, top center y center top  
Igual a '50% 0%'.
- right top y top right  
Igual a '100% 0%'.
- left, left center y center left  
Igual a '0% 50%'.
- center y center center  
Igual a '50% 50%'.
- right, right center y center right  
Igual a '100% 50%'.
- bottom left y left bottom  
Igual a '0% 100%'.
- bottom, bottom center y center bottom  
Igual a '50% 100%'.
- bottom right y right bottom  
Igual a '100% 100%'.



Si solamente se da un valor de porcentaje o de medida, éste determina sólo la posición horizontal, la posición vertical será 50%. Si se dan dos valores, la posición horizontal viene primero. La combinación de valores de medida y de porcentajes está permitida (ej., '50% 2cm'). Las posiciones negativas están permitidas. Las palabras clave no pueden ser combinadas con valores de porcentaje o de medida (todas las combinaciones posibles están dadas arriba).

```
BODY { background: url("banner.jpeg") right top } /* 100% 0% */
BODY { background: url("banner.jpeg") top center } /* 50% 0% */
BODY { background: url("banner.jpeg") center } /* 50% 50% */
BODY { background: url("banner.jpeg") bottom } /* 50% 100% */
```

Si la imagen de fondo está fija en el acceso visual (ver la propiedad 'background-attachment'), la imagen es ubicada en relación al acceso visual en vez del área de relleno del elemento. Por ejemplo,

```
BODY {
    background-image: url("logo.png");
    background-attachment: fixed;
    background-position: 100% 100%;
    background-repeat: no-repeat;
}
```

En el ejemplo de arriba, la (única) imagen es ubicada en la esquina inferior derecha del acceso visual.

**'background'**

- Valor: [<'background-color'> || <'background-image'> || <'background-repeat'> || <'background-attachment'> || <'background-position'>] | inherit
- Inicial: no definido para las propiedades resumidas
- Se aplica a: todos los elementos
- Se hereda: no
- Porcentajes: permitidos en 'background-position'
- Medios: visuales

La propiedad 'background' es una propiedad resumida para fijar la propiedades individuales del fondo (es decir, 'background-color', 'background-image', 'background-repeat', 'background-attachment' y 'background-position') en un mismo lugar en la hoja de estilo.

La propiedad 'background' primero coloca a todas las propiedades individuales del fondo en sus valores iniciales, luego asigna los valores explícitamente dados en la declaración.

En la primer regla del ejemplo siguiente, sólo un valor para 'background-color' ha sido fijado y las otras propiedades individuales son puestas con sus valores iniciales. En la segunda regla, todas las propiedades individuales han sido especificadas.

```
BODY { background: red }
P { background: url("chess.png") gray 50% repeat fixed }
```

### 14.3 Corrección de la gama

Para información acerca del tema de la gama, por favor consulte la Guía de gama en la especificación PNG ([PNG10](#)).

En el cómputo de la corrección de gama, las AU visualizadas en un CRT pueden asumir un CRT ideal e ignorar cualquier efecto de gama aparente provocado por el dithering. Esto significa que el tratamiento mínimo que necesita hacer en la plataforma actual es:

- PC usando MS-Windows
  - ninguno
- Unix usando X11
  - ninguno
- Mac usando QuickDraw
  - aplicar una gama de 1.45 [\[ICC32\]](#) (Las aplicaciones ColorSync-savvy pueden simplemente pasar el perfil sRGB ICC a ColorSync para llevar a cabo la adecuada corrección de color)
- SGI usando X
  - aplicar el valor de gama de /etc/config/system.glGammaVal (siendo el valor predeterminado de 1.70; las aplicaciones corriendo en Irix 6.2 o superior pueden simplemente pasar el perfil sRGB ICC al sistema de administración de color)
- NeXT usando NeXTStep
  - aplicar una gama de 2.22

"Aplicar una gama" significa que cada uno de los tres R, G y B deben ser convertidos a  $R'=R^{gama}$ ,  $G'=G^{gama}$ ,  $B'=B^{gama}$ , antes de ser entregado al SO.

Esto puede efectuarse rápidamente construyendo una paleta de 256 elementos una vez por cada llamada al navegador, así:

```
for i := 0 to 255 do
  raw := i / 255.0;
  corr := pow (raw, gamma);
  table[i] := trunc (0.5 + corr * 255.0)
end
```

lo que entonces evita cualquier necesidad de efectuar cálculos complicados por atributo de color, mucho menos por pixel.

# 15 Fuentes

## 15.1 Introducción

Cuando el texto de un documento va a ser desplegado visualmente, a los caracteres (elementos abstractos de información) se le deben asignar *signos abstractos*. Uno o más caracteres pueden ser representados por uno o más signos abstractos, de un modo, tal vez, dependiente del contexto. Una *signo* es la representación artística concreta de un signo abstracto, en cierto estilo tipográfico, en forma de contorno o mapa de bits que puede dibujarse en la pantalla o en papel. Una *fuentes* es un conjunto de signos, observando todos ellos el mismo motivo básico de acuerdo al diseño, tamaño, apariencia y otros atributos asociados a todo el conjunto, y una asignación de signos abstractos por cada carácter.

Una aplicación del usuario visual debe resolver las siguientes cuestiones antes de procesar visualmente un carácter:

- ¿Está especificada, directamente o por herencia, una fuente para este carácter?
- ¿La aplicación del usuario tiene disponible esta fuente?
- Si es así, ¿qué signo(s) le será asignado a este carácter o secuencia de caracteres?
- Si no, ¿qué se deberá hacer? ¿Deberá ser reemplazada por otra fuente? ¿Puede ser sintetizada la fuente? ¿Puede recuperarse de la Web?

En ambos, CSS1 y CSS2, los autores especifican las características de una fuente a través de una serie de propiedades de la fuente.

El modo en que las aplicaciones del usuario operan estas propiedades, cuando en el sistema cliente no existe una fuente equivalente, se ha ampliado entre CSS1 y CSS2. En CSS1, se asumía que todas las fuentes estaban presentes en el sistema cliente y eran identificadas solamente por su nombre. Podían especificarse fuentes alternativas por medio de las propiedades, pero más allá de eso, las aplicaciones del usuario no tenían ninguna manera de proponer otra fuente al usuario (incluso fuentes estilísticamente similares que la aplicación del usuario tenía disponible) más allá de las fuentes genéricas predefinidas.

CSS2 cambia todo esto y concede mucha más libertad a:

- los autores de hojas de estilo para describir las fuentes que desean que se usen.
- las aplicaciones del usuario para seleccionar una fuente cuando la requerida por el autor no está disponible en ese momento.

CSS2 mejora la equivalencia de fuentes en el sistema cliente, permite el sintetizado de fuentes y el procesamiento progresivo, y posibilita que las fuentes sean bajadas de la Web. Estas capacidades mejoradas son aludidas como 'Fuentes Web'.

En el modelo de fuente CSS2, como en CSS1, cada aplicación del usuario tiene una "base de datos de fuentes" a su disposición. CSS1 se refería a esta base de datos pero no daba detalle de lo que había en ella. CSS2 define la información de esa base de datos y permite a los autores de hojas de estilo aportar a la misma. Cuando se le solicita mostrar un carácter con una fuente particular, la aplicación del usuario primero identifica en la base de datos la fuente que "más se asemeja" con la fuente especificada (de acuerdo con el [algoritmo de equivalencia de fuentes](#)). Una vez que ha identificado una fuente, recupera los datos de la fuente localmente o desde la Web y puede mostrar el carácter usando esos signos.

Como consecuencia de este modelo, hemos organizado la especificación en dos secciones. La primera concierne a los [mecanismos de especificación de fuentes](#), por los cuales los autores especifican qué fuentes desearían que fueran usadas. La segunda es respecto al [mecanismo de selección de fuentes](#), por cuyo medio las aplicaciones del usuario del sistema cliente identifican y cargan la fuente que mejor equivale con la especificación del autor.

El modo en que las aplicaciones del usuario construyen la base de datos está fuera del alcance de esta especificación toda vez que la implementación de bases de datos depende de factores tales como el sistema operativo o el tipo de cliente.

## 15.2 Especificación de la fuente

La primer etapa del mecanismo de fuente de CSS tiene que ver con el modo en que los autores de hojas de estilo especifican cuáles fuentes deben ser usadas por una aplicación del usuario. En principio, parece que la manera obvia de especificar una fuente es por su nombre, una cadena simple -que da la impresión de estar separada en distintas partes-, por ejemplo "BT Swiss 721 Heavy Italic".

Desafortunadamente, no existe una taxonomía bien definida y universalmente aceptada para la clasificación de las fuentes en base a sus nombres, y los términos aplicados al nombre de una familia de fuentes pueden no ser apropiados para otras. Por ejemplo, el término 'itálica' es usado comúnmente para definir al texto inclinado, pero el texto inclinado también puede denominarse *oblicuo*, *sesgado*, *inclinado* o *cursivo*. De modo similar, los nombre de las fuentes típicamente contienen términos que describen el "peso" de una fuente. El principal rol de estos nombres es distinguir versiones de distinta densidad dentro de una misma familia de fuentes. No hay expresiones universalmente aceptadas para los distintos pesos y su utilización varía enormemente. Por ejemplo, una fuente que podría considerarse como negrita puede ser descripta como *regular*, *media*, *semi-negra* o *ultra-negra*, dependiendo de cuán negra sea la versión "normal" de la fuente dentro del diseño.

Esta ausencia de una denominación sistemática imposibilita, dentro del marco general, componer el nombre de una tipografía modificada que difiera en un sentido particular, como ser más negra.

Debido a esto, CSS usa un [modelo diferente](#). Las fuentes son solicitadas no a través del simple nombre de la fuente sino por la vía de establecer una serie de propiedades de la fuente. Estos valores de las propiedades forman la base del mecanismo de [selección de fuentes](#) de las aplicaciones del usuario. Las propiedades de la fuente pueden modificarse individualmente, por ejemplo para aumentar la densidad, y el nuevo conjunto de valores de las propiedades de la fuente será usado entonces para seleccionar nuevamente de la base de datos de las fuentes. El resultado es un incremento en la regularidad para los autores de hojas de estilo y los implementadores, y un aumento en la solidez.

### 15.2.1 Propiedades para la especificación de las fuentes

CSS2 especifica las fuentes de acuerdo a estas características:

**Font-family**

La familia de fuentes especifica qué familia será usada para procesar visualmente el texto. Una familia de fuentes es un grupo de fuentes diseñadas para ser usadas en combinación y exhibiendo similitudes en su diseño. Un miembro de la familia puede ser cursiva, otra negrita, otra condensada o versalitas. Los nombres de familias de fuentes incluyen a "Helvetica", "New Century Schoolbook" y "Kyokasho ICA L". Los nombres de las familias de fuentes no se restringen a caracteres latinos. Las familias de fuentes pueden ser agrupadas en diferentes categorías: aquellas con o sin serif, aquellas cuyos caracteres están o no espaciados proporcionalmente, las que imitan la escritura manual, aquellas que son tipos de fantasía, etc.

**Font-style**

El estilo de la fuente especifica si el texto será procesado visualmente usando un tipo en versión normal, itálica u oblicua. La *itálica* es la versión cursiva que acompaña al tipo normal, pero no tanto como para resultar una letra manuscrita. La oblicua es una versión inclinada del tipo normal y es usada más comúnmente en las sans-serif. Esta definición evita tener que denominar un tipo ligeramente inclinado como oblicuo o una tipografía griega normal como itálica.

**Font-variant**

La variante de la fuente indica si el texto debe ser procesado visualmente usando los signos normales para los caracteres en minúsculas o usando versalitas para las minúsculas. Una fuente particular puede contener sólo signos en minúsculas, sólo versalitas o ambos tipos de signos; esta propiedad se usa para solicitar la fuente apropiada y, si la fuente contiene ambas variantes, el signo apropiado.

**Font-weight**

El peso de la fuente se refiere al grado de solidez o de ligereza de los signos usados para procesar visualmente el texto, con relación a otras fuentes en la misma familia.

**Font-stretch**

El ancho de la fuente indica la cantidad deseada de condensación o expansión de los signos usados para procesar visualmente el texto, con relación a otras fuentes en la misma familia.

**Font-size**

El tamaño de la fuente se refiere al tamaño de la fuente desde una línea de base a otra cuando tiene asignada una interlinea normal (en términos CSS, esto es así cuando la propiedades 'font-size' y 'line-height' tienen el mismo valor).

En todas las propiedades, excepto 'font-size', los valores de las medidas 'em' y 'ex' se refieren al tamaño de fuente del elemento actual. Para 'font-size', estas unidades de medida se refieren al tamaño de fuente del elemento padre. Por favor, consulte la sección sobre [unidades de medida](#) para más información.

Las propiedades de las fuentes en CSS son usadas para describir la apariencia deseada para el texto del documento. Los descriptores de las fuentes, por el contrario, son usados para describir las características de las fuentes, de modo que pueda elegirse una fuente conveniente para crear la apariencia deseada. Por información acerca de la clasificación de las fuentes, por favor, consulte la sección sobre [descriptores de fuentes](#).

### 15.2.2 Familias de fuentes: la propiedad 'font-family'

**'font-family'**

- Value:* [[ <nombre-de-la-familia> | <familia-genérica> ],]\* [<nombre-de-la-familia> | <familia-genérica>] | inherit
- Inicial:* depende de la aplicación del usuario
- Se aplica a:* todos los elementos
- Se hereda:* si
- Porcentajes:* N/A
- Medios:* visuales

Esta propiedad especifica una lista de prioridades de nombres de familias de fuentes y/o nombres de familias genéricas. Para resolver el problema de que una sola fuente puede carecer de signos para mostrar todos los caracteres en un documento, o de que no todas las fuentes están disponibles en todos los sistemas, esta propiedad permite a los autores especificar un listado de fuentes, todas del mismo estilo y tamaño, que son probadas en sucesión para ver si contienen un signo para cierto carácter. Esta lista se llama un *juego de fuentes*.

Por ejemplo, un texto que contiene palabras en castellano mezcladas con símbolos matemáticos puede necesitar un juego de dos fuentes, una conteniendo letras y números latinos, la otra conteniendo los símbolos matemáticos. Aquí hay un ejemplo de un juego de fuentes conveniente para un texto que se espera que contenga caracteres latinos, en japonés y símbolos matemáticos:

```
BODY { font-family: Baskerville, "Heisi Mincho W3", Symbol, serif }
```

Los signos disponibles en la "Baskerville" (una fuente que cubre sólo los caracteres latinos) se tomarán de esa fuente, los signos japoneses serán tomados de la "Heisi Mincho W3" y los símbolos matemáticos vendrán de la "Symbol". Cualquier otra vendrá de la familia genérica 'serif'.

La familia genérica de fuentes se usará si una o más de las otras fuentes del juego de fuentes no está disponible. Aunque muchas fuentes proporcionan un signo para los "caracteres ausentes", típicamente un rectángulo blanco, como su nombre lo indica éste no debe ser considerada una equivalencia excepto en la última fuente de un juego de fuentes.

Hay dos tipos de nombres de familias de fuentes:

<nombre-de-la-familia>

El nombre de una familia de fuentes de preferencia. En el ejemplo anterior, "Baskerville", "Heisi Mincho W3" y "Symbol" son familias de fuentes. Los nombres de las familias que contienen [espacios en blanco](#) deben ir entre comillas. Si se omiten las comillas, cualquier [espacio en blanco](#) antes y después del nombre de la fuente son [ignorados](#) y cualquier secuencia de espacios en blanco dentro del nombre de la fuente se convierte en su espacio simple.

<familia-genérica>

Son definidas las siguientes familias genéricas: 'serif', 'sans-serif', 'cursive', 'fantasy' y 'monospace'. Por favor, vea la sección sobre [familias genéricas de fuentes](#) para las descripciones de estas familias. Los nombres de las familias genéricas de fuentes son palabras clave y, por consiguiente, no deben ir entre comillas.

Se exhorta a los autores a ofrecer una familia genérica de fuentes como última alternativa, para lograr mayor solidez.

Por ejemplo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Prueba de fuentes</TITLE>
    <STYLE type="text/css">
      BODY { font-family: "new century schoolbook", serif }
    </STYLE>
  </HEAD>
  <BODY>
    <H1 style="font-family: 'Mi propia fuente', fantasy">Prueba</H1>
    <P>¿Qué hay de nuevo, viejo?
  </BODY>
</HTML>
```

La enriquecida sintaxis del selector CSS2 puede usarse para crear una tipografía sensible al idioma. Por ejemplo, algunos caracteres chinos y japoneses son unificados para tener la misma posición en Unicode, aunque los signos abstractos no sean iguales en los dos idiomas.

```
*:lang(ja-jp) { font: 900 14pt/16pt "Heisei Mincho W9", serif }
*:lang(zh-tw) { font: 800 14pt/16.5pt "Li Sung", serif }
```

Esto selecciona cualquier elemento que está en el idioma dado -japonés o chino tradicional- y solicita la fuente apropiada.

### 15.2.3 Estilos de la fuente: las propiedades 'font-style', 'font-variant', 'font-weight' y 'font-stretch'

**'font-style'**

- Valor: normal | italic | oblique | inherit
- Inicial: normal
- Se aplica a: todos los elementos
- Se hereda: si
- Porcentajes: N/A
- Medios: visuales

La propiedad 'font-style' solicita la versión normal de una fuente (llamada también "romana" o "redonda") dentro de una familia. Los valores tienen los siguientes significados:

**normal**

Especifica una fuente clasificada como 'normal' en la base de datos de fuentes de la AU.

**oblique**

Especifica una fuente clasificada como 'oblique' en la base de datos de fuentes de la AU. Las fuentes que incluyan en sus nombres las expresiones *Oblique*, *Slanted* o *Incline* serán típicamente definidas como 'oblique'

en la base de datos de fuentes de la AU. Una fuente definida 'oblique' en la base de datos de fuentes de la AU puede haber sido generada, en realidad, inclinando electrónicamente una fuente normal.

**italic**

Especifica una fuente clasificada como 'italic' en la base de datos de fuentes de la AU o, si esa no está disponible, una definida como 'oblique'. Las fuentes que incluyan en sus nombres las expresiones *Italic*, *Cursive* o *Kursiv* serán típicamente definidas como 'italic'.

En este ejemplo, el texto normal en un elemento H1, H2 o H3 será mostrado con una fuente itálica. Sin embargo el texto enfatizado (EM) dentro de un elemento H1 aparecerá en una versión normal.

```
H1, H2, H3 { font-style: italic }
H1 EM { font-style: normal }
```

**'font-variant'**

Value: normal | small-caps | inherit  
Inicial: normal  
Se aplica a: todos los elementos  
Se hereda: si  
Porcentajes: N/A  
Medios: visuales

En una fuente small-caps, los signos para la letras minúsculas son iguales a las mayúsculas, pero en un tamaño menor y con proporciones ligeramente diferentes (*en castellano se denominan versales o versalitas*). La propiedad 'font-variant' solicita tales fuentes para los sistemas de "doble caja" (que tiene dos cajas, como la letra latina). Esta propiedad no tiene efectos visibles en escrituras de *caja única* (que contienen sólo una caja, como en la mayoría de los sistemas de escritura del mundo). Los valores tienen los siguientes significados:

**Nota de la traducción:**

Por sistemas de caja doble se entienden aquellos que tienen dos versiones -o más- de cada letra (en la escritura latina, mayúsculas y minúsculas). Los sistemas de caja única son aquellos que tienen una sola versión de letras.  
Los términos utilizados en inglés ("bicameral" y "unicameral") no significan demasiado en castellano, por lo que se prefirió utilizar la denominación "caja" ya que en la jerga tipográfica aún se estila denominar a las minúsculas como de "caja baja" y de "caja alta" a las mayúsculas (denominaciones derivadas del antiguo sistema de composición con tipos móviles).

**normal**

Especifica una fuente que no está definida como versalita.

**small-caps**

Especifica una fuente definida como versalita. Si no está disponible una versalita genuina, las aplicaciones del usuario deberán simularla, por ejemplo tomando una fuente normal y reemplazando las letras minúsculas por mayúsculas reducidas. Como último recurso, los signos de las letras mayúsculas sin reducir de una fuente normal pueden reemplazar los signos de una fuente versalita de modo que el texto aparezca todo en mayúsculas.

El resultado del siguiente ejemplo es un elemento H3 en versalitas con las palabras enfatizadas (EM) en versalitas oblicuas:

```
H3 { font-variant: small-caps }
EM { font-style: oblique }
```

En la medida en que esta propiedad provoca que el texto sea convertido a mayúsculas, se aplican las mismas consideraciones que para 'text-transform'.

**'font-weight'**

Valor: normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | inherit  
Inicial: normal  
Se aplica a: todos los elementos  
Se hereda: si  
Porcentajes: N/A  
Medios: visuales

La propiedad 'font-weight' especifica el peso de una fuente. Los valores tienen los siguientes significados:

**100 to 900**

Estos valores forman un secuencia ordenada, donde cada número indica un peso que es al menos tan fuerte como su predecesor.

**normal**

Igual a '400'.

**bold**

Igual a '700'.

**bolder**

Especifica el siguiente peso que es asignado a una fuente que es mayor que el valor heredado. Si no existe tal peso, el resultado es simplemente el siguiente valor numérico (y la fuente permanece sin cambios), a menos que el valor heredado sea '900', en cuyo caso el peso resultante será también '900'.

**lighter**

Especifica el siguiente peso que es asignado a una fuente es más liviano que el valor heredado. Si no existe tal peso, el resultado es simplemente el siguiente valor numérico más liviano (y la fuente permanece sin cambios), a menos que el valor heredado sea '100', en cuyo caso el peso resultante será también '100'.

```
P { font-weight: normal } /* 400 */
H1 { font-weight: 700 } /* negrita */
BODY { font-weight: 400 }
STRONG { font-weight: bolder } /* 500 si está disponible */
```

Los elementos hijos heredan el **valor computado** del peso.

**'font-stretch'**

|              |   |
|--------------|---|
| Value:       | normal   wider   narrower   ultra-condensed   extra-condensed   condensed   semi-condensed   semi-expanded   expanded   extra-expanded   ultra-expanded   inherit |
| Inicial:     | normal  |
| Se aplica a: | todos los elementos   |
| Se hereda:   | si  |
| Porcentajes: | N/A   |
| Medios:      | visuales  |

La propiedad 'font-stretch' selecciona un tipo normal, condensado o expandido de una familia de fuentes. Los valores de las palabras clave absolutas tienen el siguiente orden, del más estrecho al más ancho:

- 1. ultra-condensed
- 2. extra-condensed
- 3. condensed
- 4. semi-condensed
- 5. normal
- 6. semi-expanded
- 7. expanded
- 8. extra-expanded
- 9. ultra-expanded

La palabra clave relativa 'wider' modifica el valor al siguiente más expandido por sobre el valor heredado (mientras no lo aumente por sobre 'ultra-expanded'); la palabra clave relativa 'narrower' modifica el valor al siguiente más condensado por debajo del valor heredado (mientras no lo disminuya por debajo de 'ultra-condensed').

15.2.4 Tamaño de la fuente: las propiedades 'font-size' y 'font-size-adjust'

**'font-size'**

|              |   |
|--------------|---|
| Valor:       | <tamaño-absoluto>   <tamaño-relativo>   <medida>   <porcentaje>   inherit |
| Inicial:     | medium  |
| Se aplica a: | todos los elementos   |
| Se hereda:   | si, el valor computado es heredado  |
| Porcentajes: | se refieren al tamaño de la fuente del elemento padre                     |
| Medios:      | visuales  |

Esta propiedad describe el tamaño de la fuente con un interlineado normal. Los valores tienen los siguientes significados:

**<tamaño-absoluto>**

Una palabra clave <tamaño-absoluto> se refiere a una entrada en una tabla de tamaños de fuentes computada y guardada por la aplicación del usuario. Los posibles valores son:

```
[ xx-small | x-small | small | medium | large | x-large | xx-large ]
```

En una pantalla de computadora se sugiere un factor de 1.2 entre índices adyacentes; si la fuente 'medium' es de 12pt, la fuente 'large' debería ser de 14.4pt. Los diferentes medios pueden requerir distintos factores en su escala. Además, la aplicación del usuario debería tener en cuenta la calidad y disponibilidad de las fuentes cuando computa la tabla. La tabla puede diferir de una familia de fuentes a otra.

*Nota. En CSS1, el factor sugerido entre índices adyacentes era 1.5 que la experiencia de los usuarios demostró era demasiado grande.*

**<tamaño-relativo>**

Una palabra clave <tamaño-relativo> es interpretada como relativa a la tabla de tamaños de fuentes y al tamaño de fuente del elemento padre. Los valores posibles son:

```
[ larger | smaller ]
```



Por ejemplo, si el elemento padre tiene una fuente de tamaño 'medium', un valor 'larger' hará que el tamaño de la fuente del elemento actual sea 'large'. Si el tamaño del elemento padre no es cercano a una entrada de la tabla, la aplicación del usuario es libre de interpolarlo en las entradas de la tabla o redondearlo a la más cercana. La aplicación del usuario puede tener que extrapolar valores de la tabla si los valores numéricos sobrepasan las palabras clave.

**<medida>**

Un valor de medida especifica un tamaño absoluto para la fuente (es decir independiente de la tabla de fuentes de la aplicación del usuario). Las medidas negativas son ilícitas.

**<porcentaje>**

Un valor porcentual especifica un tamaño absoluto de la fuente relativo al tamaño de fuente del elemento padre. El uso de porcentajes o de valores 'em' conduce a hojas de estilo más sólidas y con mejor aprovechamiento del concepto de cascada.

El **valor real** de esta propiedad puede diferir del **valor computado** debido a un valor numérico en 'font-size-adjust' y la indisponibilidad de ciertos tamaños de fuente.

Los elementos hijos heredan el valor computado de 'font-size' (de otro modo, el efecto de 'font-size-adjust' sería acumulado).

```
P { font-size: 12pt; }
BLOCKQUOTE { font-size: larger }
EM { font-size: 150% }
EM { font-size: 1.5em }
```

**'font-size-adjust'**

- Valor: <número> | none | inherit
- Inicial: ninguno
- Se aplica a: todos los elementos
- Se hereda: si
- Porcentajes: N/A
- Medios: visuales

En las escrituras de doble caja, el tamaño aparente y la legibilidad de una fuente depende menos de su valor 'font-size' (tamaño de la fuente) que del valor 'x-height' (altura de la x), o, mejor aún de la proporción entre estos dos valores, llamado el *valor de relación* (tamaño de la fuente dividido por la altura de la x). A mayor valor de relación es más probable que una fuente de pequeño tamaño resulte legible. Inversamente, tipos con un menor valor de relación se tornarán ilegibles más rápidamente por debajo de cierto tamaño límite que los tipos con un valor de relación más alto. Una franca sustitución de fuentes que descansa solamente en el tamaño de las mismas puede conducir a caracteres ilegibles.

Por ejemplo, la popular fuente Verdana tiene un valor de relación de 0.58; cuando el tamaño de la fuente Verdana tiene 100 unidades, la altura de la x es de 58 unidades. En comparación, la Times New Roman tiene un valor de relación de 0.46. La Verdana, por consiguiente, tenderá a mantener su legibilidad en cuerpos más pequeños que la Times New Roman. Inversamente, la Verdana a menudo parecerá 'demasiado grande' si sustituye a la Times New Roman en un tamaño determinado.

Esta propiedad permite a los autores especificar un valor de relación para un elemento que preserve la altura de la x de la fuente elegida en primera instancia en la fuente sustituta. Los valores tienen los siguientes significados:

**none**

No preserva la altura de la x de la fuente.

**<número>**

Especifica el valor de relación. El número se refiere al valor de relación de la fuente seleccionada en primer término. El factor de escala para las fuentes disponibles se computa según la siguiente fórmula:

$$y(a/a') = c$$

donde:

- y = 'font-size' de la primera opción de fuentes
- a = valor de relación de la fuente seleccionada en primer término
- a' = valor de relación de la fuente disponible
- c = 'font-size' a aplicar a la fuente disponible

Por ejemplo, si la Verdana de 14px (con un valor de relación de 0.58) no está disponible y una fuente disponible tiene un valor de relación de 0.46, el tamaño de fuente de la sustituta debería ser 14\* (0.58/0.46) = 17.65px.

Los ajustes de tamaño de las fuentes tienen lugar al computar el **valor real** de 'font-size'. Dado que la herencia está basada en el **valor computado**, los elementos hijos heredarán los valores sin ajustar.

La primer imagen abajo muestra varias tipografías rasterizadas en un mismo tamaño de fuente (11pt. a 72 ppi), junto con sus valores de relación. Ovserve que los tipos con un valor de relación más alto parecen más grandes que aquellos con valor bajo. Los tipos con muy bajo valor de relación son ilegibles en el tamaño mostrado.

Verdana: .58  
xylophone synergy diaphragm partially hydrogenated  
vegetable shortening or lengthening or resting

Comic Sans MS: .54  
xylophone synergy diaphragm partially hydrogenated  
vegetable shortening or lengthening or resting

Trebuchet MS: .53  
xylophone synergy diaphragm partially hydrogenated  
vegetable shortening or lengthening or resting

Georgia: .5  
xylophone synergy diaphragm partially hydrogenated  
vegetable shortening or lengthening or resting

Myriad Web: .48  
xylophone synergy diaphragm partially hydrogenated  
vegetable shortening or lengthening or resting

Minion Web: .47  
xylophone synergy diaphragm partially hydrogenated  
vegetable shortening or lengthening or resting

Times New Roman: .46  
xylophone synergy diaphragm partially hydrogenated  
vegetable shortening or lengthening or resting

Gill Sans: .46  
xylophone synergy diaphragm partially hydrogenated  
vegetable shortening or lengthening or resting

Bernhard Modern: .4  
xylophone synergy diaphragm partially hydrogenated  
vegetable shortening or lengthening or resting

Café Script Web: .37  
*xylophone synergy diaphragm partially hydrogenated  
vegetable shortening or lengthening or resting*

Flemish Script: .28  
*xylophone synergy diaphragm partially hydrogenated  
vegetable shortening or lengthening or resting*

La imagen siguiente muestra el resultado de 'font-size-adjust' (ajuste del tamaño de la fuente) donde la Verdana ha sido tomada como la "primera opción", junto con el factor de escala aplicado. A medida que son ajustados, los tamaños aparentes son bastante similares a través de todos los tipos, aunque el tamaño real (em) varía en más del 100%. Observe que 'font-size-adjust' tiende a estabilizar también la medida horizontal de las líneas.

|                     |  |
|---------------------|--|
| Verdana:            | <b>1</b>   |
|                     | xylophone synergy diaphragm partially hydrogenated<br>vegetable shortening or lengthening or resting         |
| Comic Sans MS:      | <b>1.07</b>  |
|                     | xylophone synergy diaphragm partially hydrogenated<br>vegetable shortening or lengthening or resting         |
| Trebuchet MS:       | <b>1.09</b>  |
|                     | xylophone synergy diaphragm partially hydrogenated<br>vegetable shortening or lengthening or resting         |
| Georgia:            | <b>1.16</b>  |
|                     | xylophone synergy diaphragm partially hydrogenated<br>vegetable shortening or lengthening or resting         |
| Myriad Web:         | <b>1.2</b>   |
|                     | xylophone synergy diaphragm partially hydrogenated<br>vegetable shortening or lengthening or resting         |
| Minion Web:         | <b>1.23</b>  |
|                     | xylophone synergy diaphragm partially hydrogenated<br>vegetable shortening or lengthening or resting         |
| Times New Roman:    | <b>1.26</b>  |
|                     | xylophone synergy diaphragm partially hydrogenated<br>vegetable shortening or lengthening or resting         |
| Gill Sans:          | <b>1.26</b>  |
|                     | xylophone synergy diaphragm partially hydrogenated<br>vegetable shortening or lengthening or resting         |
| Bernhard Modern:    | <b>1.45</b>  |
|                     | xylophone synergy diaphragm partially hydrogenated<br>vegetable shortening or lengthening or resting         |
| Caflich Script Web: | <b>1.57</b>  |
|                     | <i>xylophone synergy diaphragm partially hydrogenated<br/>vegetable shortening or lengthening or resting</i> |
| Flemish Script:     | <b>2.07</b>  |
|                     | <i>xylophone synergy diaphragm partially hydrogenated<br/>vegetable shortening or lengthening or resting</i> |

### 15.2.5 Propiedad resumida de la fuente: la propiedad 'font'

#### 'font'

|              |  |
|--------------|--|
| Valor:       | [ [ <'font-style'>    <'font-variant'>    <'font-weight'> ]? <'font-size'> [ / <'line-height'> ]? <'font-family'> ]   caption   icon   menu   message-box   small-caption   status-bar   inherit |
| Inicial:     | ver propiedades individuales   |
| Se aplica a: | todos los elementos  |
| Se hereda:   | si   |
| Porcentajes: | permitidos en 'font-size' y 'line-height'  |
| Medios:      | visuales   |

La propiedad 'font' es, exceptuando los casos descriptos [abajo](#), un resumen de la propiedad para determinar 'font-style', 'font-variant', 'font-weight', 'font-size', 'line-height' y 'font-family', en un mismo lugar en la hoja de estilo. La sintaxis de esta propiedad está basada en una notación extractada de la tipografía tradicional para determinar las múltiples propiedades relacionadas con las fuentes.

Todas las propiedades relacionadas con las fuentes son primero restablecidas a sus valores iniciales, incluyendo aquellas listadas en el párrafo anterior, más 'font-stretch' y 'font-size-adjust'. Entonces, aquellas propiedades que tienen asignados valores explícitos en la fórmula resumida 'font' son establecidas con esos valores. Por una definición de los valores iniciales y permitidos, vea las propiedades definidas anteriormente. Por razones de compatibilidad, no es posible establecer para 'font-stretch' y 'font-size-adjust' otros valores que no sean los iniciales usando la propiedad resumida 'font', en su lugar, establezca las propiedades individuales.

```
P { font: 12pt/14pt sans-serif }
P { font: 80% sans-serif }
P { font: x-large/110% "new century schoolbook", serif }
P { font: bold italic large Palatino, serif }
P { font: normal small-caps 120%/120% fantasy }
P { font: oblique 12pt "Helvetica Nue", serif; font-stretch: condensed }
```

En la segunda regla, el valor del porcentaje del tamaño de la fuente ('80%') se refiere al tamaño de fuente del elemento padre. En la tercer regla, el porcentaje de la altura de la línea ('110%') se refiere al tamaño de fuente del propio elemento.

Las primeras tres reglas no especifican 'font-variant' y 'font-weight' explícitamente, entonces estas propiedades reciben su valor inicial ('normal'). Observe que el nombre de la familia de fuentes "new century schoolbook", que contiene espacios, está entre comillas. La cuarta regla establece 'font-weight' como 'bold', 'font-style' como 'italic' e implícitamente establece 'font-variant' como 'normal'.

La quinta regla establece 'font-variant' ('small-caps'), the 'font-size' (120% del tamaño de fuente del elemento padre), 'line-height' (120% del tamaño de fuente) y 'font-family' ('fantasy'). Se entiende que la palabra clave 'normal' se aplica a las dos propiedades restantes: 'font-style' y 'font-weight'.

La sexta regla establece 'font-style', 'font-size' y 'font-family', a las otras propiedades de la fuente les serán aplicados sus valores iniciales. Luego establece 'font-stretch' como 'condensed' ya que esta propiedad no puede establecerse con ese valor usando la propiedad resumida 'font'.

Los siguientes valores se refieren a las fuentes del sistema:

- caption**  
La fuente usada en los títulos de los controles (ej., botones, cajas, etc.)
- icon**  
La fuente usada para rotular los íconos.
- menu**  
La fuente usada en los menús (ej., menús desplegables y listas de menús).
- message-box**  
La fuente usada en las cajas de diálogo.
- small-caption**  
La fuente usada para rotular controles pequeños.
- status-bar**  
La fuente usada en la barra de estado de las ventanas.

Las fuentes del sistema sólo pueden asignarse como un conjunto; es decir, la familia, el tamaño, el peso, el estilo, etc. son establecidos al mismo tiempo. Estos valores pueden modificarse individualmente si se desea. Si en determinada plataforma no existe una fuente con las características indicadas, la aplicación del usuario debe realizar una sustitución inteligente (ej., una versión reducida de la fuente para 'caption' puede ser usada como fuente para 'smallcaption'), o sustituir la fuente predeterminada de la aplicación del usuario. Al igual que para las fuentes regulares, si, para una fuente del sistema, cualquiera de las propiedades individuales no formaran parte de las preferencias del usuario disponibles en el sistema operativo, esas propiedades deberán ser fijadas en sus valores iniciales.

Esta es la razón por la cual esta propiedad es "casi" una propiedad resumida: las fuentes del sistema sólo pueden ser especificadas con esta propiedad, no con 'font-family' en sí, de manera que 'font' permite a los autores hacer más cosas que con la suma de sus subpropiedades. Sin embargo, las propiedades individuales como 'font-weight' aún están dando valores tomados de la fuente del sistema, los cuales pueden ser variados independientemente.

```

    BUTTON { font: 300 italic 1.3em/1.7em "FB Armada", sans-serif }
    BUTTON P { font: menu }
    BUTTON P EM { font-weight: bolder }
```

Si la fuente usada por los menús desplegables en un sistema en particular pasó a ser, por ejemplo, la Charcoal de 9 puntos con un peso igual a 600, entonces los elementos P que sean descendientes de BUTTON serían mostrados como si esta regla estuviera en efecto:

```

    BUTTON P { font: 600 9pt Charcoal }
```

Debido a que fórmula resumida 'font' restablece a su valor inicial cualquier propiedad sin un valor explícito dado, esto tiene el mismo efecto que esta declaración:

```

    BUTTON P {
        font-style: normal;
        font-variant: normal;
        font-weight: 600;
        font-size: 9pt;
        line-height: normal;
        font-family: Charcoal
    }
```

### 15.2.6 Familias genéricas de fuentes

Las familias genéricas de fuentes son un mecanismo de apoyo, un medio para preservar algo de la intención de la hoja de estilo del autor cuando ninguna de las fuentes especificadas pueden ser seleccionadas. Para un óptimo control tipográfico, en las hojas de estilo deben usarse las fuentes con nombres propios.

Las cinco familias genéricas de fuentes se establecen para existir en todas las implementaciones CSS (estas no necesariamente tienen que asignar cinco fuentes concretas distintas). Las aplicaciones del usuario deben proveer de razonables opciones predeterminadas para las familias genéricas de fuentes, las cuales expresen las características de cada familia tanto como sea posible dentro de los límites permitidos por la tecnología subyacente.

Las aplicaciones del usuario son alentadas a permitir al usuario seleccionar opciones alternativas para las fuentes genéricas.

***serif***

Los signos de las fuentes con serif, tal como el término es usado en CSS, tienen rasgos con remates, con terminaciones acampanadas o biseladas, o auténticas terminaciones con serif (incluyendo los serif cuadrados o slab serif). Las fuentes con serif son típicamente de espaciado proporcional. A menudo tienen mayor variación entre los rasgos finos y gruesos que las fuentes de las familias 'sans-serif'. CSS utiliza el término 'serif' para aplicar a una fuente en cualquier tipo de escritura, aunque otros nombres resulten más familiares para algunas escrituras en particular, como Mincho (Japonés), Sung o Song (Chino), Pathang (Coreano). Cualquier fuente descrita de este modo puede usarse para representar la familia genérica 'serif'.

Los ejemplos de fuentes que se ajustan a esta descripción incluyen:

|                   |   |
|-------------------|---|
| Fuentes latinas   | Times New Roman, Bodoni, Garamond, Minion Web, ITC Stone Serif, MS Georgia, Bitstream Cyberbit        |
| Fuentes griegas   | Bitstream Cyberbit  |
| Fuentes cirílicas | Adobe Minion Cyrillic, Excelcior Cyrillic Upright, Monotype Albion 70, Bitstream Cyberbit, ER Bukinst |
| Fuentes hebreas   | New Peninim, Raanana, Bitstream Cyberbit  |
| Fuentes japonesas | Ryumin Light-KL, Kyokasho ICA, Futo Min A101  |
| Fuentes árabes    | Bitstream Cyberbit  |
| Fuentes cherokee  | Lo Cicero Cherokee  |

***sans-serif***

Los signos de las fuentes sans-serif, tal como el término es usado en CSS, no tienen remates -sin formas acampanadas, trazos finales u otros ornamentos-. Las fuentes sans-serif son típicamente de espaciado proporcional. A menudo tienen menor variación entre los rasgos finos y gruesos, comparadas con las fuentes con serif. CSS utiliza el término 'sans-serif' para aplicar a una fuente en cualquier tipo de escritura, aunque otros nombres resulten más familiares para algunas escrituras en particular, como Gothic (Japonés), Kai (Chino), Totum o Kodig (Coreano). Cualquier fuente descrita de este modo puede usarse para representar la familia genérica 'sans-serif'.

Los ejemplos de fuentes que se ajustan a esta descripción incluyen:

|                   |   |
|-------------------|---|
| Fuentes latinas   | MS Trebuchet, ITC Avant Garde Gothic, MS Arial, MS Verdana, Univers, Futura, ITC Stone Sans, Gill Sans, Akzidenz Grotesk, Helvetica |
| Fuentes griegas   | Attika, Typiko New Era, MS Tahoma, Monotype Gill Sans 571, Helvetica Greek  |
| Fuentes cirílicas | Helvetica Cyrillic, ER Univers, Lucida Sans Unicode, Bastion  |
| Fuentes hebreas   | Arial Hebrew, MS Tahoma   |
| Fuentes japonesas | Shin Go, Heisei Kaku Gothic W5  |
| Fuentes árabes    | MS Tahoma   |

***cursive***

Los signos de las fuentes cursive, tal como el término es usado en CSS, generalmente tienen rasgos unidos u otra característica cursiva más marcada que los tipos itálicos. Los signos están parcial o completamente conectados y el resultado se asemeja más a la escritura manual o de pincel que a un tipo impreso. En algunos sistemas de escritura, como el arábigo, las fuentes son casi siempre cursivas. CSS usa el término 'cursive' para ser aplicado a una fuente en cualquier tipo de escritura, aunque otros nombres como Chancery, Brush, Swing y Script son también usados en los nombres de las fuentes.

Los ejemplos de fuentes que se ajustan a esta descripción incluyen:

|                   |  |
|-------------------|--|
| Fuentes latinas   | Caflish Script, Adobe Poetica, Sanvito, Ex Ponto, Snell Roundhand, Zapf-Chancery |
| Fuentes cirílicas | ER Architekt   |
| Fuentes hebreas   | Corsiva  |
| Fuentes árabes    | DecoType Naskh, Monotype Urdu 507  |

*fantasy*

Las fuentes de fantasía, tal como el término es usado en CSS, son principalmente decorativas pero siguen representando caracteres (en oposición a las fuentes de símbolos, que no representan caracteres). Los ejemplos incluyen:

Fuentes latinas Alpha Geometrique, Critter, Cottonwood, FB Reactor, Studz

*monospace*

El único criterio para reconocer una fuente monospace es que todas los signos ocupan un mismo espacio horizontal constante. (Esto puede hacer que algunos sistemas de escritura, como el arábigo, tengan un estilo más peculiar.) El efecto es similar al de las máquinas de escribir manuales y es a menudo utilizada para ilustrar los ejemplos de códigos de computadora.

Los ejemplos de fuentes que se ajustan a esta descripción incluyen:

Fuentes latinas Courier, MS Courier New, Prestige, Everson Mono

Fuentes griegas MS Courier New, Everson Mono

Fuentes cirílicas ER Kurier, Everson Mono

Fuentes japonesas Osaka Monospaced

Fuentes cherokee Everson Mono

### 15.3 Selección de fuentes

La segunda fase del mecanismo de fuentes de CSS2 concierne a la selección por parte de la aplicación del usuario de una fuente basada en las propiedades de la fuente especificadas por el autor, las fuentes disponibles, etc. Los detalles del [algoritmo de equivalencias de fuentes](#) se proporcionan a continuación.

Existen cuatro procedimientos posibles para la selección de fuentes: equivalencia de nombre, equivalencia inteligente, síntesis y transmisión.

***equivalencia del nombre de la fuente***

En este caso, la aplicación del usuario usa una fuente existente y accesible que tiene el mismo nombre de familia que la fuente requerida. (Observe que la apariencia y la métrica no necesariamente pueden coincidir si la fuente que usó el autor del documento y la fuente del sistema cliente son de diferentes empresas.) La información para la equivalencia se restringe a las propiedades de la fuente CSS, incluyendo el nombre de la familia. Este es el único método usado por CSS1.

***equivalencia inteligente de fuentes***

En este caso, la aplicación del usuario usa una fuente existente y accesible que resulta la equivalencia más cercana en apariencia a la fuente requerida. (Observe que la métrica puede no equivaler exactamente.) La información para la equivalencia incluye los datos acerca del tipo de fuente (texto o símbolos), naturaleza del serif, peso, altura de las mayúsculas, altura de la x, ascendentes, descendentes, inclinación, etc.

***sintetizado de la fuente***

En este caso, la aplicación del usuario crea una fuente que no sólo equivale en la apariencia sino también en la métrica a la fuente requerida. La información de la síntesis incluye la información de la equivalencia y típicamente requiere valores más precisos para los parámetros que los usados por algunos esquemas de equivalencia. En particular, la síntesis requiere la métrica exacta del ancho y el carácter para la sustitución de los signos y la información posicional, si todas las características de la composición de la fuente especificada van a ser preservadas.

***transmisión de la fuente***

Finalmente, la aplicación del usuario puede recuperar una fuente de la Web. Esto es similar al proceso de transmisión de imágenes, sonidos o applets de la Web para ser mostrados en el documento actual, y del mismo modo puede ocasionar cierta demora antes de que la página pueda ser desplegada.

El *procesamiento progresivo* es una combinación de transmisión y alguno de los otros métodos; provee una fuente sustituta temporaria (usando la equivalencia de nombre, equivalencia inteligente o la síntesis) para permitir que el contenido sea leído mientras la fuente solicitada es transmitida. Una vez que la fuente verdadera ha sido recuperada con éxito, reemplaza a la fuente temporaria, con la expectativa de que no sea necesario actualizar la pantalla.

***Nota. El procesamiento progresivo necesita información sobre la métrica de la fuente para evitar la recomposición del contenido cuando la fuente real ha sido recuperada y procesada. Esta información acerca de la métrica es suficientemente elocuente por lo que sólo deberá ser especificada como máximo una vez por fuente en un documento.***

#### 15.3.1 Descripciones de fuentes y @font-face

La descripción de una fuente proporciona el puente entre la especificación de la fuente de un autor y los *datos de la fuente*, que son los datos necesarios para dar formato al texto y para procesar los signos abstractas que son asignados a los caracteres -los contornos escalables o los mapas de bits reales-. Las fuentes son *referenciadas* por las propiedades de las hojas de estilo.



La descripción de la fuente se agrega a la base de datos de la fuente y entonces es usado para seleccionar los datos relevantes de la fuente. La descripción de la fuente contiene descriptores como la ubicación de los datos de la fuente en la Web y la caracterización de esos datos de la fuente. Los descriptores de la fuente también se necesitan para comparar las propiedades de la fuente de la hoja de estilo con los datos de una fuente en particular. El nivel de detalle de la descripción de una fuente puede variar entre sólo el nombre de la fuente hasta una lista de las dimensiones de los signos.

Los descriptores de la fuente pueden ser clasificados en tres tipos:

1. aquellos que proporcionan un vínculo entre el uso de la fuente de CSS y la descripción de la fuente (éstas tienen el mismo nombre que la propiedades de la fuentes de CSS correspondientes),
2. el URI para la localización de los datos de la fuente,
3. aquellos que caracterizan mejor a la fuente, para proporcionar un vínculo entre la descripción de la fuente y los datos de la misma.

Todos los descriptores de la fuente se especifican a través de la regla-arroba *@font-face*. La fórmula general es:

```
@font-face { <descripción-de-la-fuente> }
```

donde la <descripción-de-la-fuente> tiene la forma:

```
descriptor: valor;  
descriptor: valor;  
[...]  
descriptor: valor;
```

Cada regla *@font-face* especifica un valor para cada descriptor de la fuente, ya sea implícita o explícitamente. Aquellos valores que no son adjudicados explícitamente toman el valor inicial indicado para cada descriptor en esta especificación. Estos descriptores se aplican únicamente dentro del contexto de la regla *@font-face* en la cual son definidos, y no se aplican a los elementos del lenguaje de documento. De este modo, no hay noción de cuáles son los elementos a los que se aplican los descriptores, o si los valores son heredados por los elementos hijos.

Los descriptores de fuentes disponibles se describen en secciones posteriores en esta especificación.

Por ejemplo, aquí la fuente 'Robson Celtic' es definida y referenciada en una hoja de estilo contenida en un documento HTML.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">  
<HTML>  
  <HEAD>  
    <TITLE>Prueba de fuente</TITLE>  
    <STYLE TYPE="text/css" MEDIA="screen, print">  
      @font-face {  
        font-family: "Robson Celtic";  
        src: url("http://site/fonts/rob-celt")  
      }  
      H1 { font-family: "Robson Celtic", serif }  
    </STYLE>  
  </HEAD>  
  <BODY>  
    <H1> Este título es mostrado usando la Robson Celtic</H1>  
  </BODY>  
</HTML>
```

La hoja de estilo (en el elemento STYLE) contiene una regla CSS que especifica que todos los elementos H1 usarán la familia de fuentes 'Robson Celtic'.

Una implementación de CSS1 buscará en el cliente una fuente cuyo nombre de familia y otras propiedades sean equivalentes a la 'Robson Celtic' y, si no puede encontrarla, usará la fuente con serif de respaldo específica de la aplicación del usuario (la cuál está [establecida para existir](#)).

Una aplicación del usuario que implemente CSS2 primero examinará las reglas *@font-face* en busca de un descriptor de fuente definiendo la 'Robson Celtic'. Este ejemplo contiene una regla que se ajusta. Aunque esta regla no contiene demasiados datos de la fuente, tiene una URI donde la fuente puede ser recuperada para procesar este documento. Las fuentes transmitidas no deben hacerse disponibles para otras aplicaciones. Si no se encuentra una *@font-face* que coincida, la aplicación del usuario intentará encontrar una equivalente del mismo modo que una aplicación del usuario que implementa CSS1.

Observe que si la fuente 'Robson Celtic' *hubiera* sido instalada en el sistema cliente, esto habría provocado que la AU agregara una entrada en la base de datos de las fuentes para la copia instalada tal como se describe en la sección de [algoritmo de equivalencia de fuentes](#). La copia instalada habría resultado la equivalente antes que la fuente a transmitir, en el ejemplo anterior.



Las implementaciones de CSS1, que no entienden la regla @font-face encontrarán la llave de apertura e ignorarán lo que sigue hasta la correspondiente llave de cierre. Esta regla-arroba está conforme a los requerimientos de CSS en cuanto al análisis con compatibilidad futura. Los analizadores pueden ignorar estas reglas sin provocar un error.

Teniendo los descriptores de la fuente separados de los datos de la fuente tiene un beneficio adicional además de permitir la selección y/o sustitución de la fuente. La protección de los datos y las restricciones de copia en los descriptores de fuentes puede ser mucho más débil que en los datos completos de la fuente. Así, es posible instalar la definición de la fuente localmente, o por lo menos mantenerla en un archivo temporal por si aparece en una hoja de estilo referenciada a menudo; esto no requerirá tener que acceder a la definición completa de la fuente a través de la Web más de una vez por fuente nombrada.

Si un descriptor de fuente está duplicado, el último en aparecer prevalece y el resto debe ignorarse.

También, cualquier descriptor que no es reconocido o útil para la aplicación del usuario se debe ignorar. Las futuras versiones de CSS pueden permitir descriptores adicionales con el propósito de mejorar la sustitución, equivalencia o síntesis de fuentes.

### 15.3.2 Descriptores para seleccionar una fuente: 'font-family', 'font-style', 'font-variant', 'font-weight', 'font-stretch' y 'font-size'

Los siguientes descriptores tienen el mismo nombre que las correspondientes propiedades de fuentes en CSS2, y tienen un solo valor o una lista de valores separados por coma.

Los valores de esa lista son, excepto cuando se señale explícitamente, los mismos que aquellos correspondientes a las propiedades en CSS2. Si hay un solo valor, ese es el valor que debe corresponder. Si hay una lista, cualquier ítem de la lista constituye el correspondiente. Si el descriptor es omitido en @font-face, se usa el valor inicial para el descriptor.

#### 'font-family' (Descriptor)

Valor: [ <nombre-de-la-familia> | <familia-genérica> ] [, [ <nombre-de-la-familia> | <familia-genérica> ] ]\*

Inicial: depende de la aplicación del usuario

Medios: visuales

Este es el descriptor para el nombre de la familia de fuentes de una fuente y toma los mismos valores que la propiedad 'font-family'.

#### 'font-style' (Descriptor)

Valor: all | [ normal | italic | oblique ] [, [normal | italic | oblique] ]\*

Inicial: all

Medios: visuales

Este es el descriptor para el estilo de una fuente y toma los mismos valores que la propiedad 'font-style', excepto que se permite una lista separada por comas.

#### 'font-variant' (Descriptor)

Valor: [normal | small-caps] [, [normal | small-caps]]\*

Inicial: normal

Medios: visuales

Esta es la indicación CSS de si el tipo es la versión versalita de una fuente. Toma los mismo valores que la propiedad 'font-variant', excepto que se permite una lista separada por comas.

**Nota. Los tipos cirílico pryamo• pueden ser descriptos en 'font-variant' como una variante small-caps, lo cuál le dará más consistencia con los tipos latinos (y el tipo kursiv que los acompaña descripto en 'font-style' con el estilo italic por la misma razón).**

#### 'font-weight' (Descriptor)

Valor: all | [normal | bold | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900] [, [normal | bold | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900]]\*

Inicial: all

Medios: visuales

Este es el descriptor para el peso de un tipo en relación a otros en la misma familia de fuentes. Toma los mismos valores que la propiedad 'font-weight' con tres excepciones:

1. las palabras clave relativas (bolder, lighter) no son permitidas.
2. una lista de valores separados por comas está permitida para fuentes que contienen múltiples variantes de peso.
3. una palabra clave adicional, 'all' está permitida, la cuál significa que la fuente deberá buscar equivalencia con todos los posibles pesos; ya sea porque contiene múltiples pesos o porque ese tipo posee una sola variante de peso.

**'font-stretch' (Descriptor)**

*Valor:* all | [ normal | ultra-condensed | extra-condensed | condensed | semi-condensed | semi-expanded | expanded | extra-expanded | ultra-expanded ] [, [ normal | ultra-condensed | extra-condensed | condensed | semi-condensed | semi-expanded | expanded | extra-expanded | ultra-expanded ] ]\*

*Inicial:* normal

*Medios:* visuales

Esta es la indicación en CSS de la naturaleza del condensado o expandido del tipo en relación a otros de la misma familia de fuentes. Toma los mismos valores que la propiedad 'font-stretch' excepto que:

- las palabras clave relativas (wider, narrower) no son permitidas
- una lista separada por comas es permitida
- La palabra clave 'all' es permitida

**'font-size' (Descriptor)**

*Valor:* all | <medida> [, <medida>]\*

*Inicial:* all

*Medios:* visuales

Este es el descriptor para los tamaños previstos para esta fuente. Sólo las unidades de [medida absoluta](#) son permitidos, en contraste con la propiedad 'font-size', que permite medidas y tamaños absolutos y relativos. Una lista de medidas absolutas separadas por comas es permitida.

El valor inicial 'all' es conveniente para la mayoría de las fuentes escalables, de manera que este descriptor es principalmente para usar @font-face con fuentes de mapas de bits, o fuentes escalables diseñadas para ser rasterizadas en un rango restringido de tamaños.

**15.3.3 Descriptores para la Calificación de los Datos de una Fuente: 'unicode-range'**

El siguiente descriptor es opcional dentro de la definición de la fuente, pero se usa para evitar la verificación o la transmisión de una fuente que no tiene los suficientes signos para procesar una carácter en particular.

**'unicode-range' (Descriptor)**

*Valor:* <rango-u> [, <rango-u>]\*

*Inicial:* U+0-7FFFFFFF

*Medios:* visuales

Este es el descriptor para el [rango de caracteres ISO 10646](#) cubiertos por la fuente.

Los valores de <rango-u> están expresados usando números hexadecimales precedidos por "U+", correspondiendo a las posiciones en el código de caracteres ISO 10646 ([ISO10646](#)).

Por ejemplo, U+05D1 es la 'letra hebrea bet' en caracteres ISO 10646. Para valores fuera del Plano Multilingual Básico (Basic Multilingual Plane o BMP), se anteponen números adicionales correspondientes al número de plano, como éste: U+A1234 que es el carácter en el Plano 10 en la posición hexadecimal 1234 del código. En el momento en que esto es escrito, no se han asignado caracteres fuera del BMP. Los ceros antepuestos (por ejemplo, 0000004D) son válidos, pero no requeridos.

El [valor inicial](#) de este descriptor cubre no sólo el Plano Multilingual Básico (BMP) completo, el cuál podría ser expresado como U+0-FFFF, sino también el repertorio completo de ISO 10646. De ese modo, el valor inicial expresa que la fuente puede tener signos para los caracteres en algún lugar de ISO 10646. Especificando un valor para 'unicode-range' provee información para hacer la búsqueda más eficiente, declarando un rango acotado en el que la fuente puede tener signos para los caracteres. La fuente no necesita efectuar la búsqueda de caracteres fuera de este rango.

Los valores pueden escribirse con cualquier número de dígitos. Para números simples, el carácter '?' se asume que significa 'cualquier valor', lo cuál crea un *rango* de posiciones de los caracteres. De este modo, usando un *número simple*:

unicode-range: U+20A7  
sin comodines - indica una sola posición de caracteres (el símbolo de la peseta española)

unicode-range: U+215?  
un comodín, cubre el rango entre 2150 y 215F (las fracciones)

unicode-range: U+00??  
dos comodines, cubre el rango entre 0000 y 00FF (Latin-1)

unicode-range: U+E??  
dos comodines, cubre de 0E00 a 0EFF (la escritura Lao)

Un *par de números* en este formato pueden combinarse con un guión para indicar rangos más extensos. Por ejemplo:

unicode-range: U+AC00-D7FF  
el rango es de AC00 a D7FF (el área de sílabas en Hangul)

Pueden especificarse múltiples rangos discontinuos separados por una coma. Como sucede con otras listas separadas por comas en CSS, cualquier [espacio en blanco](#) antes y después de la coma es [ignorado](#). Por ejemplo:

unicode-range: U+370-3FF, U+1F??

Esto cubre el rango entre 0370 y 03FF (griego moderno) más 1F00 a 1FFF (griego politonal antiguo).

unicode-range: U+3000-303F, U+3100-312F, U+32??, U+33??, U+4E00-9FFF, U+F9000-FAFF, U+FE30-FE4F

Uno de los peores casos en términos de verborragia, esto indica con mucha precisión que esta fuente (extremadamente extensa) contiene sólo caracteres chinos de ISO 10646, sin incluir ningún carácter que sea fundamentalmente japonés o coreano. El rango es de 3000 a 303F (CJK símbolos y puntuación) más 3100 a 312F (Bopomofo) más 3200 a 32FF (anexo de letras y meses CJK) más 3300 a 33FF (zona de compatibilidad CJK) más 4E00 a 9FFF (ideogramas unificados CJK) más F900 a FAFF (compatibilidad de ideogramas CJK) más FE30 a FE4F (formularios de compatibilidad CJK).

Una representación más adecuada para una fuente típicamente china podría ser:

unicode-range: U+3000-33FF, U+4E00-9FFF

unicode-range: U+11E00-121FF

Esto cubre un registro propuesto para los pictogramas aztecas, cubriendo un rango entre 1E00 y 21FF en el plano 1.

unicode-range: U+1A00-1A1F

Esta fuente cubre un registro propuesto para la escritura irlandesa ogham cubriendo un rango entre 1A00 y 1A1F

### 15.3.4 Descriptor para Valores Numéricos: 'units-per-em'

El siguiente descriptor especifica el número de "unidades" por eme; estas unidades pueden ser usadas por algunos otros descriptores para expresar varias medidas, de modo que 'units-per-em' se requiere si otros descriptores dependen de él.

**Nota de la traducción:**

La unidad de medida EM es lo que en la tipografía tradicional se conocía como "cuadratín", un cuadrado cuyo lado tiene igual medida que el cuerpo del tipo. **No** es la medida de la letra EME, sólo se toma esa letra para la denominación debido a que normalmente es la más ancha del alfabeto.

**'units-per-em' (Descriptor)**

Valor: <número>

Inicial: indefinido

Medios: visuales

Este es el descriptor para el número de [unidades coordinadas en el cuadrado eme](#), el tamaño de la grilla de diseño en la cuál los signos son trazados.

### 15.3.5 Descriptor para el Referenciado: 'src'

Este descriptor se requiere para referenciar los datos reales de la fuente, ya sea transmisibles o instalados localmente.

**'src' (Descriptor)**

Valor: [ <uri> [formato(<cadena> [, <cadena>]\*)] ] | <nombre-de-la-fuente> ] [, <uri> [formato(<cadena> [, <cadena>]\*)] ] | <nombre-de-la-fuente> ]\*

Inicial: indefinido

Medios: visuales

Esto es una lista separada por comas con un orden de prioridades de referencias externas y/o nombres de fuentes instaladas localmente. La referencia externa apunta a los datos de la fuente en la Web. Esto se requiere si la Fuente Web será transmitida. Los recursos de la fuente puede ser un sub-conjunto de la fuente de origen, por ejemplo puede contener sólo los signos necesarios para la página actual o para un conjunto de páginas.

La referencia externa consiste de un URI, seguido de una indicación opcional acerca del formato del recurso de la fuente que se encuentra en ese URI, y esta indicación deberá ser usada por los clientes para evitar la exploración de vínculos a fuentes en un formato inapropiado para el uso. Como con cualquier referencia en hipertexto, puede haber otros formatos disponibles, pero el cliente tiene una mejor idea acerca de lo que presuntamente va a encontrar, de una manera más sólida que intentando analizar las extensiones de archivos en los URI.

La indicación del formato contiene una lista de cadenas de formato separada por comas que denota formatos muy conocidos de fuentes. La aplicación del usuario reconocerá el nombre de los formatos de fuente que soporta, y evitará la transmisión de fuentes en formatos que no reconoce.

Una lista inicial de cadenas definidas por esta especificación y representando los formatos que probablemente serán usados por las implementaciones de varias plataformas es:

| Cadena              | Formato de la fuente               | Ejemplo de extensiones comunes |
|---------------------|------------------------------------|--------------------------------|
| "truedoc-pfr"       | TrueDoc™ Portable Font Resource    | .pfr                           |
| "embedded-opentype" | Embedded OpenType                  | .eot                           |
| "type-1"            | PostScript™ Type 1                 | .pfb, .pfa                     |
| "truetype"          | TrueType                           | .ttf                           |
| "opentype"          | OpenType, incluyendo TrueType Open | .ttf                           |
| "truetype-gx"       | TrueType con extensiones GX        |                                |
| "speedo"            | Speedo                             |                                |
| "intellifont"       | Intellifont                        |                                |

Así como [otros URIs en CSS](#), el URI puede ser parcial, en cuyo caso se determina en relación a la localización de la hoja de estilo conteniendo @font-face.

El <nombre-de-la-fuente> instalada localmente es el nombre completo de la fuente instalada localmente. El *nombre completo de la fuente* es el nombre de la fuente tal como es reportada por el sistema operativo y es el nombre más factible de ser usado en las hojas de estilo del lector, las hojas de estilo predeterminadas por el browser o las probables hojas de estilo del autor en una intranet. Algunos agregados como negrita, itálica y subrayada son usados a menudo para diferenciar tipos dentro de una misma familia de fuentes. Para [más información acerca del nombre completo de las fuentes](#) por favor consulte las notas siguientes:

La notación para el <nombre-de-la-fuente> es el nombre completo de la fuente, que debe ir entre comillas debido a que puede contener cualquier carácter, incluyendo espacios y signos de puntuación, y también debe ir entre comillas en la ubicación local "(" y ")".

```
src: url("http://foo/bar")
    un URI completo y ninguna información acerca del formato(s) de fuente disponibles en el sitio
src: local("BT Century 751 No. 2 Semi Bold Italic")
    referencia a un tipo particular de una fuente instalada localmente
src: url("../fonts/bar") format("truedoc-pfr")
    un URI parcial que tiene una fuente disponible en formato TrueDoc
src: url("http://cgi-bin/bar?stuff") format("opentype", "intellifont")
    un URI completo, en este caso a un script, que puede generar dos formatos distintos -OpenType e Intellifont
src: local("T-26 Typeka Mix"), url("http://site/magda-extra") format("type-1")
    se dan dos alternativas, primero una fuente instalada localmente y en segundo lugar una fuente descargable
    disponible en formato Type 1.
```

El acceso a las fuentes instaladas localmente es a través del <nombre-de-la-fuente>. El nombre de la fuente no es verdaderamente único, ni es realmente independiente de la plataforma o el formato de la fuente, pero por el momento es la mejor manera de identificar los datos de una fuente instalada localmente. El uso del nombre de la fuente puede hacerse más preciso proporcionando una indicación del complemento de signos requerido. Esto puede realizarse indicando el rango de posiciones de los caracteres en ISO 10646 para los cuales la fuente provee algunos signos (ver 'unicode-range').

### 15.3.6 Descriptores para las Equivalencias: 'panose-1', 'stemv', 'stemh', 'slope', 'cap-height', 'x-height', 'ascent' y 'descent'

Estos descriptores son opcionales en una definición para CSS2, pero pueden utilizarse si el autor desea una equivalencia inteligente de fuentes o un ajuste en el tamaño de la fuente.

'panose-1' (Descriptor)

Valor:    [<entero>]{10}

Inicial:   0 0 0 0 0 0 0 0 0

Medios:   visuales

Este es el descriptor para el [número Panose-1](#) y consiste en diez enteros decimales, separados por [espacios en blanco](#). Una lista separada por comas no es permitida para este descriptor, debido a que el sistema Panose-1 puede indicar que un rango de valores son equivalentes. El valor inicial es cero, que significa "cualquiera", para cada dígito PANOSE; todas las fuentes equivaldrán con el número PANOSE si se usa este valor. El uso del descriptor Panose-1 es fuertemente recomendado para las fuentes latinas. Para mayores detalles, ver el [Apéndice C](#).

'stemv' (Descriptor)

Valor:    <número>

Inicial:   indefinido

Medios:   visuales

Este es el descriptor para el [ancho del trazo vertical](#) de una fuente. Si el valor no es definido, el descriptor no se usa para efectuar la equivalencia. Si este descriptor es usado, el descriptor ['units-per-em'](#) también debe ser usado.

#### 'stemh' (Descriptor)

*Value:* <número>  
*Inicial:* indefinido  
*Medios:* visuales

Este es el descriptor para el [ancho del trazo horizontal](#) de la fuente. Si el valor no es definido, el descriptor no se usa para efectuar la equivalencia. Si este descriptor es usado, el descriptor '[units-per-em](#)' también debe ser usado.

#### 'slope' (Descriptor)

*Valor:* <número>  
*Inicial:* 0  
*Medios:* visuales

Este es el descriptor para el [ángulo del asta vertical](#) de la fuente.

#### 'cap-height' (Descriptor)

*Valor:* <número>  
*Inicial:* indefinido  
*Medios:* visuales

Este es el descriptor para el número de la [altura de los signos en mayúsculas](#) de la fuente. Si el valor no es definido, el descriptor no se usa para efectuar la equivalencia. Si este descriptor es usado, el descriptor '[units-per-em](#)' también debe ser usado.

#### 'x-height' (Descriptor)

*Valor:* <número>  
*Initial:* indefinido  
*Medios:* visuales

Este es el descriptor para la [altura de los signos en minúsculas](#) de la fuente. Si el valor no es definido, el descriptor no se usa para efectuar la equivalencia. Si este descriptor es usado, el descriptor '[units-per-em](#)' también debe ser usado. Este descriptor puede ser muy útil cuando se usa la propiedad 'font-size-adjust' debido a que el cómputo del valor z de la fuente elegida requiere del tamaño de la fuente y también de la altura de la x; por eso se recomienda incluir este descriptor.

#### 'ascent' (Descriptor)

*Valor:* <número>  
*Inicial:* indefinido  
*Medios:* visuales

Este es el descriptor para la [altura máxima sin acentos](#) de la fuente. Si el valor no es definido, el descriptor no se usa para efectuar la equivalencia. Si este descriptor es usado, el descriptor '[units-per-em](#)' también debe ser usado.

#### 'descent' (Descriptor)

*Valor:* <número>  
*Inicial:* indefinido  
*Medios:* visuales

Este es el descriptor para la [máxima profundidad sin acentos](#) de la fuente. Si el valor no es definido, el descriptor no se usa para efectuar la equivalencia. Si este descriptor es usado, el descriptor '[units-per-em](#)' también debe ser usado.

### 15.3.7 Descriptores para la Síntesis: 'widths', 'bbox' y 'definition-src'

Sintetizar una fuente significa, como mínimo, lograr la equivalencia en la métrica horizontal de la fuente especificada. Por consiguiente, para la síntesis, esta información métrica debe estar disponible. De igual modo, el procesamiento progresivo requiere de la métrica horizontal para evitar la actualización del contenido cuando la fuente real ha sido transmitida. Aunque los descriptores siguientes son opcionales para una definición en CSS2, algunos son requeridos si el autor desea el sintetizado (o el procesamiento progresivo sin actualización). La fuente real deberá quedar disponible, la sustituta deberá ser reemplazada por la fuente real. Cualquiera de estos descriptores que se encuentren presentes deberán ser usados para proporcionar una mejor y más rápida aproximación a la fuente buscada.

De estos descriptores, los más importantes son 'widths' y 'bbox', los cuales se usan para prevenir la actualización del texto cuando la fuente real quede disponible. Además, los descriptores en el [conjunto de descriptores usados para la equivalencia](#) pueden ser usados para proporcionar una mejor síntesis de la apariencia de la fuente real.

#### 'widths' (Descriptor)

*Valor:* [<rango-u> ]? [<número> ]+ [, [<rango-u> ]? <número> ]+]  
*Inicial:* indefinido  
*Medios:* visuales

Este es el descriptor para el [ancho de los signos](#). El valor es una lista separada por comas de valores de <rango-u>, cada uno seguido por uno o más anchos de los signos. Si este descriptor es usado, el descriptor '[units-per-em](#)' también debe ser usado.

Si el <rango-u> es omitido, se considera un rango de U+0-7FFFFFFF, lo que cubre todos los caracteres y sus signos. Si no se dan suficientes anchos de signos, el último de la lista es repetido hasta cubrir ese rango. Si se dan demasiados, los anchos extras son [ignorados](#).

For example:

```
widths: U+4E00-4E1F 1736 1874 1692
widths: U+1A?? 1490, U+215? 1473 1838 1927 1684 1356 1792
        1815 1848 1870 1492 1715 1745 1584 1992 1978 1770
```

En el primer ejemplo se da un rango de 32 caracteres, de 4E00 a 4E1F. El signo correspondiente al primer carácter (4E00) tiene un ancho de 1736, el segundo tiene un ancho de 1874, y el tercero, 1692. Como no se especifica la suficiente cantidad de anchos, la última medida se repite para cubrir el resto del rango especificado. El segundo ejemplo especifica un sólo ancho, 1490, para todo un rango de 256 signos y luego declara los anchos para un rango de 16 signos.

Este descriptor no puede describir múltiples signos correspondientes a un mismo carácter o ligaduras de múltiples caracteres. De modo que este descriptor *solamente* puede ser usado para escrituras que no tienen formatos contextuales o ligaduras obligatorias. No obstante es útil en esas situaciones. Las escrituras que requieren la asignación de signos a caracteres en la forma uno-a-varios o varios-a-varios no pueden usar actualmente este descriptor para posibilitar la síntesis de la fuente, aunque aún pueden usar la transmisión de fuentes o la equivalencia inteligente.

**'bbox' (Descriptor)**

Valor: <número>, <número>, <número>, <número>  
Inicial: indefinido  
Medios: visuales

Este es el descriptor para la [máxima caja externa](#) de la fuente. El valor es una lista separada por comas de exactamente cuatro números especificando, en este orden, la coordenada x inferior izquierda, la y inferior izquierda, la x superior derecha, la y superior derecha, de la caja externa de la fuente completa.

**'definition-src' (Descriptor)**

Valor: <uri>  
Inicial: indefinido  
Medios: visuales

Los descriptores de las fuentes pueden estar ya sea dentro de la definición de la fuente en una hoja de estilo o pueden ser proporcionados en un *recurso de definición de fuentes* aparte identificado por un URI. La última aproximación puede reducir el tráfico de la red cuando múltiples hojas de estilo referencian una misma fuente.

### 15.3.8 Descriptores para la Alineación: 'baseline', 'centerline', 'mathline' y 'topline'

Estos descriptores son usados para alinear las orientaciones de distintas escrituras entre sí.

**'baseline' (Descriptor)**

Valor: <número>  
Inicial: 0  
Medios: visuales

Este es el descriptor para la [línea de base inferior](#) de una fuente. Si a este descriptor se le da un valor diferente al predeterminado (distinto a cero), el descriptor '[units-per-em](#)' también debe ser usado.

**'centerline' (Descriptor)**

Valor: <número>  
Inicial: undefined  
Medios: visuales

Este es el descriptor para la [línea de base central](#) de una fuente. Si el valor no es definido, la AU puede emplear diversos métodos heurísticos como el valor del punto medio de los rasgos ascendentes y descendentes. Si se usa este descriptor, el descriptor '[units-per-em](#)' también debe ser usado.

**'mathline' (Descriptor)**

Valor: <número>  
Inicial: undefined  
Medios: visuales



Este es el descriptor para la [línea de base matemática](#) de una fuente. Si no es definida, la AU puede usar la línea de base central. Si se usa este descriptor, el descriptor '[units-per-em](#)' también debe ser usado.

**'topline' (Descriptor)**

Valor: <número>  
Inicial: indefinido  
Medios: visuales

Este es el descriptor para [línea de base superior](#) de una fuente. Si no es definida, la AU puede usar un valor aproximado como el de los rasgos ascendentes. Si se usa este descriptor, el descriptor '[units-per-em](#)' también debe ser usado.

### 15.3.9 Ejemplos

Dada la siguiente lista de fuentes:

|                     |                                  |
|---------------------|----------------------------------|
| Swiss 721 light     | light & light italic             |
| Swiss 721           | roman, bold, italic, bold italic |
| Swiss 721 medium    | medium & medium italic           |
| Swiss 721 heavy     | heavy & heavy italic             |
| Swiss 721 black     | black, black italic, & black #2  |
| Swiss 721 Condensed | roman, bold, italic, bold italic |
| Swiss 721 Expanded  | roman, bold, italic, bold italic |

Las siguientes descripciones de las fuentes podrían usarse para permitir su transmisión.

```
@font-face {
  font-family: "Swiss 721";
  src: url("swiss721lt.pfr"); /* Swiss 721 light */
  font-style: normal, italic;
  font-weight: 200;
}
@font-face {
  font-family: "Swiss 721";
  src: url("swiss721.pfr"); /* La Swiss 721 normal */
}
@font-face {
  font-family: "Swiss 721";
  src: url("swiss721md.pfr"); /* Swiss 721 medium */
  font-style: normal, italic;
  font-weight: 500;
}
@font-face {
  font-family: "Swiss 721";
  src: url("swiss721hvy.pfr"); /* Swiss 721 heavy */
  font-style: normal, italic;
  font-weight: 700;
}
@font-face {
  font-family: "Swiss 721";
  src: url("swiss721blk.pfr"); /* Swiss 721 black */
  font-style: normal, italic;
  font-weight: 800,900; /* observe el interesante problema de que
                        el peso 900 para la itálica no existe */
}
@font-face {
  font-family: "Swiss 721";
  src: url(swiss721.pfr); /* La Swiss 721 condensada */
  font-stretch: condensed;
}
@font-face {
  font-family: "Swiss 721";
  src: url(swiss721.pfr); /* Swiss 721 expandida */
  font-stretch: expanded;
}
```

## 15.4 Características de las Fuentes

### 15.4.1 Introducción a las Características de las Fuentes



En esta sección son enumeradas las características de las fuentes que se han encontrado útiles para la equivalencia, síntesis y transmisión de fuentes por parte del cliente en las diversas plataformas que acceden a la Web. La información puede resultar útil para cualquier medio que precise utilizar fuentes en la Web por algún otro sistema que no sea la incrustación de los datos de la fuente en ese medio.

Estas características son usadas para tipificar las fuentes. No son específicas de CSS o de las hojas de estilo. En CSS, cada característica es especificada por un descriptor de la fuente. Estas características también podrían asignarse a nodos VRML, estructuras de aplicaciones CGM, a una API de Java o lenguajes de hojas de estilo alternativos. Las fuentes recuperadas por un medio y guardadas en la caché de un servidor proxy podrían ser reutilizadas por otro medio, ahorrando tiempo y ancho de banda, si se usa exhaustivamente un sistema común de características de la fuentes.

Una lista parcial de ejemplos de tales medios incluye:

- Formatos de vectores en 2-D
  - Computer Graphics Metafile
  - Simple Vector Format
- Formatos de gráficos en 3-D
  - VRML
  - 3DMF
- Tecnologías de incrustación de objetos
  - Java
  - Active-X
  - Obliq

### 15.4.2 Nombre completo de la fuente

Este es el nombre completo de un tipo particular dentro de una familia de fuentes. Típicamente incluye una variedad de calificativos no estandarizados o *agregados* en el nombre de la familia de fuentes. Puede incluir también el nombre o abreviatura de la empresa, a menudo antepuesto al nombre de la familia de fuentes. Solamente se usa para referirse a fuentes instaladas localmente, debido a que el formato del nombre y sus agregados puede variar de una plataforma a otra. Debe ir entre comillas.

Por ejemplo, el nombre de la fuente TrueType y la fuente PostScript puede diferir en el uso de espacios, puntuación y en la abreviatura de ciertas palabras (ej., para ajustarse a los requerimientos de diversos sistemas o intérpretes de impresoras se limita la longitud de los nombres). Por ejemplo, los espacios no están permitidos en el nombre de una fuente PostScript, pero son comunes en el nombre completo de la fuente. La tabla de nombres de las TrueType también puede contener el nombre PostScript, que no contiene espacios.

El nombre en la definición de la fuente es importante debido a que es el vínculo a cualquier fuente instalada localmente. Es importante que el nombre sea sólido, ya sea con respecto a la independencia de la plataforma como a la independencia de la aplicación. Por esta razón, el nombre no debería ser específico de ninguna aplicación -o lenguaje-.

La solución ideal sería tener un nombre que identifique unívocamente cada colección de datos de las fuentes. En la práctica, este nombre no existe actualmente para los datos de las fuentes. Fuentes con tipos del mismo nombre pueden variar en cierto número de descriptores. Algunos de estos descriptores, como los de los diferentes complementos de los signos de la fuente, pueden resultar insignificantes si los signos requeridos se hallan en la fuente. Otros descriptores, como los de la métrica horizontal, convierten en incompatibles a fuentes con el mismo nombre. No parece posible definir una regla que identifique todas las incompatibilidades, pero esto no evitará el uso de una copia local de los datos de una fuente con un nombre dado, cuando ésta sea perfectamente adecuada. Por lo tanto, solamente el rango de caracteres ISO 10646 se usará para calificar las equivalencias en el nombre de las fuentes.

Por cuanto el principal propósito del nombre de la fuente en la definición de una fuente es el de permitir a las aplicaciones del usuario determinar dónde se encuentra una copia local de los datos de la fuente especificada, el nombre debe ser el que estará en todas las copias legítimas de los datos de la fuente. De otra manera, se generará un tráfico innecesario en la Web debido a las equivalencias fallidas con la copia local.

### 15.4.3 Unidades coordinadas en el cuadrado eme

Ciertos valores, como los de la métrica horizontal, son expresados en unidades relativas a un cuadrado abstracto cuyos lados es la distancia comprendida entre líneas de tipos en el mismo cuerpo del tipo. Este cuadrado es llamado *cuadrado eme* y representa la grilla de diseño sobre la cuál se definen las siluetas de los signos. El valor de este descriptor especifica en cuántas unidades se divide el cuadrado EME. Los valores más comunes son, por ejemplo, 250 (Intellifont), 1000 (Type 1) y 2048 (TrueType, TrueType GX y OpenType).

Si este valor no es especificado, se torna imposible conocer qué significa cada medida de la fuente. Por ejemplo, una fuente tiene los signos en minúsculas de una altura de 450; otra tiene unos más pequeños ¡de una altura de 890! Los números son en realidad fracciones; la primer fuente tiene 450/1000 y la segunda tiene 890/2048, lo cuál la hace ciertamente más chica.

### 15.4.4 Línea de Base Central

Esto da la posición, en el [cuadrado eme](#), de la línea de base central. La línea de base central es usada por las escrituras ideográficas para su alineación, del mismo modo que la línea de base inferior es usada por las escrituras latina, griega y cirílica.

### 15.4.5 Codificación de la Fuente

Ya sea explícita o implícitamente, cada fuente tiene una tabla asociada a ella, la *tabla de codificación de la fuente*, que indica a qué carácter representa cada signo. Esta tabla también es mencionada como *vector de codificación*.

De hecho, muchas fuentes contienen varios signos para el mismo carácter. Cuál de esos signos deberán ser usados depende tanto de la reglas del lenguaje como de las preferencias del diseñador.

En árabe, por ejemplo, todas las letras tiene cuatro (o dos) formas diferentes, dependiendo si la letra se usa al comienzo de una palabra, al medio, al final o aislada. Es el mismo carácter en todos los casos, de manera que hay un sólo carácter en el documento fuente, pero cuando es impreso, luce cada vez diferente.

También hay fuentes que permiten al diseñador gráfico elegir entre varias formas alternativas. Desafortunadamente, CSS2 no proporciona todavía el modo de seleccionar esas alternativas. Actualmente, la forma predeterminada es siempre la que resulta seleccionada en tales fuentes.

### 15.4.6 Nombre de la familia de fuentes

Este especifica la porción del nombre de la familia en el nombre de una fuente. Por ejemplo, el nombre de la familia en la Helvetica-Bold es Helvetica y el nombre de la familia en la ITC Stone Serif Semibold Italic es ITC Stone Serif. Algunos sistemas tratan los agregados relativos a los tipos condensados o expandidos como si fueran parte del nombre de la familia.

### 15.4.7 Ancho de los signos

Esta es una lista de anchos, en la grilla de diseño, para el signo correspondiente a cada carácter. La lista está ordenada por el punto en el código ISO 10646. Los anchos no pueden ser especificados convenientemente cuando más de un signo es adjudicado al mismo carácter o cuando hay ligaduras obligatorias.

### 15.4.8 Ancho del asta horizontal

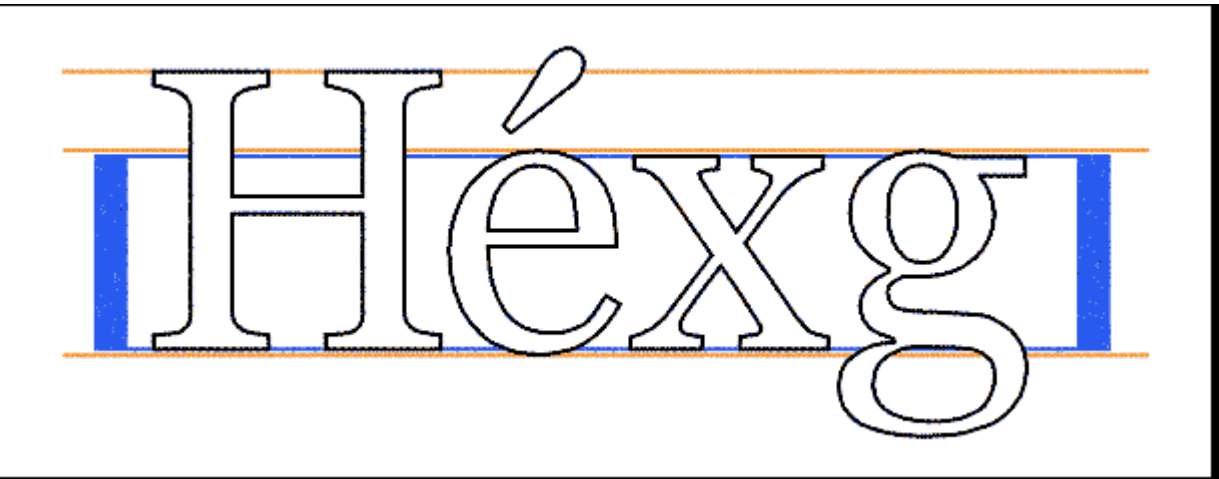
Este valor se refiere al trazo *dominante* de la fuente. Puede haber dos o más anchos diseñados. Por ejemplo, el asta vertical principal de los caracteres romanos diferirá de los trazos finos de la "M" y la "N" con serif, además puede haber diferentes anchos para los caracteres en mayúsculas y minúsculas en la misma fuente. También, ya sea por diseño o por error, todos los trazos pueden tener anchos ligeramente diferentes.

### 15.4.9 Altura de los signos en mayúsculas

Esta medida es la coordenada y de la línea superior de las mayúsculas rectas en las escrituras latina, griega y cirílica, tomada desde la línea de base. Este descriptor no es necesariamente útil para fuentes que no contienen signos para estas escrituras.

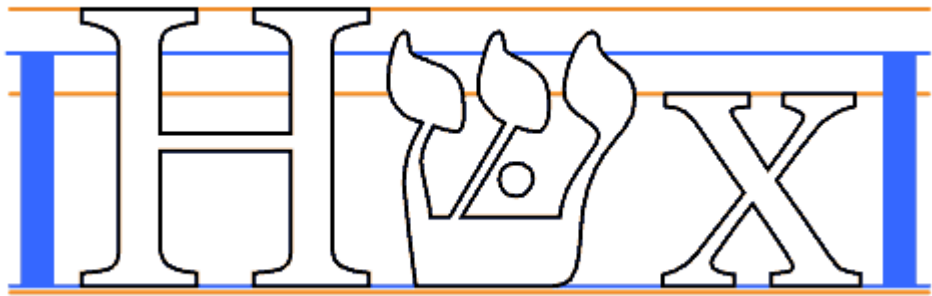
### 15.4.10 Altura de los signos en minúsculas

Esta medida es la coordenada y de la línea superior de las letras minúsculas sin acentos ni rasgos ascendentes en las escrituras latina, griega y cirílica, tomada desde la línea de base. Se toman las letras con terminaciones rectas, ignorándose toda zona de corrección óptica. Esto se emplea normalmente como la proporción entre las alturas de las mayúsculas y las minúsculas a manera de indicador para comparar familias de fuentes.



Este descriptor no es útil para fuentes que no contienen signos para estas escrituras. Como las alturas de las letras mayúsculas y minúsculas son a menudo expresadas como la proporción para comparar distintas fuentes, puede resultar útil para llevar a las minúsculas y mayúsculas al mismo valor que el de las escrituras de caja única

como el hebreo, donde para los textos latinos y hebreos mixtos, los caracteres hebreos son típicamente fijados a una altura intermedia entre las alturas de las mayúsculas y minúsculas de la fuente latina.



**15.4.11 Línea de base inferior**

Esta da la posición de la línea de base inferior en el cuadrado eme. La línea de base inferior se usa en las escrituras latina, griega y cirílica para la alineación, tal como la línea de base superior se usa en las escrituras derivadas del Sánscrito.

**15.4.12 Línea de base matemática**

Esta da la posición de la línea de base matemática en el cuadrado eme. La línea de base matemática es usada por los símbolos matemáticos para la alineación, tal como la línea de base inferior es usada por las escrituras latina, griega y cirílica.

**15.4.13 Máxima caja externa**

La máxima caja externa es el rectángulo más pequeño que encierra el molde que resulta si se ponen todos los signos de la fuente coincidentes unos con otros, y luego son trazados.

Si una fuente descargable dinámicamente ha sido generada como derivado de una fuente padre, la caja externa debe ser aquella de la fuente padre.

**15.4.14 Altura máxima sin acentos**

Esta medida, en el cuadrado eme, va desde la línea de base hasta el punto más alto alcanzado por cualquier carácter, excluyendo cualquier acento o signo diacrítico.



**15.4.15 Profundidad máxima sin acentos**

Esta medida, en el cuadrado eme, va desde la línea de base hasta el punto más bajo alcanzado por cualquier carácter, excluyendo cualquier acento o signo diacrítico.



## 15.4.16 Número Panose-1

*Panose-1* es una tecnología industrial estándar para la clasificación y equivalencia de las fuentes TrueType. El sistema PANOSE consiste en un conjunto de diez números que categorizan los atributos claves de un tipo latino, un procedimiento de clasificación para la creación de esos números y el programa validador que determina la fuente con la mejor equivalencia dentro de un conjunto de fuentes. El sistema *puede*, con modificaciones, también ser usado para el griego y el cirílico, pero no es apropiado para las escrituras de caja única e ideográficas (hebreo, armenio, árabe, chino/japonés/coreano).

## 15.4.17 Rango de caracteres ISO 10646

Este indica el repertorio de signos de la fuente, relativo a ISO 10646 (Unicode). Toda vez que este es exiguo (la mayoría de las fuentes no cubren por completo ISO 10646) este descriptor especifica bloques o rangos que tiene *cierto* alcance (no se promete ningún alcance completo) y se usa para eliminar fuentes inapropiadas (aquellas que no tendrán los signos requeridos). No indica que la fuente tiene positivamente los signos requeridos, sólo que vale la pena bajarla y mirarla. Ver [\[ISO10646\]](#) por información acerca de documentos útiles.

Este método es extensible a futuras asignaciones de caracteres en Unicode, sin cambios de sintaxis y sin invalidar el contenido existente.

Los formatos de fuente que no incluyen esta información, explícita o indirectamente, aún pueden usar esta característica, pero el valor debe ser proporcionado por el documento o por el autor de la hoja de estilo.

Hay otras clasificaciones para los sistemas de escritura, como el sistema Monotype (ver [\[MONOTYPE\]](#)) y un propuesto sistema ISO de escritura. Estos no son fácilmente extensibles.

Debido a esto, la clasificación de los repertorios de signos por el rango de caracteres de ISO 10646 que pueden ser representados con una fuente particular es usada en esta especificación. Este sistema es extensible para cubrir cualquier asignación futura.

## 15.4.18 Línea de base superior

Esta da la posición de la línea de base superior en el cuadrado eme. La línea de base superior se usa en las escrituras derivadas del sánscrito para la alineación, tal como la línea de base inferior se usa en las escrituras latina, griega y cirílica.

## 15.4.19 Ancho del asta vertical

Este es ancho del asta o trazo vertical (o casi vertical) de los signos. Esta información a menudo está sujeta a interpretación y puede no estar directamente accesible en algunos formatos de fuentes. La medición debe hacerse en el asta vertical *dominante* de la fuente, debido a que podría haber distintas agrupaciones de astas verticales (ej., una principal y una de menor peso como en la M o N mayúsculas).

## 15.4.20 Angulo del asta vertical

Este es el ángulo, en grados medidos en sentido contrario a las agujas del reloj desde la vertical, de las astas verticales dominantes de la fuente. El valor es negativo para las fuentes que se inclinan hacia la derecha, como hacen casi todas las fuentes *itálicas*. Este descriptor también puede especificarse para las fuentes oblicuas, inclinadas o manuscritas, y en general para cualquier fuente cuyas astas verticales no son precisamente verticales. Un valor distinto a cero no indica por sí mismo una fuente *itálica*.

# 15.5 Algoritmo de equivalencia de fuentes

Esta especificación extiende el algoritmo definido en la especificación CSS1. Este algoritmo reduce el algoritmo en la especificación CSS1 cuando las hojas de estilo del autor y el lector no contienen ninguna regla `@font-face`.

La equivalencia de los descriptores de tipos de fuentes debe hacerse cuidadosamente. Los descriptores son validados en un orden bien definido para asegurarse de que el resultado de este proceso de equivalencia sea tan consistente como sea posible a través de las AU (asumiendo que la misma biblioteca de fuentes y las descripciones de las fuentes son presentadas a cada una de ellas). Este algoritmo puede mejorarse, siempre que una implementación se comporte como si el algoritmo hubiera sido ejecutado exactamente.

1. La aplicación del usuario elabora (o accede a) una base de datos de descriptores de fuentes relevantes de todas las fuentes que la AU reconoce. Si hay dos fuentes con el mismo descriptor exacto, una de ellas es *ignorada*. La AU puede reconocer la fuente porque:
  - o se ha instalado localmente
  - o es declarada usando una regla `@font-face` en una de las hojas de estilo vinculadas o contenida en el documento actual
  - o es usada en la hoja de estilo predeterminada de la AU que conceptualmente existe en todas las AU y se considera que tiene todas las reglas `@font-face` para todas las fuentes que la AU usará en la presentación predeterminada, más las reglas `@font-face` para las cinco especiales familias genéricas de fuentes (ver 'font-family') definidas en CSS2
2. A un elemento dado y por cada carácter en ese elemento, la AU integra las propiedades de la fuente aplicables a ese elemento. Usando el juego completo de propiedades, la AU usa el descriptor 'font-family'

para elegir una familia de fuentes tentativa. De este modo, la equivalencia en el nombre de la familia será previa a la equivalencia en otros descriptores. Las propiedades restantes son probadas con la familia de acuerdo a los criterios de equivalencia especificados en cada descriptor. Si existen equivalencias para todas las propiedades restantes, entonces esa es la fuente equivalente para el elemento dado.

3. Si no hay ninguna fuente equivalente dentro del descriptor 'font-family' procesado en el paso 2, *las AU que implementan la equivalencia inteligente* pueden proceder a examinar otros descriptores tales como la altura de la x, el ancho de los signos y panose-1 para identificar una diferente familia de fuentes tentativa. Si existen equivalencias para todos los descriptores restantes, entonces esa es la fuente equivalente para el elemento dado. El descriptor 'font-family' reflejado en las propiedades CSS2 es la familia de fuentes requerida, no cualquier nombre que la fuente obtenida por equivalencia inteligente pueda tener. Las AU que no implementan la equivalencia inteligente se consideran fallidas en este paso.
4. Si no hay ninguna fuente equivalente dentro del descriptor 'font-family' procesado en el paso 3, *las AU que implementan la transmisión de fuentes* pueden proceder a examinar el descriptor 'src' de la fuente tentativa identificada en los pasos 2 o 3 para identificar un recurso en la red que esté disponible y del formato correcto. Si existen equivalencias para todos los descriptores restantes, entonces esa es la fuente equivalente para el elemento dado y la AU puede intentar transmitir este recurso de la fuente. La AU puede elegir detenerse en esta transmisión o puede elegir proceder al siguiente paso mientras la fuente es transmitida. Las AU que no implementan la transmisión de fuentes, o no están conectadas a una red, o cuando las preferencias del usuario han deshabilitado la transmisión de fuentes, o cuando el recurso solicitado no está disponible por cualquier razón, o cuando la fuente transmitida no puede ser usada por cualquier razón, se consideran fallidas en este paso.
5. Si no hay ninguna fuente equivalente dentro del descriptor 'font-family' procesado en el paso 3, *las AU que implementan síntesis de fuentes* pueden proceder a examinar otros descriptores como 'x-height', ancho de los signos y 'panose-1' para identificar una diferente familia de fuentes tentativa para la síntesis. Si existen equivalencias para todos los descriptores restantes, entonces esa es la fuente equivalente para el elemento dado y el sintetizado de la fuente fallida puede comenzar. La AU que no implementan la síntesis de fuentes se consideran fallidas en este paso.
6. Si todos los pasos 3, 4 y 5 son fallidos y hay un siguiente descriptor 'font-family' alternativo en el juego de fuentes, entonces se repite desde el paso 2 con el siguiente descriptor 'font-family' alternativo.
7. Si hay una fuente equivalente, pero no contiene signo(s) para el carácter(es) actual y si hay un siguiente descriptor 'font-family' alternativo en el juego de fuentes, entonces se repite desde el paso 2 con el siguiente descriptor 'font-family' alternativo. El descriptor 'unicode-range' puede utilizarse para sacar rápidamente de consideración aquellas fuentes que no tienen los signos correctos. Si el descriptor 'unicode-range' indica que una fuente contiene algunos signos en el rango correcto, puede ser examinado por la AU para ver si tiene ese en particular.
8. Si no hay ninguna fuente dentro de la familia seleccionada en 2, entonces usa el valor de 'font-family' heredado o dependiente de la AU y repite desde el paso 2, usando la mejor equivalencia que puede obtener dentro de esta fuente. Si un carácter en particular no puede ser mostrado usando esta fuente, la AU deberá indicar que un carácter no está mostrándose (por ejemplo, usando el signo para los 'caracteres ausentes').
9. Las AU que implementan el procesamiento progresivo y tienen pendiente la transmisión de una fuente pueden, una vez que la transmisión es exitosa, usar la fuente transmitida como una familia de fuentes. Si a la fuente transmitida le falta algún signo que la fuente progresiva temporaria contiene, la fuente transmitida no se usa para ese carácter y la fuente temporaria continúa siendo usada.

**Nota.** El algoritmo anterior puede optimizarse para evitar tener que volver a visitar las propiedades CSS2 por cada carácter.

Las reglas de equivalencia por descriptor anteriores (2) son las siguientes:

1. 'font-style' se intenta primero. 'Italic' se establecerá si hay una fuente en la base de datos de fuentes de la AU definida con la palabra clave de CSS 'italic' (preferida) u 'oblique'. Por otra parte, los valores deben equivaler exactamente o font-style será fallido.
2. 'font-variant' se intenta luego. 'Normal' corresponde con una fuente no designada como 'small-caps'; 'small-caps' coincide con (1) una fuente designada como 'small-caps', (2) una fuente en la cuál las mayúsculas pequeñas son sintetizadas o (3) una fuente donde todas las letras en minúsculas son reemplazadas por mayúsculas. Una fuente small-caps puede ser sintetizada escalando electrónicamente las letras en mayúsculas de una fuente normal.
3. 'font-weight' es analizado luego, nunca será fallido. (Ver 'font-weight' abajo.)
4. 'font-size' debe validarse dentro del margen de tolerancia dependiente de la AU. (Típicamente, los tamaños para las fuentes escalables son redondeados al pixel entero más cercano, mientras que la tolerancia para las fuentes de mapas de bits puede ser de hasta un 20%.) Los cálculos siguientes, ej., para valores 'em' en otras propiedades, son basados en el valor de 'font-size' que se usa, no el que está especificado.

### 15.5.1 Equivalencia de los valores de los pesos de las fuentes con los nombres de las fuentes

Los valores de la propiedad 'font-weight' se dan en una escala numérica en la cuál el valor '400' (o 'normal') corresponde al tipo 'normal' de esa familia. El nombre del peso asociado con ese tipo será típicamente *Book*, *Regular*, *Roman*, *Normal* o, a veces, *Medium*.

La asociación de otros pesos dentro de una familia con los valores numéricos de los pesos se propone sólo preservar el orden de los pesos dentro de esa familia. Las aplicaciones del usuario deben asignar los nombres a los valores de modo de preservar el orden visual; un tipo asignado a un valor no debe ser más liviano que un tipo asignado a valores más bajos. No hay ninguna seguridad de cómo una aplicación del usuario asignará los tipos de fuentes dentro de una familia a los valores de los pesos. Sin embargo, el siguiente análisis heurístico indica cómo se realiza la asignación en los casos típicos:

- Si la familia de fuentes ya usa una escala numérica con nueve valores (como hace por ej., *OpenType*), los pesos de la fuente deben ser asignados directamente.
- Si hay un tipo designado como *Medium* y otro designado *Book*, *Regular*, *Roman* o *Normal*, entonces *Medium* es normalmente asignado a '500'.
- La fuente denominada "Bold" corresponderá comúnmente al valor '700'.
- Si hay entonces menos de 9 pesos en la familia, el algoritmo predeterminado para completar los "agujeros" es como sigue. Si '500' no está asignado, será asignado a la misma fuente que '400'. Si alguno de los valores '600', '700', '800' o '900' permanecen sin asignar, es asignado al mismo tipo que la siguiente palabra clave más negra asignada, si hubiera, o la siguiente más blanca de otro modo. Si alguno de los valores '300', '200' o '100' permanecen sin asignar, es asignado al mismo tipo que la siguiente palabra clave más blanca asignada, si hubiera, o la siguiente más negra de otro modo.

No hay seguridad de que habrá una versión más negra para cada uno de los valores de 'font-weight'; por ejemplo, algunas fuentes pueden tener sólo una versión normal y una bold, otras puede tener ocho diferentes versiones de pesos.

Los siguientes dos ejemplos muestran asignaciones típicas.

Suponga cuatro variantes de peso en la familia "Rattlesnake", de la más blanca a la más negra: *Regular*, *Medium*, *Bold*, *Heavy*.

| Primer ejemplo de asignación de los pesos de la fuente |              |                          |
|--|--------------|--------------------------|
| Tipos disponibles                                      | Asignaciones | Completando los agujeros |
| "Rattlesnake Regular"                                  | 400          | 100, 200, 300            |
| "Rattlesnake Medium"                                   | 500          |                          |
| "Rattlesnake Bold"                                     | 700          | 600                      |
| "Rattlesnake Heavy"                                    | 800          | 900                      |

Suponga seis variantes en la familia "Ice Prawn": *Book*, *Medium*, *Bold*, *Heavy*, *Black*, *ExtraBlack*. Observe que en esta instancia la aplicación del usuario ha decidido *no* no asignar un valor numérico a la "ExtraBlack".

| Segundo ejemplo de asignación de los pesos de la fuente |              |                          |
|---|--------------|--------------------------|
| Tipos disponibles                                       | Asignaciones | Completando los agujeros |
| "Ice Prawn Book"  | 400          | 100, 200, 300            |
| "Ice Prawn Medium"                                      | 500          |                          |
| "Ice Prawn Bold"  | 700          | 600                      |
| "Ice Prawn Heavy"                                       | 800          |                          |
| "Ice Prawn Black"                                       | 900          |                          |
| "Ice Prawn ExtraBlack"                                  | (ninguno)    |                          |

### 15.5.2 Ejemplos de equivalencias de fuentes

El ejemplo siguiente define una fuente específica, la Alabama Italic. Se proporciona una descripción panose y un URI de origen para recuperar una fuente truetype de un servidor. Los descriptores font-weight y font-style son provistos para describir la fuente. La declaración también especifica que el peso coincidirá con cualquier instancia en el rango 300 a 500. La familia de fuentes es Alabama y el nombre de la fuente con su agregado es Alabama Italic.

```
@font-face {
  src: local("Alabama Italic"),
       url(http://www.fonts.org/A/alabama-italic) format("truetype");
  panose-1: 2 4 5 2 5 4 5 9 3 3;
  font-family: Alabama, serif;
  font-weight: 300, 400, 500;
  font-style: italic, oblique;
}
```

El siguiente ejemplo define una familia de fuentes. Un solo URI es proporcionado para la recuperación de los datos de la fuente. Este archivo de datos contendrá múltiples estilos y pesos de la fuente nombrada. Una vez que estas definiciones de @font-face han sido consideradas, los datos estarán en la memoria caché de la AU para aquellos tipos que usen el mismo URI.



```

@font-face {
  src: local("Helvetica Medium"),
       url(http://www.fonts.org/sans/Helvetica_family) format("truedoc");
  font-family: "Helvetica";
  font-style: normal
}
@font-face {
  src: local("Helvetica Oblique"),
       url("http://www.fonts.org/sans/Helvetica_family") format("truedoc");
  font-family: "Helvetica";
  font-style: oblique;
  slope: -18
}

```

El siguiente ejemplo agrupa tres fuentes físicas en una fuente virtual con cobertura extendida. En cada caso, el nombre de la fuente con sus agregados se da en el descriptor src para permitir que las versiones localmente instaladas sean usadas preferentemente si están disponibles. Una cuarta regla apunta a una fuente con la misma cobertura pero contenida en un único recurso.

```

@font-face {
  font-family: Excelsior;
  src: local("Excelsior Roman"), url("http://site/er")
format("intellifont");
  unicode-range: U+???; /* Latin-1 */
}
@font-face {
  font-family: Excelsior;
  src: local("Excelsior EastA Roman"), url("http://site/ear")
format("intellifont");
  unicode-range: U+100-220; /* Latin Extendido A y B */
}
@font-face {
  font-family: Excelsior;
  src: local("Excelsior Cyrillic Upright"), url("http://site/ecr")
format("intellifont");
  unicode-range: U+4???; /* Cirílico */
}
@font-face {
  font-family: Excelsior;
  src: url("http://site/excels") format("truedoc");
  unicode-range: U+??,U+100-220,U+4???;
}

```

Este siguiente ejemplo podría encontrarse en una hoja de estilo predeterminada de la AU. Implementa la familia genérica de fuentes CSS2 serif asignándola a una gran variedad de fuentes con serif que podrían existir en varias plataformas. No se da ninguna métrica debido a que ésta varía entre las posibles alternativas.

```

@font-face {
  src: local("Palatino"),
       local("Times New Roman"),
       local("New York"),
       local("Utopia"),
       url("http://somewhere/free/font");
  font-family: serif;
  font-weight: 100, 200, 300, 400, 500;
  font-style: normal;
  font-variant: normal;
  font-size: all
}

```



# 16 Texto

Las propiedades definidas en las siguientes secciones afectan la presentación visual de los caracteres, espacios, palabras y párrafos.

## 16.1 Sangría: la propiedad 'text-indent'

### 'text-indent'

- Valor: <medida> | <porcentaje> | inherit
- Inicial: 0
- Se aplica a: los elementos a nivel de bloque
- Se hereda: si
- Porcentajes: se refieren al ancho del bloque de contención
- Medios: visuales

Esta propiedad especifica la sangría de la primer línea de texto en un bloque. Más precisamente, especifica la sangría de la primer caja que fluye dentro de la primer [caja de línea](#) dentro del bloque. La caja es sangrada con respecto al límite izquierdo (o derecho, para las composiciones de derecha a izquierda) de la caja de línea. Las aplicaciones del usuario deben procesar este sangrado como espacio en blanco.

Los valores tienen los siguientes significados:

- <medida>  
El sangrado tiene una medida fija.
- <porcentaje>  
El sangrado es un porcentaje del ancho del bloque de contención.

El valor de 'text-indent' puede ser negativo, pero pueden haber limitaciones específicas de la implementación.

El siguiente ejemplo produce una sangría de '3em' en el texto.

```
P { text-indent: 3em }
```

## 16.2 Alineación: la propiedad 'text-align'

### 'text-align'

- Valor: left | right | center | justify | <cadena> | inherit
- Inicial: depende de la aplicación del usuario y el sentido de la escritura
- Se aplica a: los elementos a nivel de bloque
- Se hereda: si
- Porcentajes: N/A
- Medios: visuales

Esta propiedad describe cómo el contenido a nivel del línea de un bloque es alineado. Los valores tienen los siguientes significados:

- left, right, center y justify  
Texto alineado a la izquierda, derecha, centro y justificado, respectivamente.
- <cadena>  
Especifica una cadena con la cuál las celdas de una columna de una tabla se alinearán (ver la sección sobre [alineamiento horizontal en una columna](#) para detalles y un ejemplo). Este valor se aplica *solamente* a las celdas de una [tabla](#). Si se aplica a otros elementos, será tratado como 'left' o 'right', dependiendo de si 'direction' es 'ltr' o 'rtl', respectivamente.

Un bloque de texto es una pila de [cajas de líneas](#). En el caso de 'left', 'right' y 'center', esta propiedad especifica cómo las cajas a nivel de línea dentro de cada caja de línea se alinean con respecto a los lados izquierdo y derecho de la caja de línea; la alineación no es con respecto al [acceso visual](#). En el caso de 'justify', la AU pueden estirar las cajas a nivel de línea además de ajustar su posición. (Ver también 'letter-spacing' y 'word-spacing'.)

En este ejemplo, observe que mientras 'text-align' es heredado, todos los elementos a nivel de bloque dentro del elemento DIV con 'class=center' tendrán su contenido centrado.

```
DIV.center { text-align: center }
```

**Nota.** El real algoritmo de justificación usado depende de la aplicación del usuario y del lenguaje de escritura.

*Las aplicaciones del usuario con conformidad pueden interpretar el valor 'justify' como 'left' o 'right', dependiendo de si la dirección de escritura predeterminada del elemento es de izquierda a derecha o de derecha a izquierda, respectivamente.*

## 16.3 Decoración

### 16.3.1 Subrayado, súper-rayado, tachado y parpadeo:: la propiedad 'text-decoration'

**'text-decoration'**

- Valor: none | [ underline || overline || line-through || blink ] | inherit
- Inicial: none
- Se aplica a: todos los elementos
- Se hereda: no (ver texto)
- Porcentajes: N/A
- Medios: visuales

Esta propiedad describe las decoraciones que se agregan al texto de un elemento. Si la propiedad es especificada para un elemento [a nivel de bloque](#), afecta a todos los descendientes a nivel de línea del elemento. Si es especificada para (o afecta) un elemento [a nivel de línea](#), afecta a todas las cajas generadas por el elemento. Si el elemento no tiene ningún contenido o ningún contenido textual (ej., el elemento IMG en HTML), las aplicaciones del usuario deben [ignorar](#) esta propiedad.

Los valores tienen los siguientes significados:

- none**  
No produce ninguna decoración en el texto.
- underline**  
Cada línea de texto es subrayada.
- overline**  
Cada línea de texto tiene una línea encima.
- line-through**  
Cada línea de texto tiene una línea por el medio.
- blink**  
El texto parpadea (alterna entre visible e invisible). [Las aplicaciones del usuario con conformidad](#) no están obligadas a soportar este valor.

El(los) color(es) requerido por la decoración del texto deberían derivar del valor de la propiedad 'color'.

Esta propiedad no es heredada, pero las cajas descendientes de una caja de bloque deberían tener en su formato la misma decoración (ej., deberían estar todas subrayadas). El color de las decoraciones debería permanecer igual aunque los elementos descendientes tuvieran diferente valores de 'color'.

En el siguiente ejemplo para HTML, el contenido textual de todos los elementos A que actúan como hipervínculos estarán subrayados:

```
A[href] { text-decoration: underline }
```

### 16.3.2 Texto con sombra: la propiedad 'text-shadow'

**'text-shadow'**

- Valor: none | [<color> || <medida> <medida> <medida>? ,]\* [<color> || <medida> <medida> <medida>?] | inherit
- Inicial: none
- Se aplica a: todos los elementos
- Se hereda: no (ver texto)
- Porcentajes: N/A
- Medios: visuales

Esta propiedad acepta una lista separada por comas de efectos de sombra para ser aplicados al texto del elemento. Los efectos de sombra son aplicados en el orden especificado y por eso pueden cubrirse unos a otros, pero nunca cubrirán al propio texto. Los efectos de sombra no alteran el tamaño de una caja, pero pueden extenderse más allá de sus límites. El [nivel en la pila](#) de los efectos de sombra es el mismo que el del elemento mismo.

Cada efecto de sombra debe especificar un desplazamiento de la sombra y puede especificar opcionalmente un radio de desenfoque y un color para la sombra.

Un desplazamiento de la sombra es especificado con dos valores de <medida> que indican la distancia con el texto. El primer valor especifica la distancia horizontal a la derecha del texto. Un valor horizontal negativo coloca la sombra a la izquierda del texto. El segundo valor especifica la distancia vertical hacia abajo del texto. Un valor vertical negativo coloca la sombra por sobre el texto.

Un radio de desenfoque puede opcionalmente especificarse después del desplazamiento de la sombra. El radio de desenfoque es el valor de la medida que indica los límites del efecto de desenfoque. El algoritmo exacto para computar el efecto de desenfoque no es especificado.

Un valor de color puede especificarse opcionalmente antes o después de los valores de medida del efecto de sombra. El valor del color será usado como la base del efecto de sombra. Si no se especifica ningún color, el valor de la propiedad 'color' será usado en su lugar.

Las sombras del texto pueden ser usadas con los pseudo-elementos `:first-letter` y `:first-line`.

El ejemplo de abajo pondrá una sombra del texto a la derecha y abajo del texto del elemento. Como no se ha especificado ningún color, la sombra tendrá el mismo color del elemento, y como no se ha especificado ningún radio de desenfoque, la sombra del texto no será desenfocada:

```
H1 { text-shadow: 0.2em 0.2em }
```

El siguiente ejemplo coloca una sombra a la derecha y abajo del texto de un elemento. La sombra tendrá un radio de desenfoque de 5px y será roja.

```
H2 { text-shadow: 3px 3px 5px red }
```

El siguiente ejemplo especifica una lista de efectos de sombra. La primer sombra estará a la derecha y abajo del texto del elemento y será roja sin desenfoque. La segunda sombra cubrirá al primer efecto de sombra, y será amarillo, desenfocado, y colocado a la izquierda y abajo del texto. El tercer efecto de sombra será colocado a la derecha y abajo del texto. Como no se ha especificado ningún color de sombra para el tercer efecto, el valor de la propiedad 'color' del elemento será usado:

```
H2 { text-shadow: 3px 3px red, yellow -3px 3px 2px, 3px -3px }
```

Considere este ejemplo:

```
SPAN.glow {
  background: white;
  color: white;
  text-shadow: black 0px 0px 5px;
}
```

Aquí, las propiedades 'background' y 'color' tienen el mismo valor y la propiedad 'text-shadow' se usa para crear un efecto de "eclipse solar":



**Nota.** Esta propiedad no está definida en CSS1. Algunos efectos de sombra (como el del último ejemplo) pueden resultar en un texto invisible en las AU que solamente soporten CSS1.

## 16.4 Espaciado de letras y palabras: las propiedades 'letter-spacing' y 'word-spacing'

### 'letter-spacing'

|              |                             |
|--------------|-----------------------------|
| Valor:       | normal   <medida>   inherit |
| Inicial:     | normal                      |
| Se aplica a: | todos los elementos         |
| Se hereda:   | si                          |
| Porcentajes: | N/A                         |
| Medios:      | visuales                    |

Esta propiedad especifica el comportamiento de los espacios entre los caracteres del texto. Los valores tienen los siguientes significados:

#### normal

El espaciado es el normal de la fuente actual. Este valor permite a las aplicaciones del usuario alterar el espacio entre los caracteres para justificar el texto.

#### <medida>

Este valor indica el espacio entre caracteres *además* del espacio predeterminado entre caracteres. Los valores pueden ser negativos, pero pueden existir limitaciones específicas de la implementación. Las aplicaciones del usuario pueden no aumentar o disminuir más el espacio entre caracteres para poder justificar el texto.

Los algoritmos para el espaciado de caracteres dependen de la aplicación del usuario. El espaciado entre caracteres también puede ser influenciado por la justificación (ver la propiedad 'text-align').

En este ejemplo, el espacio entre caracteres en los elementos BLOCKQUOTE es incrementado en '0.1em'.

```
BLOCKQUOTE { letter-spacing: 0.1em }
```

En el siguiente ejemplo, no se permite a la aplicación del usuario alterar el interletrado:

```
BLOCKQUOTE { letter-spacing: 0cm } /* Igual a '0' */
```

Cuando el espacio resultante entre dos caracteres no es el mismo que el espacio predeterminado, las aplicaciones del usuario no deberían usar ligaduras.

Las [aplicaciones del usuario con conformidad](#) pueden considerar al valor de la propiedad 'letter-spacing' como 'normal'.

| 'word-spacing' |                             |
|----------------|-----------------------------|
| Valor:         | normal   <medida>   inherit |
| Inicial:       | normal                      |
| Se aplica a:   | todos los elementos         |
| Se hereda:     | si                          |
| Porcentajes:   | N/A                         |
| Medios:        | visuales                    |

Esta propiedad especifica el comportamiento de los espacios entre palabras. Los valores tienen los siguientes significados:

- normal**  
El espaciado normal entre palabras, como está definido por la fuente actual y/o la AU.
- <medida>**  
Este valor indica el espacio entre palabras *además* del espacio enre palabras predeterminado. Los valores pueden ser negativos, pero pueden existir limitaciones específicas de la implementación.

Los algoritmos para el espaciado entre palabras dependen de la aplicación del usuario. Los espacios entre palabras también son influenciados por la justificación (ver la propiedad 'text-align').

En este ejemplo, el espacio entre cada palabra en los elementos H1 es incrementado en '1em'.

```
H1 { word-spacing: 1em }
```

Las [aplicaciones del usuario con conformidad](#) pueden considerar al valor de la propiedad 'word-spacing' como 'normal'.

## 16.5 Cambio de minúsculas y mayúsculas: la propiedad 'text-transform'

| 'text-transform' |   |
|------------------|---|
| Valor:           | capitalize   uppercase   lowercase   none   inherit |
| Inicial:         | none  |
| Se aplica a:     | todos los elementos                                 |
| Se hereda:       | si  |
| Porcentajes:     | N/A   |
| Medios:          | visuales  |

Esta propiedad controla el efecto de los cambios entre mayúsculas y minúsculas en el texto de un elemento. Los valores tienen los siguientes significados:

- capitalize**  
Pone al primer carácter de cada palabra en mayúscula.
- uppercase**  
Pone a todos los caracteres de cada palabra en mayúsculas.
- lowercase**  
Pone a todos los caracteres de cada palabra en minúsculas.
- none**  
Ningún efecto de cambio entre mayúsculas y minúsculas.

La transformación real en cada caso depende del lenguaje escrito. Ver RFC 2070 ([RFC2070](#)) por las modos de encontrar el lenguaje de un elemento.

Las [aplicaciones del usuario con conformidad](#) pueden considerar al valor de 'text-transform' como 'none' para los caracteres que no son parte del repertorio de Latin-1 y para los elementos en lenguajes para los cuales la transformación es distinta de la especificada por la tabla de conversión de mayúsculas y minúsculas de ISO 10646 ([[ISO10646](#)]).

En este ejemplo, todo el texto de un elemento H1 es transformado en mayúsculas.

```
H1 { text-transform: uppercase }
```

## 16.6 Espacios en blanco: la propiedad 'white-space'

### 'white-space'

|              |                                 |
|--------------|---------------------------------|
| Valor:       | normal   pre   nowrap   inherit |
| Inicial:     | normal                          |
| Se aplica a: | elementos a nivel de bloque     |
| Se hereda:   | si                              |
| Porcentajes: | N/A                             |
| Medios:      | visuales                        |

Esta propiedad declara cómo son tratados los [espacios en blanco](#) dentro del elemento. Los valores tienen los siguientes significados:

#### normal

Este valor le indica a las aplicaciones del usuario cerrar las secuencias de espacios en blanco, y cortar las líneas como sea necesario para llenar las cajas de línea. Pueden crearse cortes de línea adicionales por la aparición de "\A" en el contenido generado (ej., por el elemento BR en HTML).

#### pre

Este valor impide a las aplicaciones del usuario cerrar las secuencias de espacios en blanco. Las líneas son cortadas solamente en los avances de línea de la fuente o por la aparición de "\A" en el contenido generado.

#### nowrap

Este valor cierra los espacios en blanco como en 'normal', pero suprime los cortes de línea dentro del texto excepto aquellos creados por "\A" en el elemento generado (ej., por el elemento BR en HTML).

Los siguientes ejemplos muestran qué comportamiento de los [espacios en blanco](#) se espera en los elementos PRE y P, y el atributo "nowrap" en HTML.

```
PRE      { white-space: pre }
P        { white-space: normal }
TD[nowrap] { white-space: nowrap }
```

Las [aplicaciones del usuario con conformidad](#) pueden [ignorar](#) la propiedad 'white-space' en las hojas de estilo del autor o del usuario pero deben especificar un valor para la misma en la hoja de estilo predeterminada.

# 17 Tablas

## 17.1 Introducción a las tablas

Las tablas representan relaciones entre datos. Los autores especifican estas relaciones en el [lenguaje del documento](#) y especifican su *presentación* en CSS, de dos maneras: visualmente y auditivamente.

Los autores pueden especificar el formato visual de una tabla como una grilla rectangular de celdas. Las filas y columnas de celdas pueden organizarse en grupos de filas y en grupos de columnas. Las filas, columnas, grupos de filas, grupos de columnas y las celdas pueden tener bordes dibujados alrededor de ellos (hay dos modelos de bordes en CSS2). Los autores pueden alinear los datos verticalmente u horizontalmente dentro de la celda y alinear los datos en todas las celdas de una fila o columna.

Los autores también pueden especificar el procesamiento auditivo de una tabla; cómo serán leídos los títulos y los datos. En el lenguaje del documento, los autores pueden etiquetar las celdas y los grupos de celdas de manera que cuando son procesadas auditivamente, los títulos de las celdas son expresados antes que los datos de la celda. En realidad, esto "serializa" la tabla: los usuarios que recorren la tabla auditivamente escuchan una secuencia de títulos seguidos por los datos.

Aquí hay una tabla simple descrita en HTML 4.0 de tres filas y tres columnas:

```
<TABLE>
<CAPTION>Esta es una tabla simple de 3x3</CAPTION>
<TR id="fila1">
  <TH>Título 1      <TD>Celda 1      <TD>Celda 2
<TR id="fila2">
  <TH>Título 2      <TD>Celda 3      <TD>Celda 4
<TR id="fila3">
  <TH>Título 3      <TD>Celda 5      <TD>Celda 6
</TABLE>
```

Este código crea una tabla (el elemento TABLE), tres filas (los elementos TR), tres celdas de título (los elementos TH) y seis celdas de datos (los elementos TD). Observe que las tres columnas de este ejemplo están especificadas implícitamente: hay tantas columnas en la tabla como se requiere para las celdas de los títulos y los datos.

La siguiente regla CSS centra horizontalmente el texto en las celdas de títulos y presenta los datos con una fuente en negritas:

```
TH { text-align: center; font-weight: bold }
```

Las siguientes reglas alínean el texto de las celdas de títulos por su línea de base y centra verticalmente el texto en cada celda de datos:

```
TH { vertical-align: baseline }
TD { vertical-align: middle }
```

Las reglas siguientes especifican que la fila superior será rodeada con un borde sólido azul de 3px y cada una de las otras filas serán rodeadas con un borde sólido negro de 1px:

```
TABLE { border-collapse: collapse }
TR#fila1 { border-top: 3px solid blue }
TR#fila2 { border-top: 1px solid black }
TR#fila3 { border-top: 1px solid black }
```

Observe, sin embargo, que los bordes alrededor de las filas se superponen cuando las filas se juntan. ¿De qué color (negro o azul) y grosor (1px o 3px) será el borde entre fila1 y fila2? Discutiremos esto en la sección sobre [resolución de los conflictos de borde](#).

La siguiente regla coloca el encabezado de la tabla encima de la tabla:

```
CAPTION { caption-side: top }
```

Finalmente, la siguiente regla especifica que, cuando sea procesada auditivamente, cada fila de datos se expresará con un "Título, Datos, Datos":

```
TH { speak-header: once }
```

Por ejemplo, la primer fila sería leída como "Título1 Celda1 Celda2". Por otra parte, con la siguiente regla:

```
TH { speak-header: always }
```

sería expresada como "Título1 Celda1 Título1 Celda2".

El ejemplo precedente muestra cómo CSS trabaja con los elementos de HTML 4.0; en HTML 4.0, la semántica de los distintos elementos de la tabla (TABLE, CAPTION, THEAD, TBODY, TFOOT, COL, COLGROUP, TH y TD) están bien definidos. En otros lenguajes de documento (como las aplicaciones XML) pueden no existir elementos predefinidos para la tabla. Por eso, CSS2 permite a los autores "asignar" a los elementos del lenguaje del documento, elementos de la tabla a través de la propiedad 'display'. Por ejemplo, la siguiente regla hace que el elemento FOO actúe como un elemento HTML TABLE y el elemento BAR actúe como un elemento CAPTION:

```
FOO { display : table }
BAR { display : table-caption }
```

Discutiremos los diversos elementos de la tabla en la sección siguiente. En esta especificación, el término *elemento de la tabla* se refiere a cualquier elemento involucrado en la creación de una tabla. Un elemento "interno" de la tabla es uno que produce una fila, un grupo de filas, una columna, un grupo de columnas o una celda.

## 17.2 El modelo de tabla CSS

El modelo de tabla CSS está basado en el modelo de tabla de HTML 4.0, en el que la estructura de una tabla parangona estrechamente la composición visual de la tabla. En este modelo, una tabla consiste de un encabezamiento opcional y cualquier número de filas y columnas. Se dice que el modelo de tabla es "principalmente de filas" porque los autores especifican filas, no columnas, explícitamente en el lenguaje del documento. Las columnas se derivan una vez que las filas han sido especificadas (la primer celda de cada fila pertenece a la primer columna, la segunda a la segunda columna, etc.). Las filas y columnas pueden se agrupadas estructuralmente y este agrupación se refleja en la presentación (ej., puede dibujarse un borde alrededor de un grupo de filas).

Así, el modelo de tabla consiste en tablas, encabezados, filas, grupos de filas, columnas, grupos de columnas y celdas.

El modelo CSS no requiere que el [lenguaje del documento](#) incluya elementos que correspondan a cada uno estos componentes. Para lenguajes de documento (como las aplicaciones XML) que no tienen los elementos de la tabla predefinidos, los autores pueden asignar a los elementos del lenguaje de documento los elementos de la tabla; esto se hace con la propiedad 'display'. Los siguientes valores para 'display' asignan la semántica de la tabla a un elemento arbitrario:

**table** (En HTML: TABLE)

Especifica que un elemento define una tabla [a nivel de bloque](#): es un bloque rectangular que participa de un [contexto de formato de bloque](#).

**inline-table** (En HTML: TABLE)

Especifica que un elemento define una tabla [a nivel de línea](#): es un bloque rectangular que participa de un [contexto de formato de línea](#)).

**table-row** (En HTML: TR)

Especifica que un elemento es una fila de celdas.

**table-row-group** (En HTML: TBODY)

Especifica que un elemento agrupa una o más filas.

**table-header-group** (En HTML: THEAD)

Como 'table-row-group', pero para el formato visual, el grupo de filas es siempre mostrado antes que todas las otras filas y grupos de filas y después de los encabezados superiores. Las aplicaciones del usuario que imprimen pueden repetir las filas de encabezado de página en cada página ocupada por una tabla.

**table-footer-group** (En HTML: TFOOT)

Como 'table-row-group', pero para el formato visual, el grupo de filas es siempre mostrado después que todas las otras filas y grupos de filas y antes de los encabezamientos inferiores. Las aplicaciones del usuario que imprimen pueden repetir las filas de pie de página en cada página ocupada por una tabla.

**table-column** (En HTML: COL)

Especifica que un elemento describe una columna de celdas.

**table-column-group** (En HTML: COLGROUP)

Especifica que un elemento agrupa una o más columnas.

**table-cell** (En HTML: TD, TH)

Especifica que un elemento representa una celda de la tabla.

**table-caption** (En HTML: CAPTION)

Especifica un encabezado para la tabla.

Los elementos con 'display' definido como 'table-column' o 'table-column-group' no son procesados (exactamente como si tuvieran 'display: none'), pero son útiles porque pueden tener atributos que inducen a cierto estilo para las columnas que representan.



La [hoja de estilo predeterminada para HTML 4.0](#) en el apéndice, ilustra el uso de estos valores en HTML 4.0:

```
TABLE      { display: table }
TR         { display: table-row }
THEAD      { display: table-header-group }
TBODY      { display: table-row-group }
TFOOT      { display: table-footer-group }
COL        { display: table-column }
COLGROUP   { display: table-column-group }
TD, TH     { display: table-cell }
CAPTION    { display: table-caption }
```

Las aplicaciones del usuario pueden [ignorar](#) estos valores para la propiedad 'display' en documentos HTML, ya que los autores no deben alterar el comportamiento esperado de un elemento.

### 17.2.1 Objetos anónimos de la tabla

Otros lenguajes de documento fuera de HTML pueden no contener todos los elementos del modelo de tabla CSS2. En este caso, los elementos "faltantes" deben asumirse para que el modelo de tabla funcione. Los elementos faltantes generan objetos [anónimos](#) (ej., cajas anónimas en la composición visual de la tabla) de acuerdo con las siguientes reglas:

1. Cualquier elemento de la tabla generará automáticamente, alrededor de sí mismo, los objetos anónimos de tabla necesarios, que consisten en tres objetos anidados correspondientes a un elemento 'table'/'inline-table', un elemento 'table-row' y un elemento 'table-cell'.
2. Si el padre P de un elemento 'table-cell' T no es un elemento 'table-row', un objeto correspondiente a 'table-row' será generado entre P y T. Este objeto abarcará todos los hermanos 'table-cell' consecutivos (en la estructura del documento) de T.
3. Si el padre P de un elemento 'table-row' T no es un elemento 'table', 'inline-table' o 'table-row-group', un objeto correspondiente al elemento 'table' será generado entre P y T. Este objeto abarcará todos los hermanos consecutivos (en la estructura del documento) de T que requieran un padre 'table': 'table-row', 'table-row-group', 'table-header-group', 'table-footer-group', 'table-column', 'table-column-group' y 'caption'.
4. Si el padre P de un elemento 'table-row-group' (o 'table-header-group' o 'table-footer-group') T no es un elemento 'table' o 'inline-table', un objeto correspondiente a un elemento 'table' será generado entre P y T. Este objeto abarcará todos los hermanos consecutivos (en la estructura del documento) de T que requieran un padre 'table': 'table-row', 'table-row-group', 'table-header-group', 'table-footer-group', 'table-column', 'table-column-group' y 'caption'.
5. Si un hijo T de un elemento 'table-row' P no es un elemento 'table-cell', un objeto correspondiente a un elemento 'table-cell' será generado entre P y T. Este objeto abarcará todos los hermanos consecutivos de T que no sean elementos 'table-cell'.

En este ejemplo XML, un elemento 'table' se asume para contener al elemento HBOX:

```
<HBOX>
  <VBOX>George</VBOX>
  <VBOX>4287</VBOX>
  <VBOX>1998</VBOX>
</HBOX>
```

porque la hoja de estilo asociada es:

```
HBOX { display: table-row }
VBOX { display: table-cell }
```

En este ejemplo, tres elementos 'table-cell' se asumen para contener el texto de cada FILA. Observe que el texto es además encapsulado en cajas de línea anónimas, como se explica en el [modelo de formato visual](#):

```
<PILA>
  <FILA>Esta es la fila <D>superior</D>.</FILA>
  <FILA>Esta es la fila <D>media</D>.</FILA>
  <FILA>Esta es la fila <D>inferior</D>.</FILA>
</PILA>
```

La hoja de estilo es:

```
PILA { display: inline-table }
FILA  { display: table-row }
D     { display: inline; font-weight: bolder }
```

No se exige a las aplicaciones del usuario HTML que creen objetos anónimos según las reglas anteriores.

## 17.3 Selectores de columnas

Las celdas de la tabla pueden pertenecer a dos contextos: filas y columnas. Sin embargo, en el documento fuente las celdas son descendientes de las filas, nunca de las columnas. No obstante, algunos aspectos de las celdas pueden ser influenciados poniendo propiedades a las columnas.

Las siguientes propiedades se aplican a los elementos columna y grupo de columnas:

- 'border'  
Las distintas propiedades de borde se aplican a las columnas sólo si 'border-collapse' es definido como 'collapse' en el elemento tabla. En ese caso, los bordes definidos para las columnas y grupos de columnas son introducidos al [algoritmo de resolución de conflictos](#) que elige el estilo de borde para cada lado de la celda.
- 'background'  
Las propiedades background determinan el fondo de las celdas en la columna, pero sólo si tanto la celda como la fila tienen fondos transparentes. Ver "[Capas y transparencia de la tabla](#)".
- 'width'  
La propiedad 'width' determina el ancho mínimo de la columna.
- 'visibility'  
Si en una columna, 'visibility' es puesta como 'collapse' ninguna de las celdas de la columna es procesada, y las celdas que abarcan otras columnas son recortadas. Además, el ancho de la tabla es disminuido en el ancho de la columna que se hubiera quitado. Ver "[Efectos dinámicos](#)" abajo. Otros valores para 'visibility' no tienen ningún efecto.

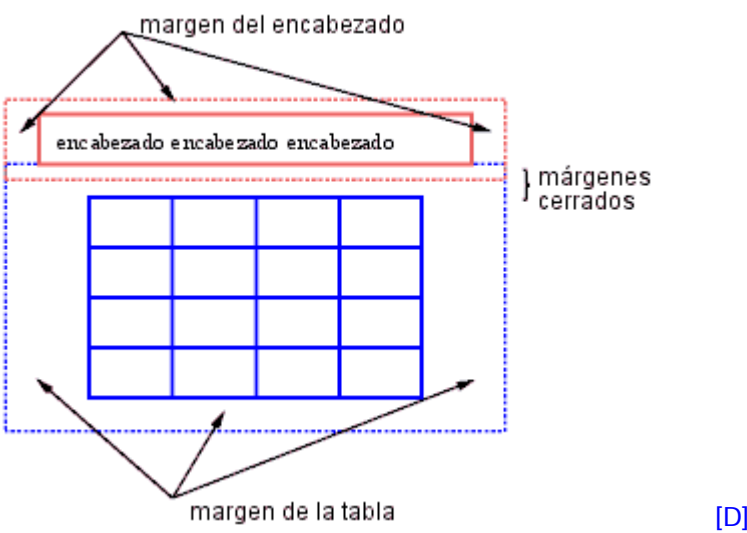
Aquí hay algunos ejemplos de reglas de estilo que ponen propiedades a las columnas. Las primeras dos reglas juntas implementan el atributo "rules" de HTML 4.0 con un valor de "cols". La tercer regla hace que la columna "totales" sea azul, las dos reglas finales muestran cómo hacer una columna de tamaño fijo, usando el [algoritmo de estructura fija](#).

```
COL { border-style: none solid }
TABLE { border-style: hidden }
COL.totales { background: blue }
TABLE { table-layout: fixed }
COL.totales { width: 5em }
```

## 17.4 Las tablas en el modelo de formato visual

En los términos del [modelo de formato visual](#), una tabla debe funcionar como un elemento [a nivel de bloque](#) o un elemento [a nivel de línea](#) reemplazado. Las tablas tienen contenido, relleno, bordes y márgenes.

En ambos casos, el elemento tabla genera una caja [anónima](#) que contiene a la propia caja de la tabla y a la caja del encabezado (si está presente). Las cajas de la tabla y el encabezado retienen sus propias áreas de contenido, relleno, márgenes y bordes, y las dimensiones de la caja rectangular anónima son las mínimas requeridas para contener a ambas. Los márgenes verticales se cierran donde la caja de la tabla y la caja del encabezado se tocan. Cualquier reposicionamiento de la tabla debe mover la caja anónima entera, no sólo la caja de la tabla, de modo que el encabezado siga a la tabla.



*Diagrama de una tabla con un encabezado encima de ella; el margen inferior del encabezado se cierra con el margen superior de la tabla.*

## 17.4.1 Posición y alineación del encabezado

### 'caption-side'

|              |                                       |
|--------------|---------------------------------------|
| Valor:       | top   bottom   left   right   inherit |
| Inicial:     | top                                   |
| Se aplica a: | los elementos 'table-caption'         |
| Se hereda:   | si                                    |
| Porcentajes: | N/A                                   |
| Medios:      | visuales                              |

Esta propiedad especifica la posición de la caja del encabezado con respecto a la caja de la tabla. Los valores tienen los siguientes significados:

**top**  
Coloca la caja del encabezado encima de la caja de la tabla.

**bottom**  
Coloca la caja del encabezado abajo de la caja de la tabla.

**left**  
Coloca la caja del encabezado a la izquierda de la caja de la tabla.

**right**  
Coloca la caja del encabezado a la derecha de la caja de la tabla.

Los encabezados encima o debajo del elemento 'table' son estructurados mucho más como si fueran un elemento de bloque antes o después de la tabla, excepto que (1) heredan las propiedades heredables de la tabla, y (2) no se consideran como una caja de bloque para el propósito de ningún elemento 'compact' o 'run-in' que pueda preceder a la tabla.

Un encabezado que está arriba o debajo de una caja de tabla también se comporta como una caja de bloque para el cálculo del ancho; el ancho es computado con respecto al ancho del bloque de contención de la caja de la tabla.

Para un encabezado que está a la izquierda o a la derecha de la caja de la tabla, por otro lado, un valor distinto a 'auto' en 'width' pone el ancho explícitamente, pero 'auto' le dice a la aplicación del usuario que elija un "ancho razonable". Este puede variar entre "la caja más angosta posible" y "una sola línea", de modo que recomendamos que los usuarios no especifiquen 'auto' para el ancho de los encabezados sobre la izquierda o derecha.

Para alinear horizontalmente el contenido del encabezado dentro de la caja del encabezado, use la propiedad 'text-align'. Para la alineación vertical de una caja de encabezado sobre la izquierda o derecha con respecto a la caja de la tabla, use la propiedad 'vertical-align'. Los únicos valores significativos en este caso son 'top', 'middle' y 'bottom'. Todos los demás valores son tratados como 'top'.

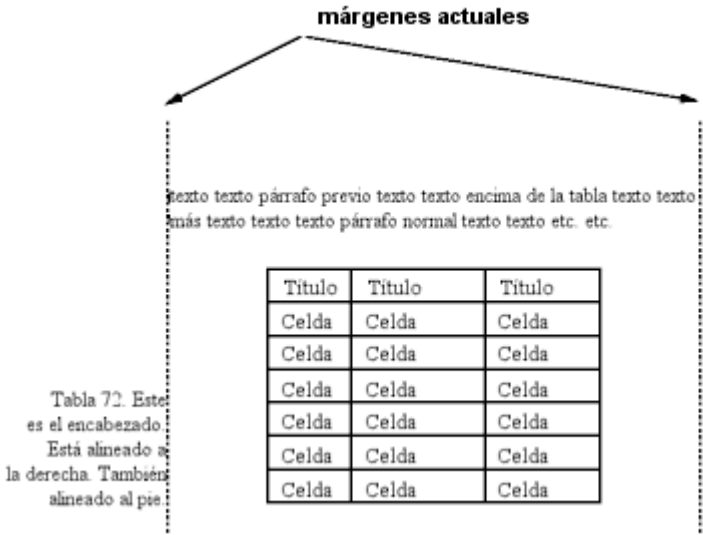
En este ejemplo, la propiedad 'caption-side' coloca los encabezados debajo de las tablas. El encabezado será tan ancho como el padre de la tabla y el texto del encabezado estará justificado a la izquierda.

```
CAPTION { caption-side: bottom;
           width: auto;
           text-align: left }
```

El siguiente ejemplo muestra como poner un encabezado en el margen izquierdo. La tabla en sí misma está centrada, definiendo sus márgenes izquierdo y derecho como 'auto', y la caja global con la tabla y el encabezado es desplazada dentro del margen izquierdo en la misma medida que el ancho del encabezado.

```
BODY {
    margin-left: 8em
}
TABLE {
    margin-left: auto;
    margin-right: auto
}
CAPTION {
    caption-side: left;
    margin-left: -8em;
    width: 8em;
    text-align: right;
    vertical-align: bottom
}
```

Asumiendo que el ancho de la tabla es menor que el ancho disponible, el formato será similar a esto:



[D]

Diagrama mostrando una tabla centrada con el encabezado extendido dentro del margen izquierdo, como resultado de una propiedad 'margin-left' negativa.

## 17.5 Composición visual del contenido de las tablas

Como otros elementos del [lenguaje del documento](#), los elementos internos de la tabla generan [cajas](#) rectangulares con contenido, relleno y bordes. No tienen márgenes, sin embargo.

La composición visual de estas cajas está gobernada por una grilla rectangular e irregular de filas y columnas. Cada caja ocupa un número entero de celdas de la grilla, determinado de acuerdo a las siguientes reglas. Estas reglas no se aplican a HTML 4.0 o a versiones más antiguas de HTML; HTML impone sus propias limitaciones al alcance de las filas y columnas.

1. Cada caja de fila ocupa una fila de celdas de la grilla. Juntas, las cajas de filas llenan la tabla desde arriba hacia abajo en el orden en que aparecen en el documento fuente (es decir, la tabla ocupa exactamente tantas filas de la grilla como elementos fila existen).
2. Un grupo de filas ocupa tantas celdas de la grilla como las filas que contiene.
3. Una caja de columna ocupa una o más columnas de celdas de la grilla. Las cajas de columnas son colocadas una al lado de otra en el orden en que aparecen. La primer caja de columna puede estar a la izquierda o a la derecha, depende del valor de la propiedad 'direction' de la tabla.
4. Una caja de un grupo de columnas ocupan tantas celdas de la grilla como las columnas que contiene.
5. Las celdas pueden abarcar varias filas o columnas. (Aunque CSS2 no define como se determina el número de filas y columnas abarcadas, una aplicación del usuario puede tener un conocimiento especial acerca del documento fuente; una versión futura de CSS puede proporcionar una manera de expresar este conocimiento en la sintaxis de CSS.) Cada celda es, en consecuencia, una caja rectangular, de una o más celdas de la grilla de ancho y de alto. La fila superior de este rectángulo es la fila especificada por el padre de la celda. El rectángulo debe estar tan a la izquierda como sea posible, pero no puede encimarse con ninguna otra caja de celda, y debe estar a la derecha de todas las celdas de la misma fila que aparecen antes en el documento fuente. (Esta restricción se mantiene mientras la propiedad 'direction' de la tabla sea 'ltr'; si 'direction' es 'rtl', intercambie "izquierda" y "derecha" en la oración anterior.)
6. Una caja de celda no puede extenderse más allá de la caja de la última fila o grupo de filas de una tabla; las aplicaciones del usuario deben acortarla hasta que se ajuste.

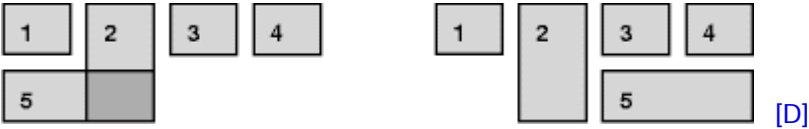
**Nota.** Las celdas de la tabla pueden ser posicionadas relativa y absolutamente, pero no se recomienda: el posicionamiento y la flotación quitan a una caja del flujo, afectando la alineación de la tabla.

Aquí hay dos ejemplos. El primero se supone que aparece en un documento HTML:

```
<TABLE>
<TR><TD>1 <TD rowspan="2">2 <TD>3 <TD>4
<TR><TD colspan="2">5
</TABLE>

<TABLE>
<ROW><CELL>1 <CELL rowspan="2">2 <CELL>3 <CELL>4
<ROW><CELL colspan="2">5
</TABLE>
```

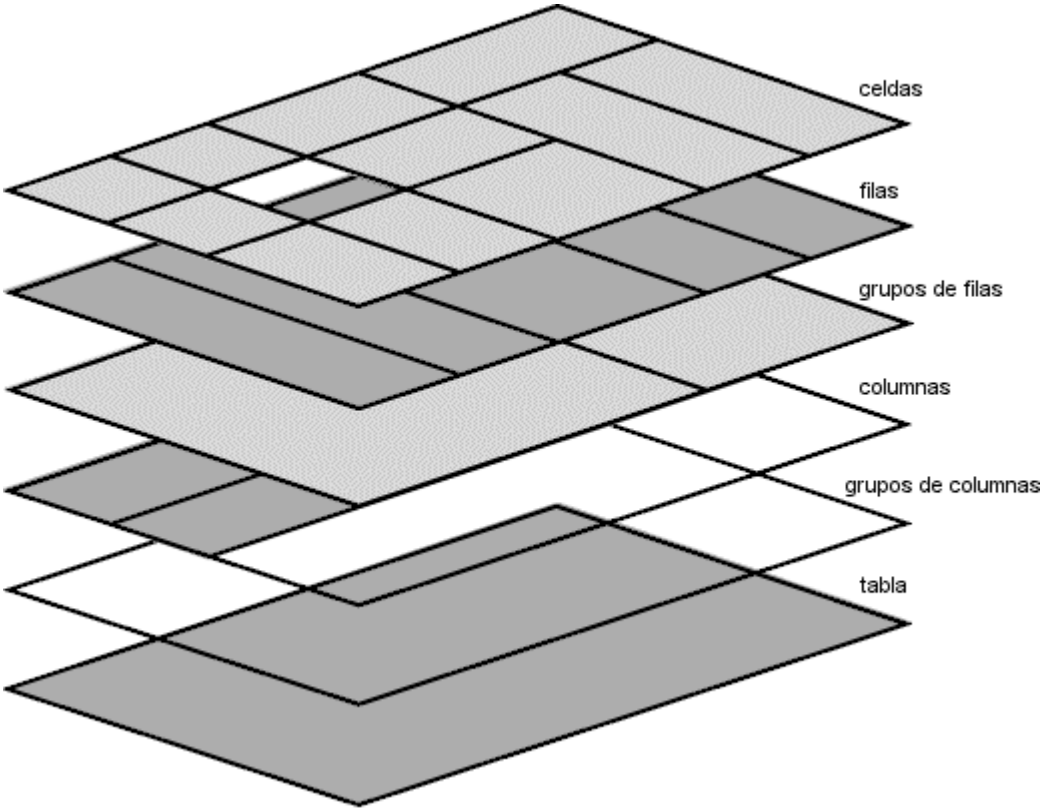
La segunda tabla es estructurada como en la figura de la derecha. Sin embargo, el procesamiento de las tablas en HTML está explícitamente indefinido por HTML, y CSS no trata de definirlo. Las aplicaciones del usuario son libres de procesarlas, por ejemplo, como en la figura de la izquierda.



A la izquierda, un posible procesamiento de una equivocada tabla en HTML 4.0; a la derecha, el único formato posible de una tabla no HTML similar.

### 17.5.1 Capas y transparencia en la tabla

Con el propósito de encontrar el fondo de cada celda de la tabla, los distintos elementos de la tabla pueden considerarse como si estuvieran en seis capas superpuestas. El fondo aplicado a un elemento de una de estas capas sólo será visible si las capas encima de ella tienen un fondo transparente.



[D]

Esquema de las capas de la tabla.

1. La capa inferior es un simple plano, representando la caja de la tabla en sí. Como todas las cajas, puede ser transparente.
2. La capa siguiente contiene los grupos de columnas. Los grupos de columnas son tan altos como la tabla, pero no necesariamente cubren horizontalmente toda la tabla.
3. Sobre los grupos de columnas están las áreas representando las cajas de las columnas. Como los grupos de columnas, las columnas son tan altas como la tabla, pero no necesariamente cubren horizontalmente toda la tabla.
4. La siguiente es la capa que contiene los grupos de filas. Cada grupo de filas es tan ancho como la tabla. Juntos, los grupos de filas cubren completamente la tabla desde arriba hasta abajo.
5. La penúltima capa contiene las filas. Las filas también cubren toda la tabla.
6. La capa superior contiene las propias celdas. Como muestra la figura, aunque todas las filas contienen el mismo número de celdas, puede que no todas las celdas tengan especificado un contenido. Estas celdas "vacías" son transparentes, dejando ver a través de ellas las capas inferiores.

En el siguiente ejemplo, la primer fila contiene cuatro celdas, pero la segunda fila no contiene ninguna celda, y así el fondo de la tabla se distingue a través, excepto donde una celda de la primera fila alcanza hasta esta fila. El siguiente código HTML y reglas de estilo

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <STYLE type="text/css">
      TABLE { background: #ff0; border-collapse: collapse }
      TD     { background: red; border: double black }
    </STYLE>
```

```
</HEAD>
<BODY>
  <P>
    <TABLE>
      <TR>
        <TD> 1
        <TD rowspan="2"> 2
        <TD> 3
        <TD> 4
      </TR>
      <TR><TD></TD></TR>
    </TABLE>
  </BODY>
</HTML>
```

podría tener el siguiente formato:

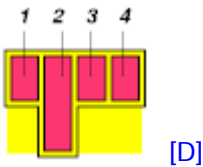


Tabla con tres celdas vacías en la fila inferior.

### 17.5.2 Algoritmos para el ancho de la tabla: la propiedad 'table-layout'

CSS no define una composición "óptima" para las tablas porque, en muchos casos, lo óptimo es una cuestión de gustos. CSS define restricciones que las aplicaciones del usuario deben respetar cuando componen una tabla. Las aplicaciones del usuario pueden usar el algoritmo que deseen para hacerlo y tienen libertad para privilegiar la velocidad de procesamiento por sobre la precisión, excepto cuando se selecciona el "algoritmo de estructura fija"

**'table-layout'**

- Valor: auto | fixed | inherit
- Inicial: auto
- Se aplica a: los elementos 'table' e 'inline-table'
- Se hereda: no
- Porcentajes: N/A
- Medios: visuales

La propiedad 'table-layout' controla el algoritmo usado para armar las celdas, filas y columnas de la tabla. Los valores tienen los siguientes significados:

- fixed**  
Se usa el algoritmo de estructura fija
- auto**  
Se usa cualquier algoritmo automático de composición de tablas

Los dos algoritmos se describen abajo.

**Composición fija de la tabla**

Con este (rápido) algoritmo, la composición horizontal de la tabla no depende del contenido de las celdas; sólo depende del ancho de la tabla, el ancho de las columnas y los bordes o espaciado de las celdas.

El ancho de la tabla puede especificarse explícitamente con la propiedad 'width'. Un valor 'auto' (tanto para 'display: table' como para 'display: inline-table') significa que se usa el algoritmo de [estructura automática de la tabla](#).

En el algoritmo de estructura fija de la tabla, el ancho de cada columna se determina de la siguiente manera:

1. Un elemento column con un valor distinto a 'auto' para la propiedad 'width' determina el ancho de esa columna.
2. Si no, una celda de la primer fila con un valor distinto a 'auto' para la propiedad 'width' determina el ancho de esa columna. Si la celda abarca más de una columna, el ancho se divide entre las columnas.
3. Las columnas restantes se dividen el espacio horizontal restante de la tabla (menos los bordes o el espaciado de las celdas).



El ancho de la tabla es entonces el mayor valor de la propiedad 'width' para el elemento tabla y la suma de los anchos de las columnas (más el espaciado de las celdas o los bordes). Si la tabla es más ancha que las columnas, el espacio extra debe ser distribuido entre las columnas.

De esta manera, la aplicación del usuario puede comenzar a construir la tabla una vez que ha recibido la primer fila completa. Las celdas de las filas subsecuentes no afectan el ancho de las columnas. Cualquier celda con un contenido que la desborda utiliza la propiedad 'overflow' para determinar si recorta el contenido que desborda.

**Composición automática de la tabla**

En este algoritmo (que generalmente no requiere más de dos pasos), el ancho de la tabla está dado por el ancho de sus columnas (y los [bordes](#) que intervienen). Este algoritmo refleja el comportamiento de algunas populares aplicaciones del usuario HTML en el momento de redactar esta especificación. Las AU no necesitan implementar este algoritmo para determinar la estructura de composición de la tabla en el caso en que 'table-layout' sea 'auto'; pueden usar cualquier otro algoritmo.

Este algoritmo puede resultar ineficiente dado que requiere a las aplicaciones del usuario tener acceso a todo el contenido de la tabla antes de determinar la composición final y puede demandar más de una pasada.

El ancho de las columnas se determina del siguiente modo:

1. Se calcula el ancho mínimo del contenido (AMC) de cada celda: el contenido estructurado puede abarcar cualquier cantidad de líneas pero no puede desbordar la caja de la celda. Si el ancho (A) especificado para la celda es mayor que AMC, A es el ancho mínimo de la celda. Un valor de 'auto' significa que AMC es el mínimo ancho de la celda.

Además, se calcula el ancho "máximo" de celda de cada una de las celdas: se da formato entonces al contenido sin cortes de línea salvo donde aparecen cortes de línea explícitos.

2. Para cada columna, se determina un ancho máximo y mínimo de columna con las celdas que ocupan sólo esta columna. El mínimo es el que requiere la celda con el mayor ancho mínimo de celda (o el ancho especificado para la columna, cualquiera sea mayor). El máximo es el que requiere la celda con el mayor ancho máximo de celda (o el ancho especificado para la columna, cualquiera sea mayor).
3. Por cada celda que abarca más de una columna, se incrementan los anchos mínimos de las columnas que abarca de modo que, juntas, ellas sean al menos tan anchas como la celda. Se hace lo mismo con los anchos máximos. Si es posible, se ensanchan las columnas abarcadas en aproximadamente el mismo valor.

Esto da un ancho máximo y mínimo para cada columna. Los anchos de las columnas influyen en el ancho final de tabla del siguiente modo:

1. Si la propiedad 'width' del elemento 'table' o 'inline-table' tiene especificado un valor (A) distinto a 'auto', el valor computado de la propiedad es el que resulte mayor entre A y el ancho mínimo requerido por todas las columnas más el espaciado de las celdas y los bordes (MIN). Si A es mayor que MIN, el ancho extra deberá ser distribuido entre las columnas.
2. Si el elemento 'table' o 'inline-table' tiene definido 'width: auto', el ancho computado de la tabla es el que resulte mayor entre el ancho del bloque de contención de la tabla y MIN. Sin embargo, si el ancho máximo requerido por las columnas más el espaciado de las celdas y los bordes (MAX) es menor que el del bloque de contención, se usa MAX.

Un valor de porcentaje para el ancho de una columna es relativo al ancho de la tabla. Si la tabla tiene definido 'width: auto', un porcentaje representa una condición en el ancho de la columna, la que una AU deberá tratar de satisfacer. (Obviamente, esto no siempre es posible: si el ancho de la columna es '110%', la condición no puede satisfacerse.)

***Nota. En este algoritmo, tanto las filas (y grupos de filas) y las columnas (y grupos de columnas) restringen y son restringidas por las dimensiones de las celdas que contienen. Colocando el ancho a una columna se puede influir indirectamente en la altura de una fila, y viceversa.***

**17.5.3 Algoritmos para la altura de la tabla**

La altura de una tabla está dada por la propiedad 'height' del elemento 'table' o 'inline-table'. Un valor de 'auto' significa que la altura es la suma de las alturas de las filas más cualquier espaciado de celdas y bordes. Cualquier otro valor especifica la altura explícitamente; la tabla puede resultar de este modo más larga o más corta que la altura de sus filas. CSS2 no especifica un procesamiento cuando la altura especificada de una tabla difiere de la altura del contenido, en particular si la altura del contenido debe sustituir la altura especificada; si no, cómo debe distribirse el espacio extra entre las filas que sumadas resultan menores que la altura especificada para la tabla; o, en caso de que la altura del contenido exceda la altura especificada de la tabla, si la AU debe proporcionar un mecanismo de desplazamiento. **Nota.** Las futuras versiones de CSS pueden especificar esto mejor.

La altura de la caja del elemento 'table-row' se calcula una vez que la aplicación del usuario tiene disponible todas las celdas de la fila: es la máxima altura ('height') especificada para la fila y la mínima altura (MIN) requerida por las celdas. Un valor 'auto' para la propiedad 'height' de 'table-row' significa que la altura computada de la fila es MIN. MIN depende de las alturas de las cajas de las celdas y la alineación de las cajas de las celdas (muy parecido al cálculo de la altura de una [caja de línea](#)). CSS2 no define a qué se refieren los valores de porcentaje en 'height' cuando se especifican para las filas o grupos de filas de la tabla.



En CSS2, la altura de una caja de celda es la máxima de la propiedad 'height' de las celdas de la tabla y la mínima altura requerida por el contenido (MIN). Un valor 'auto' para 'height' implica un valor computado de MIN. CSS2 no define a qué se refieren los valores de porcentaje en 'height' cuando se especifican para las celdas de la tabla.

CSS2 no especifica cómo las celdas que abarcan más de una fila afectan el cálculo de la altura de la fila excepto que la suma de las alturas de las filas involucradas debe ser suficiente para contener a la celda que abarca las filas.

La propiedad 'vertical-align' de cada celda de la tabla determina su alineación dentro de la fila. Cada contenido de una celda tiene una línea de base, una parte superior, una parte media y una parte inferior, como la propia fila. En el contexto de las tablas, los valores para 'vertical-align' tienen los siguientes significados:

**baseline**

La línea de base de la celda se pone a la misma altura que la línea de base de la primera de las filas que abarca (ver abajo la definición de líneas de base de celdas y filas).

**top**

La parte superior de la caja de la celda es alineada con la parte superior de la primer fila que abarca.

**bottom**

La parte inferior de la caja de la celda es alineada con la parte inferior de la última fila que abarca.

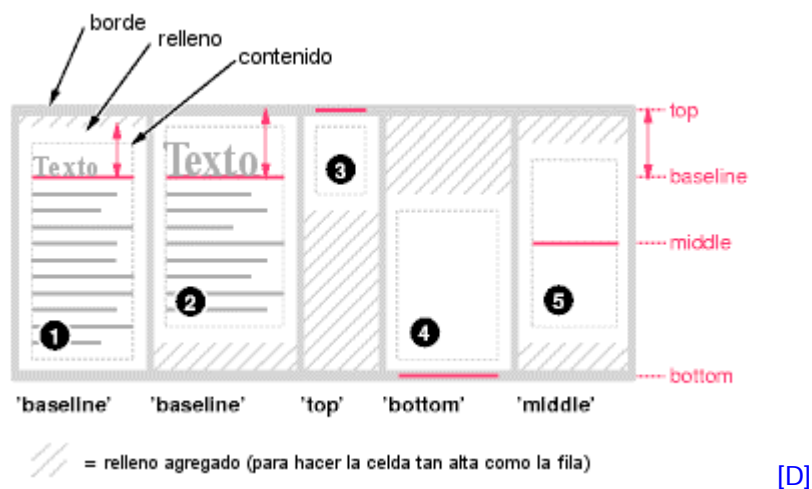
**middle**

El centro de la celda es alinado con el centro de las filas que abarca.

**sub, super, text-top, text-bottom**

Estos valores no se aplican a las celdas; en cambio la celda es alineada por la línea de base.

La línea de base de una celda es la línea de base de la primer **caja de línea** en la celda. Si no hay texto, la línea de base es la de cualquier objeto que es mostrado en la celda, o, si no hay ninguno, la parte inferior de la caja de la celda. La distancia máxima entre la parte superior de la caja de la celda y la línea de base en todas las celdas que tengan 'vertical-align: baseline' es usada por poner la línea de base de la fila. Aquí hay un ejemplo:



*Diagrama mostrando el efecto de los distintos valores de 'vertical-align' sobre las celdas de una tabla.*

Las cajas de las celdas 1 y 2 están alineadas por sus líneas de base. La caja de la celda 2 tiene mayor altura por sobre la línea de base, de modo que determina la línea de base de la fila. Observe que si no hay ninguna caja de celda alineada por su línea de base, la fila no tendrá (ni necesitará) una línea de base.

Para evitar situaciones ambiguas, la alineación de las celdas sigue el siguiente orden:

1. Primero son posicionadas las celdas que están alineadas por su línea de base. Esto establecerá la línea de base de la fila. Después se posicionan la celdas con 'vertical-align: top'.
2. La fila ahora tiene una parte superior, posiblemente una línea de base, y una altura provisoria, que es la distancia desde la parte superior hasta la base más baja de las celdas posicionadas hasta el momento. (Ver las condiciones sobre el relleno de la celda más abajo.)
3. Si alguna de las celdas restantes, aquellas alineadas por la base o la mitad, tiene una altura mayor que la altura actual de la fila, la altura de la fila será incrementada al máximo de aquellas celdas, bajando la base.
4. Finalmente se posicionan las celdas restantes.

Las cajas de celdas que son más pequeñas que la altura de la fila reciben relleno superior e inferior extra.

**17.5.4 Alineación horizontal en una columna**

La alineación horizontal del contenido de una celda dentro de una caja de celda se especifica con la propiedad 'text-align'.

Cuando la propiedad 'text-align' de más de una celda en una columna es puesta con el valor de una <cadena>, el contenido de esas celdas es alineado a través de un eje vertical. El comienzo de la cadena toca este eje. La direccionalidad de los caracteres determina si la cadena queda a la izquierda o a la derecha del eje.

El texto alineado de este modo solamente es útil si el texto cabe en una línea. El resultado es indefinido si el contenido de la celda ocupa más de una línea.

Si el valor de 'text-align' en una celda de una tabla es una cadena pero la cadena no aparece en el contenido de la celda, el final del contenido de la celda toca el eje vertical de alineación.

Observe que las cadenas no tienen que ser las mismas para cada celda, aunque usualmente lo son.

CSS no proporciona una forma de especificar el desplazamiento del eje de alineación vertical con respecto al borde de la caja de una columna.

La siguiente hoja de estilo:

```
TD { text-align: "." }
TD:before { content: "$" }
```

provocará que la columna de las cifras en pesos de la siguiente tabla HTML:

```
<TABLE>
<COL width="40">
<TR> <TH>Llamadas de larga distancia
<TR> <TD> 1.30
<TR> <TD> 2.50
<TR> <TD> 10.80
<TR> <TD> 111.01
<TR> <TD> 85.
<TR> <TD> 90
<TR> <TD> .05
<TR> <TD> .06
</TABLE>
```

se alinee por el punto decimal. En chiste, hemos usado el pseudo-elemento :before para colocar un signo pesos antes de cada cifra. La tabla podría ser procesada del siguiente modo:

```
Llamadas de larga distancia
      $1.30
      $2.50
      $10.80
      $111.01
      $85.
      $90
      $.05
      $.06
```

### 17.5.5 Efectos dinámicos en filas y columnas

La propiedad 'visibility' toma el valor 'collapse' para filas, grupos de filas, columnas y grupos de columnas. Este valor provoca que la fila o columna completa sea quitada de la pantalla, y el espacio normalmente ocupado por la fila o columna quede disponible para otro contenido. La supresión de la fila o columna, sin embargo, no afecta de ningún otro modo la composición de la tabla. Esto permite efectos dinámicos al quitar la filas o columnas sin forzar una recomposición de la tabla para calcular el cambio potencial en los requerimientos de la columna.

## 17.6 Bordes

Existen dos modelos distintos para definir los bordes de las celdas de una tabla en CSS. Uno es más conveniente para los llamados bordes separados alrededor de las celdas individuales, el otro es mejor para los bordes continuos de una punta a otra de la tabla. Muchos estilos de bordes pueden conseguirse con cualquier modelo, de modo que es a menudo una cuestión de gustos cuál de ellos se elige.

#### 'border-collapse'

- Valor: collapse | separate | inherit
- Inicial: collapse
- Se aplica a: los elementos 'table' e 'inline-table'
- Se hereda: si
- Porcentajes: N/A
- Medios: visuales

Esta propiedad selecciona un modelo de borde de la tabla. El valor 'separate' elige el modelo de bordes separados. El valor 'collapse' selecciona el modelo de bordes cerrados. Los modelos se describen abajo.

### 17.6.1 El modelo de bordes separados

**'border-spacing'**

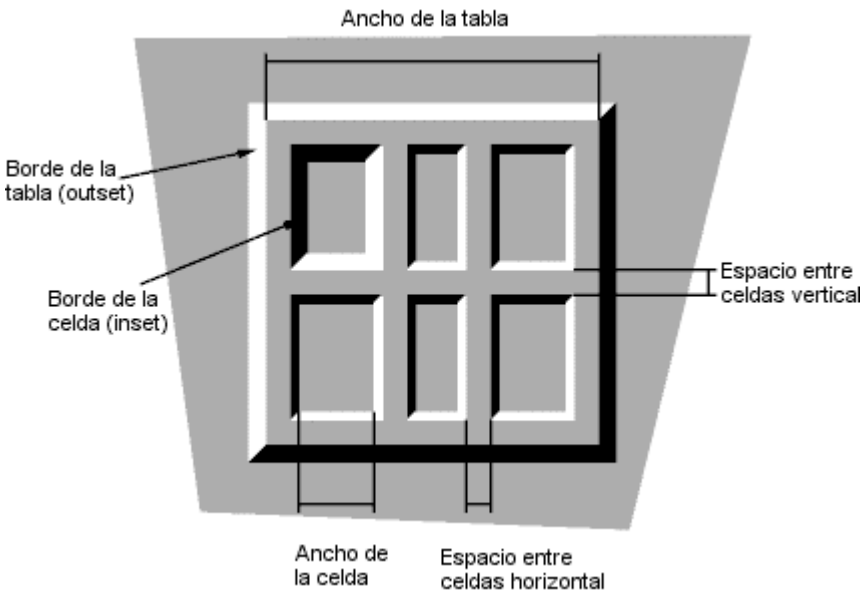
Valor: <medida> <medida>? | inherit  
Inicial: 0  
Se aplica a: los elementos 'table' e 'inline-table'  
Se hereda: si  
Porcentajes: N/A  
Medios: visuales

Las medidas especifican la distancia que separa los bordes de las celdas adyacentes. Si se especifica una medida, ésta da el espacio horizontal y vertical. Si se especifican dos, la primera da el espacio horizontal y la segunda el espacio vertical. Las medidas no pueden ser negativas.

En este modelo, cada celda tiene un borde individual. La propiedad 'border-spacing' especifica la distancia entre los bordes de celdas adyacentes. Este espacio es llenado con el fondo del elemento tabla. Las filas, columnas, grupos de filas y grupos de columnas no pueden tener bordes (es decir, las aplicaciones del usuario deben ignorar las propiedades de bordes para estos elementos).

La tabla de la imagen de abajo podría ser el resultado de una hoja de estilo como ésta:

```
TABLE      { border: outset 10pt;
              border-collapse: separate;
              border-spacing: 15pt }
TD         { border: inset 5pt }
TD.especial { border: inset 10pt } /* La celda superior izquierda */
```



[D]

*Una tabla con 'border-spacing' puesto con un valor de medida. Vea que cada celda tiene su propio borde y la tabla también tiene un borde separado.*

**Bordes alrededor de celdas vacías: la propiedad 'empty-cells'**

**'empty-cells'**

Valor: show | hide | inherit  
Inicial: show  
Se aplica a: los elementos 'table-cell'  
Se hereda: si  
Porcentajes: N/A  
Medios: visuales

En el modelo de bordes separados, esta propiedad controla el procesamiento de los bordes alrededor de las celdas que no tienen ningún contenido visible. Las celdas vacías y las celdas con la propiedad 'visibility' definida como 'hidden' son consideradas de no tener un contenido visible. El contenido visible incluye "&nbsp;" y otros espacios en blanco excepto ASCII CR ("0D"), LF ("0A"), tab ("09") y space ("20").

Cuando esta propiedad tiene el valor 'show', los bordes son dibujados alrededor de las celdas vacías (como las celdas normales).

Un valor de 'hide' significa que no se dibujará ningún borde alrededor de las celdas vacías. Más aún, si todas las celdas de una fila tienen un valor de 'hide' y no tienen contenido visible, toda la fila se comporta como si tuviera 'display: none'.

La siguiente regla provoca que los bordes sean dibujados alrededor de todas las celdas:

```
TABLE { empty-cells: show }
```

### 17.6.2 El modelo de bordes cerrados

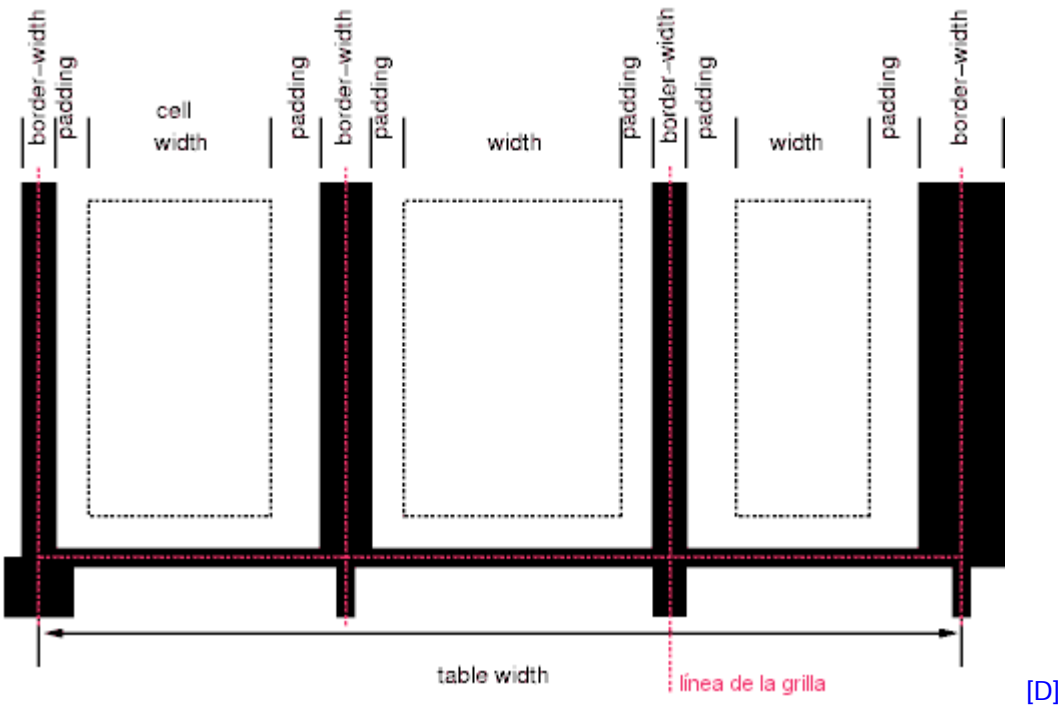
En el modelo de bordes cerrados, es posible especificar los bordes que rodean a toda o parte de una celda, fila, grupo de filas, columna y grupo de columnas. Los bordes para los atributos "rules" de HTML pueden especificarse de este modo.

Los bordes son centrados con las líneas de la grilla entre las celdas. Las aplicaciones del usuario deben encontrar una regla consistente para redondear en caso de un número impar de unidades continuas (pixels de pantalla, puntos de impresora).

El diagrama de más abajo muestra cómo interactúan el ancho de la tabla, el ancho de los bordes, el relleno y el ancho de la celda. Su relación está dada por la siguiente ecuación, que se mantiene para cada fila de la tabla:

$$\text{row-width} = (0.5 * \text{border-width}_0) + \text{padding-left}_1 + \text{width}_1 + \text{padding-right}_1 + \text{border-width}_1 + \text{padding-left}_2 + \dots + \text{padding-right}_n + (0.5 * \text{border-width}_n)$$

Aquí n es el número de celdas en la fila y border-width<sub>i</sub> se refiere al borde entre las celdas i é i + 1. Observe que sólo la mitad de los dos bordes exteriores se cuentan en el ancho de la tabla; las otras dos mitades de esos dos bordes quedan en el área de los márgenes.



*Esquema mostrando el ancho de las celdas y los bordes y el relleno de las celdas.*

Observe que en este modelo, el ancho de la tabla incluye la mitad del borde de la tabla. Además, en este modelo, una tabla no tiene relleno (pero tiene márgenes).

#### Resolución de conflictos de bordes

En el modelo de bordes cerrados, los bordes de cada lado de cada celda pueden ser especificados con las propiedades de bordes para cada elemento que se juntan en ese lado (celdas, filas, grupos de filas, columnas, grupos de columnas y la misma tabla), y esos bordes pueden variar en grosor, estilo y color. La regla de manejo es que en cada lado se elige el estilo más "atractivo" de borde, excepto que cualquier aparición del estilo 'hidden' elimina incondicionalmente el borde.

Las siguientes reglas determinan qué estilo de borde "gana" en caso de conflicto:

1. Los bordes con 'border-style' como 'hidden' tiene precedencia por sobre todos los demás bordes en conflicto. Cualquier borde con este valor suprime todos los bordes en esta ubicación.
2. Los bordes con un estilo 'none' tiene la más baja prioridad. Sólo si las propiedades de borde de todos los elementos que se encuentran en este lado son 'none' el borde será omitido (pero observe que 'none' es el valor predeterminado para el estilo de borde).
3. Si ninguno de los estilos es 'hidden' y al menos uno de ellos no es 'none', los bordes angostos son descartados en favor de los más gruesos. Si varios tienen el mismo 'border-width' los estilos son preferidos en este orden: 'double', 'solid', 'dashed', 'dotted', 'ridge', 'outset', 'groove', y el más bajo: 'inset'.
4. Si los estilos de borde difieren sólo en el color, entonces un estilo adjudicado a una celda gana a uno adjudicado a una fila, que gana a un grupo de filas, columna, grupo de columnas y, finalmente, tabla.

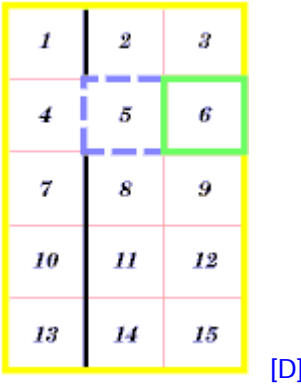
El ejemplo siguiente ilustra la aplicación de estas reglas de precedencia. Esta hoja de estilo:

```
TABLE          { border-collapse: collapse;
                  border: 5px solid yellow; }
*#col1         { border: 3px solid black; }
TD             { border: 1px solid red; padding: 1em; }
TD.solid-blue  { border: 5px dashed blue; }
TD.solid-green { border: 5px solid green; }
```

con esta fuente HTML:

```
<P>
<TABLE>
<COL id="col1"><COL id="col2"><COL id="col3">
<TR id="row1">
  <TD> 1
  <TD> 2
  <TD> 3
</TR>
<TR id="row2">
  <TD> 4
  <TD class="solid-blue"> 5
  <TD class="solid-green"> 6
</TR>
<TR id="row3">
  <TD> 7
  <TD> 8
  <TD> 9
</TR>
<TR id="row4">
  <TD> 10
  <TD> 11
  <TD> 12
</TR>
<TR id="row5">
  <TD> 13
  <TD> 14
  <TD> 15
</TR>
</TABLE>
```

producirá algo como esto:



[D]

Un ejemplo de una tabla con bordes cerrados.

Ejemplo(s):

El ejemplo siguiente muestra una tabla con líneas horizontales entre las filas. El borde superior de la tabla está definido como 'hidden' para suprimir el borde superior de la primer fila. Esto implementa el atributo "rules" de HTML 4.0 (rules="rows").

```
TABLE[rules=rows] TR { border-top: solid }
TABLE[rules=rows]    { border-collapse: collapse;
                      border-top: hidden }
```

| a | b  | c  |
|---|----|----|
| 3 | 4  | 5  |
| 5 | 12 | 13 |

[D]

Tabla con líneas horizontales entre las filas.

En este caso el mismo efecto puede lograrse sin colocar un borde 'hidden' en TABLE, suprimiendo la primer fila separadamente. El método que se prefiera es una cuestión de gustos.

```
TR:first-child { border-top: none }
TR { border-top: solid }
```

Aquí hay otro ejemplo de bordes cerrados ocultos:

|     |     |
|-----|-----|
| foo | bar |
| foo | bar |

[D]

Tabla con dos bordes internos omitidos.

Fuente HTML:

```
<TABLE style="border-collapse: collapse; border: solid;">
<TR><TD style="border-right: hidden; border-bottom: hidden">foo</TD>
  <TD style="border: solid">bar</TD></TR>
<TR><TD style="border: none">foo</TD>
  <TD style="border: solid">bar</TD></TR>
</TABLE>
```

### 17.6.3 Estilos de bordes

Algunos valores de 'border-style' tienen un significado distinto para las tablas que en otros elementos. En la lista de abajo están marcados con un asterisco.

**none**

Ningún borde.

**\*hidden**

Lo mismo que 'none', pero en el [modelo de bordes cerrados](#), también inhibe cualquier otro borde (ver la sección sobre [conflictos de bordes](#)).

**dotted**

El borde es una serie de puntos.

**dashed**

El borde es una serie de segmentos cortos de línea.

**solid**

El borde es un único segmento de línea.

**double**

El borde son dos líneas sólidas. La suma de las dos líneas y el espacio entre ellas equivalen al valor de 'border-width'.

**groove**

El borde parece como tallado en el lienzo.

**ridge**

Lo opuesto a 'groove': el borde parece estar sobresaliendo del lienzo.

- \*inset**  
En el [modelo de bordes separados](#), el borde provoca que toda la caja luzca como si estuviera en bajorrelieve en el lienzo. En el [modelo de bordes cerrados](#), es igual a 'groove'.
- \*outset**  
En el [modelo de bordes separados](#), el borde provoca que toda la caja luzca como si estuviera en relieve sobre el lienzo. En el [modelo de bordes cerrados](#), es igual a 'ridge'.

## 17.7 Procesamiento sonoro de las tablas

Cuando una tabla es expresada por un generador de voz, la relación entre las celdas de datos y las celdas de título deben ser expresada de distinto modo que por la alineación horizontal y vertical. Algunos browser sonoros pueden permitir al usuario moverse alrededor de un espacio bidimensional, de este modo le dan la oportunidad de localizar las relaciones representadas espacialmente. Cuando eso no es posible, la hoja de estilo debe especificar en qué punto son expresados los títulos.

### 17.7.1 Lectura de títulos: la propiedad 'speak-header'

- 'speak-header'**
- Valor: once | always | inherit
  - Inicial: once
  - Se aplica a: los elementos que tienen la información de títulos en una tabla
  - Se hereda: si
  - Porcentajes: N/A
  - Medios: auditivos

Esta propiedad especifica si los títulos de la tabla son expresados antes de cada celda, o sólo antes de una celda cuando la celda está asociada con un título diferente a la celda anterior. Los valores tienen los siguientes significados:

- once**  
El título se dice una vez, antes de una serie de celdas.
- always**  
El título se dice antes de cada celda pertinente.

Cada lenguaje de documento puede tener distintos mecanismos que permitan a los autores especificar los títulos. Por ejemplo, en HTML 4.0 ([HTML40](#)), es posible especificar la información de los títulos con tres atributos distintos ("headers", "scope" y "axis"), y la especificación brinda un algoritmo para determinar la información de los títulos cuando estos atributos no han sido especificados.

| Informe de gastos de viaje |        |        |            |            |
|----------------------------|--------|--------|------------|------------|
|                            | Comida | Hotel  | Transporte | subtotales |
| <b>Buenos Aires</b>        |        |        |            |            |
| 25-Ago-97                  | 37.74  | 112.00 | 45.00      |            |
| 26-Ago-97                  | 27.28  | 112.00 | 45.00      |            |
| subtotales                 | 65.02  | 224.00 | 90.00      | 379.02     |
| <b>Córdoba</b>             |        |        |            |            |
| 27-Ago-97                  | 96.25  | 109.00 | 36.00      |            |
| 28-Ago-97                  | 35.00  | 109.00 | 36.00      |            |
| subtotales                 | 131.25 | 218.00 | 72.00      | 421.25     |
| <b>Totales</b>             | 196.27 | 442.00 | 162.00     | 800.27     |

[D]

*Imagen de una tabla con celdas de títulos ("Buenos Aires" y "Córdoba") que no se encuentran en la misma columna o fila que los datos a los que se aplican.*

Este ejemplo HTML presenta el dinero gastado en comida, hotel y transporte en dos localidades (Buenos Aires y Córdoba) en días sucesivos. Conceptualmente, puede considerar la tabla en términos de un espacio dimensional n. Los títulos de este espacio son: localidad, día, categoría y subtotal. Algunas celdas definen las marcas en un eje mientras otras dan el gasto de dinero en cierto punto de este espacio. El código de esta tabla es:

```
<TABLE>
<CAPTION>Informe de gastos de viaje</CAPTION>
<TR>
  <TH></TH>
  <TH>Comida</TH>
  <TH>Hotel</TH>
  <TH>Transporte</TH>
  <TH>subtotales</TH>
```



```
</TR>
<TR>
  <TH id="buenos-aires" axis="buenos-aires">Buenos Aires</TH>
</TR>
<TR>
  <TH headers="buenos-aires">25-Ago-97</TH>
  <TD>37.74</TD>
  <TD>112.00</TD>
  <TD>45.00</TD>
  <TD></TD>
</TR>
<TR>
  <TH headers="buenos-aires">26-Ago-97</TH>
  <TD>27.28</TD>
  <TD>112.00</TD>
  <TD>45.00</TD>
  <TD></TD>
</TR>
<TR>
  <TH headers="buenos-aires">subtotales</TH>
  <TD>65.02</TD>
  <TD>224.00</TD>
  <TD>90.00</TD>
  <TD>379.02</TD>
</TR>
<TR>
  <TH id="cordoba" axis="cordoba">Córdoba</TH>
</TR>
<TR>
  <TH headers="cordoba">27-Ago-97</TH>
  <TD>96.25</TD>
  <TD>109.00</TD>
  <TD>36.00</TD>
  <TD></TD>
</TR>
<TR>
  <TH headers="cordoba">28-Ago-97</TH>
  <TD>35.00</TD>
  <TD>109.00</TD>
  <TD>36.00</TD>
  <TD></TD>
</TR>
<TR>
  <TH headers="cordoba">subtotales</TH>
  <TD>131.25</TD>
  <TD>218.00</TD>
  <TD>72.00</TD>
  <TD>421.25</TD>
</TR>
<TR>
  <TH>Totales</TH>
  <TD>196.27</TD>
  <TD>442.00</TD>
  <TD>162.00</TD>
  <TD>800.27</TD>
</TR>
</TABLE>
```

Proporcionando el modelo de datos de este modo, los autores hacen posible que los browsers con capacidades de voz exploren la tabla productivamente, por ej., cada celda puede ser leída como una lista, repitiendo los títulos aplicables antes de cada celda de datos:

```
Buenos Aires, 25-Ago-97, Comida: 37.74
Buenos Aires, 25-Ago-97, Hotel: 112.00
Buenos Aires, 25-Ago-97, Transporte: 45.00
...
```

El browser puede leer los títulos sólo cuando cambian:

```
Buenos Aires, 25-Ago-97, Comida: 37.74
  Hotel: 112.00
  Transporte: 45.00
26-Ago-97, Comida: 27.28
  Hotel: 112.00
...
```

# 18 Interfaz del usuario

## 18.1 Cursores: la propiedad 'cursor'

|                 |  |
|-----------------|--|
| <b>'cursor'</b> |  |
| Valor:          | [ [<uri> ,]* [ auto   crosshair   default   pointer   move   e-resize   ne-resize   nw-resize   n-resize   se-resize   sw-resize   s-resize   w-resize  text   wait   help ] ]   inherit |
| Inicial:        | auto   |
| Se aplica a:    | todos los elementos  |
| Se hereda:      | si   |
| Porcentajes:    | N/A  |
| Medios:         | visuales, interactivos   |

Esta propiedad especifica el tipo de cursor que será utilizado para el dispositivo señalador. Los valores tienen los siguientes significados:

- auto**  
La AU determina el cursor a mostrar basado en el contexto actual.
- crosshair**  
Una cruz simple (ej., segmentos cortos de línea semejantes a un signo "+").
- default**  
El cursor predeterminado dependiente de la plataforma. A menudo procesado como una flecha.
- pointer**  
El cursor es un puntero que indica un enlace.
- move**  
Indica que algo será movido.
- e-resize, ne-resize, nw-resize, n-resize, se-resize, sw-resize, s-resize, w-resize**  
Indica que algún borde será movido. Por ejemplo, el cursor 'se-resize' se usa cuando el movimiento comienza desde la esquina sudeste de la caja.
- text**  
Indica texto que puede seleccionarse. A menudo procesado como una barra-l.
- wait**  
Indica que el programa está ocupado y el usuario debe esperar. A menudo procesado como un reloj de pulsera o un reloj de arena.
- help**  
Hay ayuda disponible sobre el objeto señalado por el cursor. A menudo procesado como un signo de interrogación o un globo.
- <uri>**  
La aplicación del usuario recupera el cursor de un recurso señalado por el URI. Si la aplicación del usuario no puede manejar el primero de una lista de cursores, debe intentar manejar el segundo, etc. Si la aplicación del usuario no puede manejar ningún cursor definido por el usuario, debe usar el cursor genérico al final de la lista.

```
P { cursor : url("micursor.cur"), url("segundo.csr"), text; }
```

## 18.2 Preferencias de color del usuario

Además de poder asignar [valores de color](#) predeterminados a los textos, fondos, etc., CSS2 permite a los autores especificar colores de manera que se integren con el entorno gráfico del usuario. Las reglas de estilo que tienen en cuenta de este modo las preferencias del usuario ofrecen las siguientes ventajas:

- 1. Producen páginas que coinciden con un determinado aspecto y gusto del usuario.
- 2. Producen páginas que pueden ser más accesibles en tanto las preferencias del usuario pueden estar relacionadas con una discapacidad.

El conjunto de valores definidos para los colores del sistema intenta ser exhaustivo. Para los sistemas que no tienen los valores correspondientes, el valor especificado debe ser obtenido del atributo más cercano del sistema, o del color predeterminado.

El siguiente es un listado de los valores adicionales para los atributos de CSS relacionados con el color y sus significados generales. Cualquier propiedad de color (por ej., 'color' o 'background-color') puede tomar uno de los siguientes nombres. Aunque estos no distinguen entre mayúsculas y minúsculas, se recomienda que la mezcla de mayúsculas y minúsculas mostrada abajo sea utilizada para hacer más legibles los nombres.

- ActiveBorder**  
Borde de la ventana activa.
- ActiveCaption**  
Título de la ventana activa.
- AppWorkspace**  
Color de fondo de una interfaz de documentos múltiples.
- Background**  
Fondo del escritorio.
- ButtonFace**  
Color de la cara de los elementos tridimensionales de la pantalla.

|                            |   |
|----------------------------|---|
| <b>ButtonHighlight</b>     | Sombra oscura de los elementos tridimensionales de la pantalla (para los bordes más alejados de la fuente luminosa).            |
| <b>ButtonShadow</b>        | Color de la sombra de los elementos tridimensionales de la pantalla.  |
| <b>ButtonText</b>          | Texto de los botones.   |
| <b>CaptionText</b>         | Texto de los títulos, cajas de dimensiones y cajas de flechas de las barras de desplazamiento.                                  |
| <b>GrayText</b>            | Texto grisado (deshabilitado). Este color es fijado en #000 si el manejador de pantalla actual no soporta un color gris sólido. |
| <b>Highlight</b>           | Item(s) seleccionados en un control.  |
| <b>HighlightText</b>       | Texto del(los) item(s) seleccionados en un control.   |
| <b>InactiveBorder</b>      | Borde de la ventana inactiva.   |
| <b>InactiveCaption</b>     | Título de la ventana inactiva.  |
| <b>InactiveCaptionText</b> | Color del texto de un título inactivo.  |
| <b>InfoBackground</b>      | Color de fondo de los controles de sugerencias (tooltip).   |
| <b>InfoText</b>            | Color del texto de los controles de sugerencias (tooltip).  |
| <b>Menu</b>                | Fondo del menu.   |
| <b>MenuText</b>            | Texto de los menús.   |
| <b>Scrollbar</b>           | Area gris de la barra de desplazamiento.  |
| <b>ThreeDDarkShadow</b>    | Sombra oscura de los elementos tridimensionales de pantalla.  |
| <b>ThreeDFace</b>          | Color de la cara de los elementos tridimensionales de pantalla.   |
| <b>ThreeDHighlight</b>     | Color resaltado de los elementos tridimensionales de pantalla.  |
| <b>ThreeDLightShadow</b>   | Color claro de los elementos tridimensionales de pantalla (para los bordes enfrentados a la fuente luminosa).                   |
| <b>ThreeDShadow</b>        | Sombra oscura de los elementos tridimensionales de pantalla.  |
| <b>Window</b>              | Fondo de la ventana.  |
| <b>WindowFrame</b>         | Marco de la ventana.  |
| <b>WindowText</b>          | Texto de las ventanas.  |

Por ejemplo, para poner como colores de primer plano y de fondo de un párrafo los mismos colores de primer plano y de fondo que la ventana del usuario, escriba lo siguiente:

```
P { color: WindowText; background-color: Window }
```

## 18.3 Preferencias de fuentes del usuario

De igual modo que para los colores, los autores pueden especificar las fuentes de manera de utilizar los recursos del sistema del usuario. Por favor, consulte la propiedad 'font' para más detalles.

## 18.4 Contornos dinámicos: la propiedad 'outline'

A veces, los autores de hojas de estilo pueden desear crear contornos alrededor de los objetos visuales como botones, campos activos de los formularios, mapas de imágenes, etc., para enfatizarlos. Los contornos de CSS difieren de los **borders** en los siguientes sentidos:

1. Los contornos no ocupan espacio.
2. Los contornos pueden ser no rectangulares.

Las propiedades de los contorno controlan el estilo de estos contornos dinámicos.

|                     |   |
|---------------------|---|
| <b>'outline'</b>    |   |
| <i>Valor:</i>       | [ <'outline-color'>    <'outline-style'>    <'outline-width'> ]   inherit |
| <i>Inicial:</i>     | ver las propiedades individuales  |
| <i>Se aplica a:</i> | todos los elementos   |
| <i>Se hereda:</i>   | no  |

Porcentajes: N/A  
Medios: visuales, interactivos

'outline-width'

Valor: <border-width> | inherit  
Inicial: medium  
Se aplica a: todos los elementos  
Se hereda: no  
Porcentajes: N/A  
Medios: visuales, interactivos

'outline-style'

Valor: <border-style> | inherit  
Inicial: none  
Se aplica a: todos los elementos  
Se hereda: no  
Porcentajes: N/A  
Medios: visuales, interactivos

'outline-color'

Valor: <color> | invert | inherit  
Inicial: invert  
Se aplica a: todos los elementos  
Se hereda: no  
Porcentajes: N/A  
Medios: visuales, interactivos

El contorno creado con las propiedades de contorno es dibujado "sobre" una caja, es decir, el contorno está siempre encima, y no influye en la posición o tamaño de la caja, o de ninguna otra caja. Por lo tanto, mostrar u ocultar los contornos no provoca el redibujado.

El contorno es dibujado comenzando junto fuera del [límite del borde](#).

Los contornos pueden ser no rectangulares. Por ejemplo, si el elemento es dividido entre varias líneas, el contorno es el contorno mínimo que encierra a todas las cajas del elemento. En contraste con los [bordes](#), el contorno no es abierto en la línea de finalización o comienzo de la caja, pero está siempre totalmente conectado.

La propiedad 'outline-width' acepta los mismos valores que 'border-width'.

La propiedad 'outline-style' acepta los mismos valores que 'border-style', excepto que 'hidden' no es un estilo de contorno lícito.

La propiedad 'outline-color' acepta todos los colores, como así también la palabra clave 'invert'. Se espera que 'invert' produzca una inversión de color en los píxeles de la pantalla. Este es un truco común para asegurarse que el borde del foco sea visible, sin que importe el color de fondo.

La propiedad 'outline' es una propiedad resumida, y determina las tres de 'outline-style', 'outline-width' y 'outline-color'.

**Observe que el contorno es el mismo en todos los lados. Contrariamente a los bordes, no hay propiedades 'outline-top' o 'outline-left'.**

Esta especificación no define cómo se dibujan los múltiples contornos solapados, o cómo los contornos son dibujados para las cajas que están parcialmente ocultas detrás de otros elementos.

**Nota. Como el contorno del foco no afecta al formato (es decir, no se deja ningún espacio para él en el modelo de caja), bien puede tapar otros elementos de la página.**

Aquí hay un ejemplo del dibujo de un contorno grueso alrededor del elemento `BUTTON`:

```
BUTTON { outline-width : thick }
```

Los scripts pueden usarse para cambiar dinámicamente el ancho del contorno, sin provocar el redibujado.

### 18.4.1 Contornos y foco

Las interfaces gráficas del usuario pueden usar contornos alrededor de los elementos para indicarle al usuario qué elemento de la página tiene el *foco*. Estos contornos son sumados a cualquier borde, y pasar entre contornos activados y desactivados no debe provocar el redibujado del documento. El foco es el sujeto de la interacción del usuario con un documento (por ej., introduciendo texto, seleccionando un botón, etc.). Las aplicaciones del usuario que dan soporte a los [grupos de medios interactivos](#) deben mantener identificada la ubicación del foco y

también debe representar el foco. Esto puede realizarse utilizando contornos dinámicos junto con la pseudo-clase :focus.

Por ejemplo, para dibujar una línea negra gruesa alrededor de un elemento cuando éste tiene el foco, y una línea roja gruesa cuando está activado, pueden usarse las siguientes reglas:

```
:focus { outline: thick solid black }
:active { outline: thick solid red }
```

## 18.5 Amplificación

El grupo de trabajo de CSS considera que la amplificación de un documento o de partes de un documento de debe especificarse a través de las hojas de estilo. Las aplicaciones del usuario pueden dar soporte a tales amplificaciones de distintos modos (ej., imágenes más grandes, sonidos más fuertes, etc.).

Cuando amplía una página, las AU deben mantener las relaciones entre los elementos posicionados. Por ejemplo, una tira cómica puede estar compuesta de imágenes con elementos de texto superpuestos. Cuando amplía esta página, una aplicación del usuario debe mantener el texto dentro de los globos de la tira cómica.

# 19 Hojas de estilo auditivas

## 19.1 Introducción a las hojas de estilo auditivas

El procesamiento sonoro de un documento, comúnmente usado ya por las comunidades de ciegos y discapacitados visuales, combina la síntesis de voz e "íconos sonoros". A menudo tal representación sonora se produce convirtiendo al documento en texto puro y procesándolo con un *lector de pantalla* --programa o dispositivo que simplemente lee todos los caracteres sobre la pantalla--. Esto resulta en una representación menos efectiva que la que se conseguiría si se mantuviera la estructura del documento. Las hojas de estilo para la presentación sonora pueden utilizarse conjuntamente con las propiedades visuales (mezcla de medios) o como una alternativa sonora a la presentación visual.

Además de las obvias ventajas de accesibilidad, hay otros grandes mercados para escuchar la información, incluyendo el uso en automóviles, sistemas de documentación industrial y médica (intranets), entretenimiento hogareño y ayuda a los usuarios que están aprendiendo a leer o que tienen dificultades de lectura.

Cuando se usan las propiedades auditivas, el lienzo consiste en un espacio físico tridimensional (sonido circundante) y un espacio temporal (uno puede especificar sonidos antes, durante y después de otros sonidos). Las propiedades CSS también permiten a los autores variar la calidad en la síntesis de la voz (tipo de voz, frecuencia, inflexión, etc.).

```
H1, H2, H3, H4, H5, H6 {
    voice-family: pablo;
    stress: 20;
    richness: 90;
    cue-before: url("ping.au")
}
P.heidi { azimuth: center-left }
P.pedro { azimuth: right }
P.suave { volume: x-soft }
```

Esto dirigirá al sintetizador de voz a leer los títulos con una voz (una especie de "fuente auditiva") llamada "pablo", en un tono llano, pero en voz muy alta. Antes de decir los títulos, se escuchará un archivo de sonido del URL dado. Los párrafos con clase "heidi" parecerán venir del frente y la izquierda (si el sistema de sonidos soporta el audio espacial), y los párrafos con clase "pedro" vendrán de la derecha. Los párrafos con clase "suave" serán muy suaves.

## 19.2 Propiedades del volumen: 'volume'

### 'volume'

- Valor: <número> | <porcentaje> | silent | x-soft | soft | medium | loud | x-loud | inherit
- Inicial: medium
- Se aplica a: todos los elementos
- Se hereda: si
- Porcentajes: referidos al valor heredado
- Medios: auditivos

Volume se refiere al volumen medio del sonido. En otras palabras, una voz altamente modulada a un volumen de 50 podría alcanzar bastante más que eso. Los valores globales pueden ser ajustado por las personas para mayor confort, por ejemplo con un control físico del volumen (que aumentaría los valores 0 y 100 proporcionalmente), lo que esta propiedad hace es ajustar el rango dinámico.

Los valores tienen los siguientes significados:

#### <número>

Cualquier número entre '0' y '100'. '0' representa el *mínimo nivel de volumen audible* y 100 corresponde al *máximo nivel comfortable*.

#### <porcentaje>

Los valores de porcentaje son calculados con relación al valor heredado, y son entonces son recortados a un rango de '0' a '100'.

#### silent

Ningún sonido en absoluto. El valor '0' no significa lo mismo que 'silent'.

#### x-soft

Igual a '0'.

#### soft

Igual a '25'.

#### medium

Igual a '50'.

#### loud

Igual a '75'.

#### x-loud

Igual a '100'.

Las aplicaciones del usuario deben permitir que los valores correspondientes a '0' y '100' sean determinados por el oyente. Ninguna configuración es universalmente aplicable; los valores apropiados dependen del equipamiento en uso (parlantes, auriculares), el medio ambiente (en el automóvil, un equipo hogareño, la biblioteca) y las preferencias personales. Algunos ejemplos:

- Un navegador para usar en el automóvil tiene una configuración apropiada para cuando hay mucho ruido ambiente. '0' equivaldría a un nivel bastante alto y '100' a un nivel realmente alto. El habla es fácilmente audible por sobre el ruido del camino pero el rango dinámico global es reducido. Los automóviles con mejor sistema de aislación podrían permitir un rango dinámico más amplio.
- Otro navegador audible está utilizándose en un departamento, tarde por la noche, o en un cuarto de estudio compartido. '0' es colocado a un nivel muy bajo y '100' a un nivel bastante bajo, también. Como en el primer ejemplo, hay un pequeño desnivel; el rango dinámico es reducido. Los volúmenes reales son bajos aquí, mientras que eran altos en el primer ejemplo.
- En una silenciosa y aislada casa, un caro equipo hogareño de alta fidelidad. '0' es puesto basta bajo y '100' es totalmente alto; hay un gran rango dinámico.

La misma hoja de estilo del autor puede ser usada en todos los casos, simplemente ajustando adecuadamente los puntos '0' y '100' del lado del cliente.

### 19.3 Propiedades del habla: 'speak'

**'speak'**

*Valor:* normal | none | spell-out | inherit  
*Inicial:* normal  
*Se aplica a:* todos los elementos  
*Se hereda:* si  
*Porcentajes:* N/A  
*Medios:* auditivos

Esta propiedades especifica si el texto se procesará sonoramente y en ese caso, de qué manera (algo análogo a la propiedad 'visibility' en algunos sentidos y a 'display' en otros). Los posibles valores son:

**none**

Suprime el procesamiento sonoro para que el elemento no requiera tiempo para procesarse. Observer, sin embargo, que los descendientes pueden anular este valor y ser expresados verbalmente. (Para asegurarse de suprimir el procesamiento de un elemento y sus descendientes, utilice la propiedad 'display').

**normal**

Utiliza las reglas de pronunciación adecuadas al idioma para el procesamiento de un elemento y sus descendientes.

**spell-out**

Deletrea el texto una letra por vez (útil para siglas y abreviaturas).

Note la diferencia entre un elemento cuya propiedad 'volume' tiene un valor de 'silent' y un elemento cuya propiedad 'speak' tiene el valor 'none'. El procesamiento toma el mismo tiempo que si hubiera sido expresado, incluyendo cualquier pausa antes y después del elemento, pero ningún sonido es generado. El último no requiere ningún tiempo y no es procesado (aunque sus descendientes pueden serlo).

### 19.4 Propiedades de la pausa: 'pause-before', 'pause-after' y 'pause'

**'pause-before'**

*Valor:* <tiempo> | <porcentaje> | inherit  
*Inicial:* depende de la aplicación del usuario  
*Se aplica a:* todos los elementos  
*Se hereda:* no  
*Porcentajes:* ver el texto  
*Medios:* auditivos

**'pause-after'**

*Valor:* <timtiempo> | <porcentaje> | inherit  
*Inicial:* depende de la aplicación del usuario  
*Se aplica a:* todos los elementos  
*Se hereda:* no  
*Porcentajes:* ver el texto  
*Medios:* auditivos

Estas propiedades especifican una pausa que debe observarse antes (o después) de expresar verbalmente el contenido de un elemento. Los valores tienen los siguientes significados:

**<tiempo>**

Expresa la pausa en unidades de tiempo absolutas (segundos y milisegundos).



<porcentaje>

Se refiere al inverso del valor de la propiedad 'speech-rate'. Por ejemplo, si 'speech-rate' es de 120 palabras por minuto (es decir, una palabra toma medio segundo, o 500ms) entonces una 'pause-before' de 100% significa una pausa de 500 ms y una 'pause-before' de 20% significa 100ms.

Las pausas se insertan entre el contenido del elemento y cualquier contenido 'cue-before' o 'cue-after'.

Los autores deben usar unidades relativas para crear hojas de estilo más sólidas ante los grandes cambios en la velocidad del discurso (speech-rate).

'pause'

Valor: [ [<tiempo> | <porcentaje>]{1,2} ] | inherit  
Inicial: depende de la aplicación del usuario  
Se aplica a: todos los elementos  
Se hereda: no  
Porcentajes: ver las descripciones de 'pause-before' y 'pause-after'  
Medios: auditivos

La propiedad 'pause' es una fórmula abreviada para determinar 'pause-before' y 'pause-after'. Si se dan dos valores, el primer valor es 'pause-before' y el segundo es 'pause-after'. Si sólo se da un valor, éste se aplica a ambas propiedades.

```
H1 { pause: 20ms } /* pause-before: 20ms; pause-after: 20ms */  
H2 { pause: 30ms 40ms } /* pause-before: 30ms; pause-after: 40ms */  
H3 { pause-after: 10ms } /* pause-before: ?; pause-after: 10ms */
```

19.5 Propiedades de la señal: 'cue-before', 'cue-after' y 'cue'

'cue-before'

Valor: <uri> | none | inherit  
Inicial: none  
Se aplica a: todos los elementos  
Se hereda: no  
Porcentajes: N/A  
Medios: auditivos

'cue-after'

Value: <uri> | none | inherit  
Inicial: none  
Se aplica a: todos los elementos  
Se hereda: no  
Porcentajes: N/A  
Medios: auditivos

Los íconos sonoros son otra manera de distinguir los elementos semánticos. Los sonidos pueden oírse antes y/o después del elemento para delimitarlo. Los valores tienen los siguientes significados:

<uri>

El URI debe señalar un recurso de ícono sonoro. Si el URI resulta en otra cosa que un archivo de audio, como una imagen, el recurso debe ignorarse y la propiedad tratada con si tuviera el valor 'none'.

none

Ningún ícono sonoro es especificado.

```
A {cue-before: url("bell.aiff"); cue-after: url("dong.wav") }  
H1 {cue-before: url("pop.au"); cue-after: url("pop.au") }
```

'cue'

Valor: [ <'cue-before'> || <'cue-after'> ] | inherit  
Inicial: no definido para las propiedades resumidas  
Se aplica a: todos los elementos  
Se hereda: no  
Porcentajes: N/A  
Medios: auditivos

La propiedad 'cue' es una fórmula resumida para fijar 'cue-before' and 'cue-after'. Si se especifican dos valores, el primer valor es 'cue-before' y el segundo es 'cue-after'. Si solamente se especifica un valor, éste se aplica a ambas propiedades.

Las siguientes dos reglas son equivalentes:

```
H1 {cue-before: url("pop.au"); cue-after: url("pop.au") }  
H1 {cue: url("pop.au") }
```

Si una aplicación del usuario no puede procesar un ícono sonoro (ej., el sistema del usuario no lo permite), recomendamos que produzca una señal alternativa (por ej., haciendo aparecer un aviso, emitiendo un sonido de advertencia, etc.).

Por favor, ver las secciones sobre [los pseudo-elementos :before y :after](#) por información acerca de otras técnicas de generación de contenidos.

## 19.6 Propiedades de la mezcla: 'play-during'

### 'play-during'

|              |  |
|--------------|--|
| Valor:       | <uri> mix? repeat?   auto   none   inherit |
| Inicial:     | auto                                       |
| Se aplica a: | todos los elementos                        |
| Se hereda:   | no   |
| Porcentajes: | N/A  |
| Medios:      | auditivos                                  |

Similar a las propiedades 'cue-before' y 'cue-after', estas propiedades especifican un sonido que será escuchado como fondo mientras el contenido de un elemento es enunciado. Los valores tienen los siguientes significados:

#### <uri>

El sonido designado por este <uri> se escucha como fondo mientras el contenido del elemento es enunciado.

#### mix

Cuando está presente, esta palabra clave significa que el sonido heredado de la propiedad 'play-during' del elemento padre continúa escuchándose y el sonido designado por el <uri> es mezclado con él. Si 'mix' no es especificado, el sonido de fondo del elemento reemplaza al del padre.

#### repeat

Cuando está presente, esta palabra clave significa que el sonido se repetirá si es demasiado corto para completar toda la duración del elemento. De otro modo, el sonido se escucha una vez y entonces se detiene. Esto es similar a la propiedad 'background-repeat'. Si el sonido es demasiado largo para el elemento, es cortado una vez que el elemento ha sido enunciado.

#### auto

El sonido del elemento padre continúa escuchándose (no es reiniciado, que habría sido el caso si la propiedad hubiera sido heredada).

#### none

Esta palabra clave significa que hay silencio. El sonido del elemento padre (si lo hubiera) es silenciado durante el elemento actual y continúa después del elemento actual.

```
BLOCKQUOTE.triste { play-during: url("violines.aiff") }
BLOCKQUOTE.Q      { play-during: url("arpa.wav") mix }
SPAN.silencio      { play-during: none }
```

## 19.7 Propiedades espaciales: 'azimuth' y 'elevation'

El audio espacial es una importante propiedad de estilo para la presentación auditiva. Proporciona un modo natural de articular varias voces separadamente, como en la vida real (las personas raramente están todas en el mismo lugar de un cuarto). Los parlantes en estéreo producen sonidos laterales. Los auriculares bi-aurales o los equipos hogareños de 5 parlantes de creciente popularidad pueden generar un sonido completamente envolvente, y los equipos de múltiples parlantes pueden crear un verdadero sonido tridimensional. VRML 2.0 también incluye el audio espacial, lo que implica que en poco tiempo el equipamiento de audio espacial, a un precio accesible para el consumidor, estará más ampliamente disponible.

### 'azimuth'

|              |   |
|--------------|---|
| Valor:       | <ángulo>   [[ left-side   far-left   left   center-left   center   center-right   right   far-right   right-side ]    behind ]   leftwards   rightwards   inherit |
| Inicial:     | center  |
| Se aplica a: | todos los elementos   |
| Se hereda:   | si  |
| Porcentajes: | N/A   |
| Medios:      | auditivos   |

Los valores tienen los siguientes significados:

#### <ángulo>

La posición se describe en términos de un ángulo dentro del rango '-360deg' a '360deg'. El valor '0deg' significa adelante en el centro del espacio de sonido. '90deg' es a la derecha, '180deg' atrás y '270deg' (o, de modo equivalente y más conveniente, '-90deg') a la izquierda.

#### left-side

Igual a '270deg'. Con 'behind', '270deg'.

#### far-left

Igual a '300deg'. Con 'behind', '240deg'.

#### left

Igual a '320deg'. Con 'behind', '220deg'.

#### center-left

Igual a '340deg'. Con 'behind', '200deg'.

**center**  
Igual a '0deg'. Con 'behind', '180deg'.  
**center-right**  
Igual a '20deg'. Con 'behind', '160deg'.  
**right**  
Igual a '40deg'. Con 'behind', '140deg'.  
**far-right**  
Igual a '60deg'. Con 'behind', '120deg'.  
**right-side**  
Igual a '90deg'. Con 'behind', '90deg'.  
**leftwards**

Mueve el sonido a la izquierda, con relación al ángulo actual. Más precisamente, le resta 20 grados. La aritmética se lleva en módulos de 360 grados. Observe que 'leftwards' es más precisamente descripto como 'rotado en sentido contrario a las agujas del reloj', debido a que *siempre* resta 20 grados, aún si el azimuth heredado ya está detrás del oyente (en cuyo caso el sonido parece en realidad moverse a la derecha).

**rightwards**  
Mueve el sonido a la derecha, con relación al ángulo actual. Más precisamente, agrega 20 grados. Ver 'leftwards' por la aritmética.

Esta propiedad probablemente será implementada mezclando la misma señal en distintos canales a diferente volumen. Podría usarse también cambio de fase, pausa digital u otras técnicas para proporcionar la ilusión de un espacio de sonido. Los medios exactos para lograr este efecto y el número de parlantes utilizados para hacerlo dependen de la aplicación del usuario, esta propiedad sólo identifica el resultado final deseado.

```
H1 { azimuth: 30deg }
TD.a { azimuth: far-right } /* 60deg */
#12 { azimuth: behind far-right } /* 120deg */
P.comentario { azimuth: behind } /* 180deg */
```

Si el azimuth espacial es especificado y el dispositivo de salida no puede producir sonidos *detrás* de la posición del oyente, las aplicaciones del usuario deben convertir los valores en la retaguardia del hemisferio en valores hacia adelante del hemisferio. Un método es el siguiente:

- if 90deg < x <= 180deg then x := 180deg - x
- if 180deg < x <= 270deg then x := 540deg - x

**'elevation'**  
Valor: <ángulo> | below | level | above | higher | lower | inherit  
Inicial: level  
Se aplica a: todos los elementos  
Se hereda: si  
Porcentajes: N/A  
Medios: auditivos

Los valores para esta propiedad tienen los siguientes significados:

**<ángulo>**  
Especifica la elevación como un ángulo, entre '-90deg' y '90deg'. '0deg' significa el horizonte de enfrente, que vagamente significa a nivel del oyente. '90deg' significa directamente sobre la cabeza y '-90deg' significa directamente debajo.  
**below**  
Igual a '-90deg'.  
**level**  
Igual a '0deg'.  
**above**  
Igual a '90deg'.  
**higher**  
Agrega 10 grados a la elevación actual.  
**lower**  
Resta 10 grados a la elevación actual.

Los medios exactos para lograr este efecto y el número de parlantes utilizados para hacerlo son indefinidos. Esta propiedad sólo identifica el resultado final deseado.

```
H1 { elevation: above }
TR.a { elevation: 60deg }
TR.b { elevation: 30deg }
TR.c { elevation: level }
```

## 19.8 Propiedades de las características de la voz: 'speech-rate', 'voice-family', 'pitch', 'pitch-range', 'stress' y 'richness'

'speech-rate'

Valor: <número> | x-slow | slow | medium | fast | x-fast | faster | slower | inherit  
Inicial: medium  
Se aplica a: todos los elementos  
Se hereda: si  
Porcentajes: N/A  
Medios: auditivos

Esta propiedad especifica la velocidad de dicción. Observe que se permiten tanto palabras clave absolutas como relativas (compare con 'font-size'). Los valores tienen los siguientes significados:

- <número>  
Especifica la velocidad de dicción en palabras por minuto, una cantidad que varía un poco según el idioma pero no obstante es ampliamente soportada por los sintetizadores de voz.
- x-slow  
Igual a 80 palabras por minuto.
- slow  
Igual a 120 palabras por minuto.
- medium  
Igual a 180 - 200 palabras por minuto.
- fast  
Igual a 300 palabras por minuto.
- x-fast  
Igual a 500 palabras por minuto.
- faster  
Agrega 40 palabras por minuto a la velocidad de dicción actual.
- slower  
Resta 40 palabras por minuto a la velocidad de dicción actual.

'voice-family'

Value: [[<voz-especifica> | <voz-genérica> ],]\* [<voz-especifica> | <voz-genérica> ] | inherit  
Inicial: depende de la aplicación del usuario  
Se aplica a: todos los elementos  
Se hereda: si  
Porcentajes: N/A  
Medios: auditivos

El valor es una lista de prioridades separadas por comas, de nombres de familias de voces (compare con 'font-family'). Los valores tienen los siguientes significados:

- <voz-genérica>  
Los valores son familias de voces. Los posibles valores son 'male' (hombre), 'female' (mujer) y 'child' (niño).
- <voz-especifica>  
Los valores son instancias específicas (ej., comediante, carlos, lani).

```
H1 { voice-family: anunciador, male }  
P.part.romeo { voice-family: romeo, male }  
P.part.juliet { voice-family: julieta, female }
```

Los nombres de las voces específicas pueden ir entrecomilladas, y de hecho deben ir entre comillas si alguna de las palabras que conforman el nombre no están conformes a las reglas sintácticas de los [identificadores](#). También se recomienda entrecomillar las voces específicas con un nombre conformado por más de una palabra. Si las comillas son omitidas, cualquier carácter de [espacio en blanco](#) antes y después del nombre de la familia de voces es ignorado y cualquier secuencia de caracteres de espacios en blanco dentro del nombre de la familia de voces se convierte en un espacio simple.

'pitch'

Valor: <frecuencia> | x-low | low | medium | high | x-high | inherit  
Inicial: medium  
Se aplica : todos los elementos  
Se hereda: si  
Porcentajes: N/A  
Medios: auditivos

Especifica el tono promedio (o frecuencia) de la voz articulada. El tono promedio de una voz depende de la familia de voces. Por ejemplo, el tono promedio de una voz masculina normal está alrededor de los 120Hz, pero para una voz femenina, ronda los 210Hz.

Los valores tienen los siguientes significados:

- <frecuencia>  
Especifica en hertz (Hz) el tono promedio de la voz articulada.
- x-low, low, medium, high, x-high  
Estos valores no equivalen a frecuencias absolutas porque estos valores dependen de la familia de voces. Las aplicaciones del usuario deben hacer corresponder estos valores con las frecuencias apropiadas en base a la

familia de voces y al entorno del usuario. Sin embargo, las aplicaciones del usuario deben distribuir estos valores en orden (es decir, 'x-low' es una frecuencia más baja que 'low', etc.).

'pitch-range'

Valor: <número> | inherit  
Inicial: 50  
Se aplica a: todos los elementos  
Se hereda: si  
Porcentajes: N/A  
Medios: auditivos

Especifica la variación en el tono promedio. El tono percibido de una voz humana está determinado por la frecuencia básica y típicamente tiene un valor de 120Hz para una voz masculina y 210Hz para una voz femenina. Los idiomas humanos se hablan variando la inflexión y el tono; estas variaciones transmiten significados y énfasis adicionales. Así, una voz muy animada, es decir, una con muchas inflexiones, muestra una amplia gama de tonos. Esta propiedad especifica el rango dentro del cual se producen estas variaciones, es decir, cuánto se puede desviar la frecuencia básica del tono promedio.

Los valores tienen los siguientes significados:

<número>

Un valor entre '0' y '100'. Un rango de tonos de '0' produce voces monótonas, chatas. Un rango de tonos de 50 produce inflexiones normales. Los rangos de tonos mayores a 50 producen voces animadas.

'stress'

Valor: <número> | inherit  
Inicial: 50  
Se aplica a: todos los elementos  
Se hereda: si  
Porcentajes: N/A  
Medios: auditivos

Especifica la altura de los "picos" en el perfil de entonaciones de una voz. Por ejemplo, el inglés es un idioma **acentuado**, y las distintas partes de una oración tienen asignados énfasis primarios, secundarios y terciarios. El valor de 'stress' controla la cantidad de inflexiones que resultan de estos acentos. Esta propiedad es compañera de la propiedad 'pitch-range' y se proporciona para permitir a los diseñadores aprovechar los sistemas de recepción más fieles.

Los valores tienen los siguientes significados:

<número>

Un valor, entre '0' y '100'. El significado de los valores depende del idioma en que se habla. Por ejemplo, un nivel de '50' para una voz angloparlante masculina normal (tono promedio = 122Hz), hablando con entonación y énfasis normales tendría un significado distinto a '50' para una voz italiana.

'richness'

Valor: <número> | inherit  
Inicial: 50  
Se aplica a: todos los elementos  
Se hereda: si  
Porcentajes: N/A  
Medios: auditivos

Especifica la riqueza, o claridad, de la voz que habla. Una voz viva será "transportada" por un cuarto grande, una voz suave no. (El término "suave" se refiere a cómo se vé la forma de la onda cuando se dibuja.)

Los valores tienen los siguientes significados:

<número>

Un valor entre '0' y '100'. Mientras más alto es el valor, más será transportada la voz. Un valor más bajo producirá una voz meliflua, suave.

19.9 Propiedades del discurso: 'speak-punctuation' y 'speak-numeral'

Una propiedad del discurso adicional, [speak-header](#), se describe en el capítulo sobre [tablas](#)

'speak-punctuation'

Valor: code | none | inherit  
Inicial: none  
Se aplica a: todos los elementos  
Se hereda: si  
Porcentajes: N/A

*Medios:*        auditivos

Esta propiedad especifica cómo se manifiesta la puntuación. Los valores tienen los siguientes significados:

**code**

Los signos de puntuación como puntos y comas, llaves y demás se manifiestan literalmente.

**none**

Los signos de puntuación no se manifiestan, pero en cambio son procesados como distintas pausas.

**'speak-numeral'**

*Valor:*        digits | continuous | inherit

*Inicial:*       continuous

*Se aplica a:*   todos los elementos

*Se hereda:*    si

*Porcentajes:* N/A

*Medios:*        auditivos

Esta propiedad controla cómo se expresan los números. Los valores tienen los siguientes significados:

**digits**

Se expresan los números como dígitos individuales. Así, "237" es expresado "Dos Tres Siete".

**continuous**

Se expresan los números como un número completo. Así, "237" es expresado "doscientos treinta y siete". La representación en palabras depende del idioma.

# Apéndice A: Una hoja de estilo de ejemplo para HTML 4.0

*Este apéndice es informativo, no normativo.*

Esta hoja de estilo describe el formato típico de todos los elementos de [\(\[HTML40\]\)](#) en base a una extensa investigación de las prácticas de la AU actuales. Se insta a los programadores a usarla como una hoja de estilo predeterminada en sus implementaciones.

La representación completa de algunos elementos de HTML no puede ser expresada en CSS2, incluyendo elementos [reemplazados](#) (IMG, OBJECT), elementos de script (SCRIPT, APPLET), elementos de formularios de control y elementos de marcos.

|                  |   |
|------------------|---|
| ADDRESS,         |   |
| BLOCKQUOTE,      |   |
| BODY, DD, DIV,   |   |
| DL, DT,          |   |
| FIELDSET, FORM,  |   |
| FRAME, FRAMESET, |   |
| H1, H2, H3, H4,  |   |
| H5, H6, IFRAME,  |   |
| NOFRAMES,        |   |
| OBJECT           | { display: inline }   |
| OL, P, UL        |   |
| APPLET           | { display: inline }   |
| CENTER, DIR,     |   |
| HR, MENU, PRE    | { display: block }  |
| LI               | { display: list-item }                                      |
| HEAD             | { display: none }   |
| TABLE            | { display: table }  |
| TR               | { display: table-row }                                      |
| THEAD            | { display: table-header-group }                             |
| TBODY            | { display: table-row-group }                                |
| TFOOT            | { display: table-footer-group }                             |
| COL              | { display: table-column }                                   |
| COLGROUP         | { display: table-column-group }                             |
| TD, TH           | { display: table-cell }                                     |
| CAPTION          | { display: table-caption }                                  |
| TH               | { font-weight: bolder; text-align: center }                 |
| CAPTION          | { text-align: center }                                      |
| BODY             | { padding: 8px; line-height: 1.12 }                         |
| H1               | { font-size: 2em; margin: .67em 0 }                         |
| H2               | { font-size: 1.5em; margin: .83em 0 }                       |
| H3               | { font-size: 1.17em; margin: 1em 0 }                        |
| H4, P,           |   |
| BLOCKQUOTE, UL,  |   |
| FIELDSET, FORM,  |   |
| OL, DL, DIR,     |   |
| MENU             | { margin: 1.33em 0 }  |
| H5               | { font-size: .83em; line-height: 1.17em; margin: 1.67em 0 } |
| H6               | { font-size: .67em; margin: 2.33em 0 }                      |
| H1, H2, H3, H4,  |   |
| H5, H6, B,       |   |
| STRONG           | { font-weight: bolder }                                     |
| BLOCKQUOTE       | { margin-left: 40px; margin-right: 40px }                   |
| I, CITE, EM,     |   |
| VAR, ADDRESS     | { font-style: italic }                                      |
| PRE, TT, CODE,   |   |
| KBD, SAMP        | { font-family: monospace }                                  |
| PRE              | { white-space: pre }  |
| BIG              | { font-size: 1.17em }                                       |
| SMALL, SUB, SUP  | { font-size: .83em }  |
| SUB              | { vertical-align: sub }                                     |
| SUP              | { vertical-align: super }                                   |
| S, STRIKE, DEL   | { text-decoration: line-through }                           |
| HR               | { border: 1px inset }                                       |
| OL, UL, DIR,     |   |
| MENU, DD         | { margin-left: 40px }                                       |
| OL               | { list-style-type: decimal }                                |
| OL UL, UL OL,    |   |



```

UL UL, OL OL      { margin-top: 0; margin-bottom: 0 }
U, INS            { text-decoration: underline }
CENTER            { text-align: center }
BR:before         { content: "\A" }

/* Un ejemplo de estilos para los elementos ABBR/ACRONYM de HTML 4.0 */

ABBR, ACRONYM     { font-variant: small-caps; letter-spacing: 0.1em }
A[href]           { text-decoration: underline }
:focus            { outline: thin dotted invert }

/* Comienzo de la configuración de bidireccionalidad (no cambiar) */
BDO[DIR="ltr"]    { direction: ltr; unicode-bidi: bidi-override }
BDO[DIR="rtl"]    { direction: rtl; unicode-bidi: bidi-override }

*[DIR="ltr"]      { direction: ltr; unicode-bidi: embed }
*[DIR="rtl"]      { direction: rtl; unicode-bidi: embed }

/* Elementos que son a nivel de bloque en HTML4 */
ADDRESS, BLOCKQUOTE, BODY, DD, DIV, DL, DT, FIELDSET,
FORM, FRAME, FRAMESET, H1, H2, H3, H4, H5, H6, IFRAME,
NOSCRIPT, NOFRAMES, OBJECT, OL, P, UL, APPLET, CENTER,
DIR, HR, MENU, PRE, LI, TABLE, TR, THEAD, TBODY, TFOOT,
COL, COLGROUP, TD, TH, CAPTION
                        { unicode-bidi: embed }
/* Fin de la configuración bidi */

@media print {
    @page            { margin: 10% }
    H1, H2, H3,
    H4, H5, H6      { page-break-after: avoid; page-break-inside: avoid }
    BLOCKQUOTE,
    PRE              { page-break-inside: avoid }
    UL, OL, DL      { page-break-before: avoid }
}

@media speech {
    H1, H2, H3,
    H4, H5, H6      { voice-family: paul, male; stress: 20; richness: 90 }
    H1               { pitch: x-low; pitch-range: 90 }
    H2               { pitch: x-low; pitch-range: 80 }
    H3               { pitch: low; pitch-range: 70 }
    H4               { pitch: medium; pitch-range: 60 }
    H5               { pitch: medium; pitch-range: 50 }
    H6               { pitch: medium; pitch-range: 40 }
    LI, DT, DD       { pitch: medium; richness: 60 }
    DT               { stress: 80 }
    PRE, CODE, TT     { pitch: medium; pitch-range: 0; stress: 0; richness: 80 }
    EM               { pitch: medium; pitch-range: 60; stress: 60; richness: 50 }
}
    STRONG           { pitch: medium; pitch-range: 60; stress: 90; richness: 90 }
}
    DFN              { pitch: high; pitch-range: 60; stress: 60 }
    S, STRIKE         { richness: 0 }
    I                 { pitch: medium; pitch-range: 60; stress: 60; richness: 50 }
}
    B                 { pitch: medium; pitch-range: 60; stress: 90; richness: 90 }
}
    U                 { richness: 0 }
    A:link             { voice-family: harry, male }
    A:visited          { voice-family: betty, female }
    A:active           { voice-family: betty, female; pitch-range: 80; pitch: x-
high }
}

```

# Apéndice B: Cambios con respecto a CSS1

*Este apéndice es informativo, no normativo.*

CSS2 se cimenta en CSS1 y todas las hojas de estilo CSS1 válidas son hojas de estilo CSS2 válidas. Los cambios entre la especificación CSS1 (ver [\[CSS1\]](#)) y esta especificación caben en tres grupos: nuevo funcionamiento, descripciones actualizadas del funcionamiento de CSS1 y cambios a CSS1.

## B.1 Nuevo funcionamiento

Además del funcionamiento de CSS1, CSS2 soporta:

- El concepto de [tipos de medios](#).
- El valor `'inherit'` para todas las propiedades.
- [Medios paginados](#)
- [Hojas de estilo auditivas](#)
- Algunas características de internacionalización, incluyendo [estilos de numeración de listas](#), soporte para [texto bidireccional](#) y soporte para [comillas](#) sensibles al idioma.
- Un mecanismo extendido de [selección de fuentes](#), incluyendo equivalencia inteligente, síntesis y fuentes descargables. También, se ha introducido el concepto de fuentes del sistema y una nueva propiedad, `'font-size-adjust'`, ha sido agregada.
- [Tablas](#), incluyendo nuevos valores en `'display'` y `'vertical-align'`.
- [Posicionamiento relativo](#) y [absoluto](#), incluyendo [posicionamiento fijo](#).
- Nuevos tipos de cajas (junto con las de bloque y de línea): [compact](#) y [run-in](#).
- La capacidad de controlar el [desbordamiento](#), [recorte](#) y [visibilidad](#) del contenido en el modelo de formato visual.
- La capacidad de especificar [anchos](#) y [alturas](#) mínimos y máximos en el modelo de formato visual.
- Un mecanismo [selector](#) extendido, incluyendo selectores de hijos, selectores de adyacentes y selectores de atributos.
- [Contenido generado](#), [contadores](#) y [numeración automática](#), y [marcadores](#).
- Texto con sombra, a través de la nueva propiedad `'text-shadow'`.
- Algunas nuevas [pseudo-clases](#), `:first-child`, `:hover`, `:focus`, `:lang`.
- [Colores](#) y [fuentes](#) del sistema.
- [Cursores](#).
- [Contornos dinámicos](#).

## B.2 Descripciones actualizadas

La especificación CSS1 era breve y concisa. Esta especificación es mucho más voluminosa y más legible. Mucho del contenido adicional describe el nuevo funcionamiento, pero la descripción de las características de CSS1 ha sido también extendida. Excepto en los pocos casos descritos abajo, las descripciones revisadas no indican un cambio en la sintaxis ni en la semántica.

## B.3 Cambios semánticos a CSS1

Mientras que todas las hojas de estilo CSS1 son hojas de estilo CSS2 válidas, hay unos pocos casos donde la hoja de estilo CSS1 tendrá un significado distinto cuando sea interpretada como una hoja de estilo CSS2. La mayoría de los cambios se deben a la implementación en la especificación de la experiencia acumulada, pero también hay alguna corrección de errores.

- El significado de `"!important"` ha sido cambiado. En CSS1, `"!important"` en una hoja de estilo del autor tenía preponderancia por sobre una especificada en una hoja de estilo del usuario. Esto ha sido invertido en CSS2.
- En CSS2, los [valores para color](#) son recortados con respecto a la gama del dispositivo, no con respecto a la gama sRGB como en CSS1.
- CSS1 simplemente decía que `'margin-right'` era ignorado si tanto `'margin-left'` como `'width'` estaban establecidos. En CSS2 la elección entre `'margin-right'` o `'margin-left'` depende de la dirección de escritura.
- En CSS1, varias propiedades (ej., `'padding'`) tenía valores referidos al ancho del elemento padre. Esto era un error; el valor debe siempre referirse al ancho del elemento a nivel de bloque y esta especificación refleja esto introduciendo el término "bloque de contención".
- El valor inicial de `'display'` es `'inline'` en CSS2, no `'block'` como en CSS1.
- En CSS1, la propiedad `'clear'` se aplicaba a todos los elementos. Esto era un error, y la propiedad sólo se aplica a los elementos a nivel de bloque en CSS2.
- En CSS1, `':link'`, `':visited'` y `':active'` eran mutuamente excluyentes; en CSS2, `':active'` puede aparecer junto con `':link'` o `':visited'`.
- El factor de escala sugerido entre índices adyacentes de `'font-size'` en la tabla de tamaños de la fuente ha sido reducido de 1.5 a 1.2.
- El valor computado, no es valor real, de `'font-size'` es heredado.
- La descripción en CSS1 de `'inside'` (para `'list-style-position'`) posibilitaba la interpretación de que el margen izquierdo del texto era afectado, en lugar de la posición de la viñeta. En CSS2 esa interpretación está corregida.
- Por favor, también consulte la sección *normativa* en [diferencias entre los comandos de CSS1 y CSS2](#).

# Apéndice C: Implementación y notas de desempeño de las fuentes

Este apéndice es informativo, no normativo.

## C.1 Glosario de términos relacionados con las fuentes

### Caché de fuentes

La *caché de fuente* permite una copia temporal de las fuentes en el sistema cliente. A menudo se guardan en el disco con otros elementos como los gráficos especificados por la AU.

### DocLock™

La tecnología *DocLock™* de Bitstream garantiza que los PFR de TrueDoc sólo puedan utilizarse con el sitio para el que fueron publicadas. Un PFR de TrueDoc trasladado a un sitio diferente o referenciado desde otro sitio no funcionará.

### Equivalencia de fuentes

La *equivalencia de fuentes* es un proceso de selección de fuentes similares en base a la utilización de uno o más atributos de la fuente primaria. Los atributos comunes incluyen serif, sans-serif, peso, altura de las mayúsculas, altura de la x, espaciado, lenguaje y variantes. La equivalencia de fuentes depende del algoritmo y de la variedad de las fuentes candidatas.

### Firma digital

Parte de una tecnología de administración empresarial, utilizada para proporcionar contratos seguros.

### Fuente

Un "manejador" que se refiere a una fuente específica, excluyendo el tamaño de la fuente.

### Fuente de servidor

Una *fente de servidor* es un recurso de fuente localizado en un servidor web que es referenciado por la definición de la Fuente Web. La aplicación del usuario puede usar este recurso para procesar la página.

### Fuentes TrueType GX

Las *Fuentes TrueType GX* contienen extensiones al estándar del formato TrueType que permiten fuentes variables, similares a las fuentes Multiple Master. Pueden haber varios ejes de variación tales como peso, altura o inclinación. Los ejes puede definirse para obtener casi cualquier efecto. TrueType GX también puede soportar la sustitución en la representación de los signos para las ligaduras, formas contextuales, fracciones, etc. A la fecha, TrueType GX está disponible sólo en la Mac (ver [\[TRUETYPEGX\]](#)).

### Fuente Type 1

La fuente *Type 1*, desarrollada por Adobe Systems, fue uno de los primeros formatos escalables disponibles. Las fuentes Type 1 generalmente contienen 228 caracteres con las representaciones de los signos descriptas usando curvas bezier de tercer grado. Mac, Windows y X tiene formatos similares pero separados; Abobe proporciona el Adobe Type Manager para las tres plataformas. Type1c es una más reciente forma de almacenamiento comprimido sin pérdida para las representaciones de los signos de Type 1.

### Infinifont

Una técnica de sintetizado de fuentes que, dado un número Panose-1 (y, opcionalmente, datos adicionales de descripción de la fuente) puede generar una fuente sin extrapolación de un único contorno maestro o interpolación entre dos o más contornos (ver [\[INFINIFONT\]](#)).

### Intellifont

La tecnología Intellifont fue desarrollada por Agfa y es el formato nativo para Hewlett-Packard y otras impresoras que usan el lenguaje PCL5. También es el formato de fuente nativo de las computadoras Amiga.

### Itálica

Un tipo de letra en las escrituras latinas que es más cursivo que las letras romanas pero menos que las letras manuscritas. A menudo, un par de fuentes son diseñadas para utilizarse en conjunto; una es una romana con serif y una es itálica. Otros términos para describir esta clase de letra incluye cursiva y, en la escritura cirílica, kursiv. Para los tipos sans-serif, el tipo acompañante es a menudo una variante inclinada u oblicua antes que un tipo diferente.

Un tipo más cursivo que el romano, per

Un tipo más cursivo que el romano, per

Un tipo más cursivo que el romano, per

Un tipo más cursivo que el romano, per

Un tipo más cursivo que el romano, p

Un tipo más cursivo que el romano, p

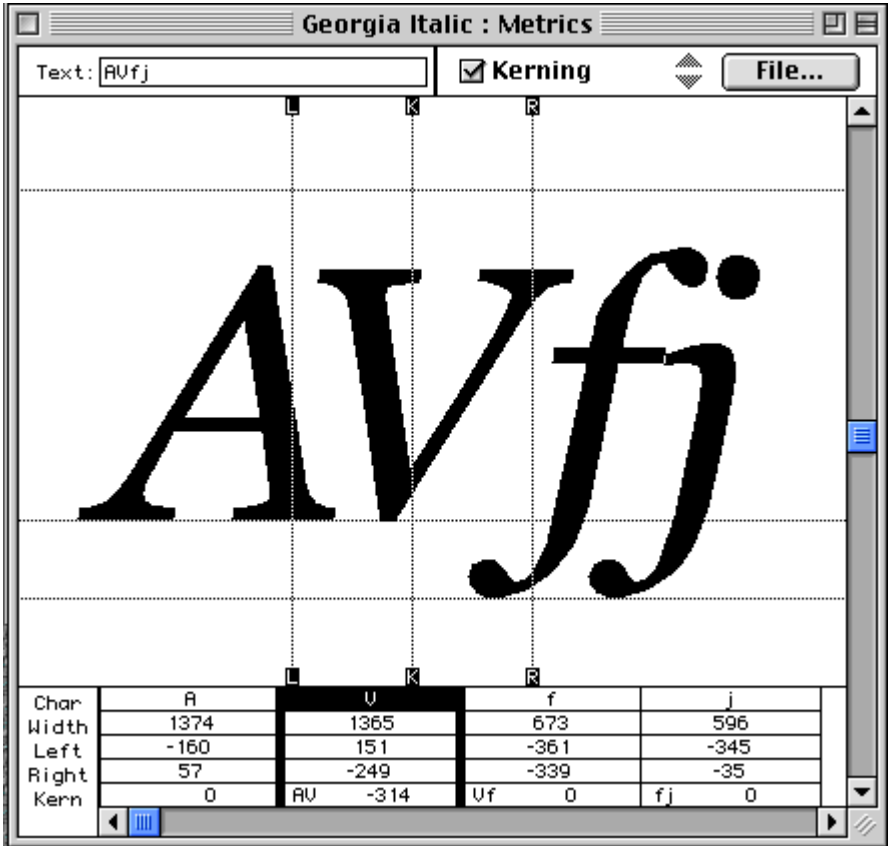
Un tipo más cursivo que el romano, per

Un tipo más cursivo que el romano, pero

[\[D\]](#)

### Kerning

Alteración del espacio entre las representaciones de signos seleccionados, que de otro modo podrían parecer demasiado juntos o demasiado separados, para obtener un color tipográfico más uniforme.



[D]

### Limitación de un URI

Un proceso de bloqueo de un recurso de fuente particular en un sitio Web determinado por la incrustación de un URI encriptado o el uso de firma digital en un recurso de fuente.

### Multiple Master Font

Una *Fuente Multiple Master* contiene dos fuentes primarias que son usadas con un programa especial de procesamiento que proporciona un resultado interpolado. Adobe Systems provee un mecanismo que proporciona los parámetros a ser utilizados para controlar la salida o la fuente de salida interpolada. Estos parámetros describen generalmente las características de una fuente original y la múltiples variantes resultantes son referenciadas como una "fuentes sintetizadas".

### Open Type

OpenType es una extensión del formato de fuentes TrueType que contiene información adicional que extiende las capacidades de las fuentes para soportar tipografía internacional de alta calidad. Open Type puede asociar un simple carácter con múltiples representaciones del signo, y combinaciones de caracteres en una única representación del signo (formación de ligaduras). Open Type incluye información en dos dimensiones para soportar características de posicionamiento complejo y vinculación de signos. TrueType Open y OpenType contienen información explícita acerca de la escritura y el lenguaje, de modo que una aplicación de procesamiento de texto puede ajustar su comportamiento adecuadamente (ver [OPENTYPE](#)).

### Reducción en la representación de signos

La *reducción en la representación de signos* es el proceso por el cual las representaciones de signos no deseados (junto con su información sobre espaciado lateral y kerning) son eliminadas de una fuente primaria para producir una subconjunto más reducido de fuentes que cubre un documento particular o un juego de documentos. Esto es particularmente conveniente para documentos con escrituras ideográficas, donde la cantidad de signos complementarios a la fuente de base puede ser muy grande. La reducción en la representación de signos para documentos que usan escrituras que requieren ligaduras, como el Arabe, es difícil si no se conocen las reglas de formación de las ligaduras del sistema de visualización final.

### Speedo

La tecnología de fuente *Speedo* fue desarrollada por Bitstream y es el formato de fuente nativo de las computadoras Atari ST y Falcon. Es también usada por las computadoras corriendo el sistema X window.

### TrueDoc

La tecnología *TrueDoc* fue desarrollada por Bitstream para la creación, transporte y representación de objetos de fuente escalable, independientes de la plataforma, en la web. La creación de objetos de fuente es realizada por la grabadora de figuras de caracteres (CSR) y el procesado de los objetos de fuente es realizado por el reproductor de figuras de caracteres (CSP) de TrueDoc. La tecnología está proyectada para ser usada en la web para la visualización e impresión.

### TrueDoc Portable Font Resource

Un Recurso de fuente portable TrueDoc (*TrueDoc Portable font resource* o **PFR**) es un objeto de fuente escalable independiente de la plataforma que es producido por un reproductor de figuras de caracteres. La entrada puede ser TrueType o Type 1 de cualquier tipo en Windows, Mac o Unix. Los recursos de fuente portables TrueDoc proporcionan buen índice de compresión, son independientes de la plataforma y debido a que no están en un formato nativo de fuentes (TrueType or Type 1) no pueden ser fácilmente instalados.

### TrueType

*TrueType* es un formato de fuente desarrollado por Apple y licenciado a Microsoft. TrueType es el formato de fuente del sistema operativo nativo para Windows y Macintosh. TrueType contiene un conjunto jerárquico de tablas y representaciones de signos. Los caracteres pueden ser indicados en base al tamaño de puntos y por carácter proporcionando excelente calidad a la resolución de pantalla. Las fuentes TrueType para Windows y Mac tienen pocas diferencias, aunque pueden ser lo suficientemente distintas como para impedir el uso entre plataformas.

### TrueType Collection

Una Colección TrueType (*TrueType Collection* o **TTC**) es una extensión al formato TrueType que incluye tablas que permiten que múltiples fuentes TrueType estén contenidas dentro de un sólo archivo de fuente TrueType. Los archivos de colecciones TrueType son relativamente raros en estos momentos.

## C.2 Recuperación de las fuentes

Hay muchos formatos distintos de fuentes usados en muchas plataformas distintas. Para seleccionar un formato de fuente de preferencia, se utiliza la negociación transparente de contenido (ver [\[NEGOT\]](#)). Siempre es posible decir cuándo una fuente ha sido referenciada, porque el URI está en la descripción de la fuente. Una determinada implementación conocerá qué formato de fuente descargable soporta y podrá así utilizar el formato indicado para evitar descargar fuentes en un formato no soportado.

## C.3 Significado de los números Panose

'OS/2' table of Georgia Italic

Page 2 of 2

PANOSE Classification:

|            |                       |                   |                   |
|------------|-----------------------|-------------------|-------------------|
| Family:    | Text and Display      | Serif Style:      | Square Cove       |
| Weight:    | Book                  | Proportion:       | Old Style         |
| Contrast:  | Medium Low            | Stroke Variation: | Gradual/Vertical  |
| Arm Style: | Straight/Single Serif | Letterform:       | Oblique/Contact   |
| Midline:   | Standard/Pointed      | XHeight:          | Constant/Standard |

|                       |      |                        |       |
|-----------------------|------|------------------------|-------|
| Vendor Id:            | MSO  | fsSelection:           | 1     |
| First Char Index:     | 32   | Last Char Index:       | 64258 |
| Typographic Ascender: | 1549 | Typographic Descender: | -444  |
| Typographic Line Gap: | 198  |                        |       |
| Windows™ Ascent:      | 1878 | Windows™ Descent:      | 449   |

Prev Page

Cancel

OK

[D]

La familia, estilo del serif y números de proporción son usados por Windows 95 para la selección y equivalencia de fuentes.

El significado de los diez números y los valores permitidos (dados entre paréntesis) se dan abajo para el más común de los casos, donde el dígito de "family" es 2, *Texto y Display*. (Si el primer dígito tiene un valor diferente, los restantes nueve dígitos tienen significados distintos). Por detalles más amplios sobre Panose-1, ver [\[PANOSE\]](#).

### Family

- Any (0)
- No Fit (1)
- [\[PANOSE\]](#) Latin Text and Display (2)
- [\[PANOSE\]](#) Latin Script (3)
- [\[PANOSE\]](#) Latin Decorative (4)
- [\[PANOSE\]](#) Latin Pictorial (5)

### Serif Style

- Any (0)
- No Fit (1)
- Cove (2)
- Obtuse Cove (3)
- Square Cove (4)
- Obtuse Square Cove (5)
- Square (6)
- Thin (7)
- Bone (8)
- Exaggerated (9)
- Triangle (10)
- Normal Sans (11)
- Obtuse Sans (12)
- Perp Sans (13)
- Flared (14)
- Rounded (15)

Weight

- Any (0)
- No Fit (1)
- Very Light (2)[100]
- Light (3) [200]
- Thin (4) [300]
- Book (5) [400] *same as CSS1 'normal'*
- Medium (6) [500]
- Demi (7) [600]
- Bold (8) [700] *same as CSS1 'bold'*
- Heavy (9) [800]
- Black (10) [900]
- Extra Black / Nord (11) [900] *force mapping to CSS1 100-900 scale*

Proportion

- Any (0)
- No Fit (1)
- Old Style (2)
- Modern (3)
- Even Width (4)
- Expanded (5)
- Condensed (6)
- Very Expanded (7)
- Very Condensed (8)
- Monospaced (9)

Contrast

- Any (0)
- No Fit (1)
- None (2)
- Very Low (3)
- Low (4)
- Medium Low (5)
- Medium (6)
- Medium High (7)
- High (8)
- Very High (9)

Stroke Variation

- Any (0)
- No Fit (1)
- No Variation (2)
- Gradual/Diagonal (3)
- Gradual/Transitional (4)
- Gradual/Vertical (5)
- Gradual/Horizontal (6)
- Rapid/Vertical (7)
- Rapid/Horizontal (8)
- Instant/Horizontal (9)
- Instant/Vertical (10)

Arm Style

- Any (0)
- No Fit (1)
- Straight Arms/Horizontal (2)
- Straight Arms/Wedge (3)
- Straight Arms/Vertical (4)
- Straight Arms/Single Serif (5)
- Straight Arms/Double Serif (6)
- Non-Straight Arms/Horizontal (7)
- Non-Straight Arms/Wedge (8)
- Non-Straight Arms/Vertical 90
- Non-Straight Arms/Single Serif (10)
- Non-Straight Arms/Double Serif (11)

Letterform

- Any (0)
- No Fit (1)
- Normal/Contact (2)

- Normal/Weighted (3)
- Normal/Boxed (4)
- Normal/Flattened (5)
- Normal/Rounded (6)
- Normal/Off Center (7)
- Normal/Square (8)
- Oblique/Contact (9)
- Oblique/Weighted (10)
- Oblique/Boxed (11)
- Oblique/Flattened (12)
- Oblique/Rounded (13)
- Oblique/Off Center (14)
- Oblique/Square (15)

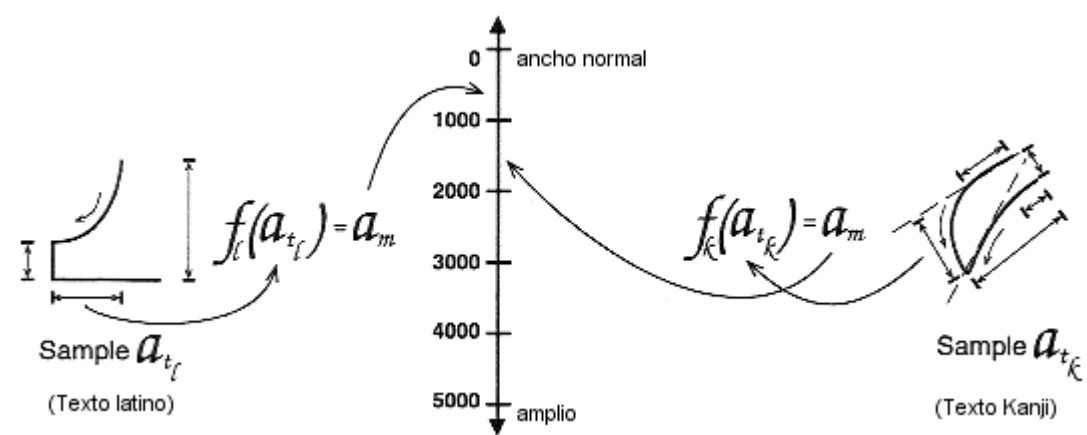
Midline

- Any (0)
- No Fit (1)
- Standard/Trimmed (2)
- Standard/Pointed (3)
- Standard/Serifed (4)
- High/Trimmed (5)
- High/Pointed (6)
- High/Serifed (7)
- Constant/Trimmed (8)
- Constant/Pointed (9)
- Constant/Serifed (10)
- Low/Trimmed (11)
- Low/Pointed (12)
- Low/Serifed (13)

XHeight

- Any (0)
- No Fit (1)
- Constant/Small (2)
- Constant/Standard (3)
- Constant/Large (4)
- Ducking/Small (5)
- Ducking/Standard (6)
- Ducking/Large (7)

Panose-2 (ver [PANOSE2](#)) es una especificación para una más comprensible tecnología de clasificación y equivalencia de fuentes que no está limitada a la tipografía latina. Por ejemplo, las características del serif de un tipo latino pueden ser comparadas con la terminación del trazo de un tipo Kanji.



El valor en Panose-2 no es almacenado dentro de ningún formato de fuente conocido, pero puede ser medido.

### C.4 Deducción de rangos de Unicode para TrueType

Esta información está disponible en la fuente mirando los bits 'ulUnicodeRange' en la tabla 'OS/2' (si hay alguna), que contiene una representación en un campo de bits del conjunto. Esta tabla está definida en la revisión 1.66 de la especificación TrueType, de Microsoft. Considerando esta información como un conjunto, cada elemento corresponde a bloque de caracteres de Unicode 1.1, y la presencia de ese elemento en el conjunto indica que la fuente tiene uno o más representaciones de signos para representar al menos un carácter en ese bloque. El conjunto tiene 128 elementos como se describe abajo. El orden generalmente sigue al standard de Unicode 1.1. Esta tabla puede ser utilizada para convertir la información en la fuente TrueType en un descriptor 'unicode-range' de CSS.



| Block Add |   | Block name   | Unicode range     |
|-----------|---|--|-------------------|
| 0         | 1 | Basic Latin  | U+0-7F            |
| 1         | 2 | Latin-1 Supplement                                   | U+80-FF           |
| 2         | 4 | Latin-1 Extended-A                                   | U+100-17F         |
| 3         | 8 | Latin Extended-B                                     | U+180-24F         |
| 4         | 1 | IPA Extensions                                       | U+250-2AF         |
| 5         | 2 | Spacing Modifier Letters                             | U+2B0-2FF         |
| 6         | 4 | Combining Diacritical Marks                          | U+300-36F         |
| 7         | 8 | Greek  | U+370-3CF         |
| 8         | 1 | <i>Greek Symbols and Coptic</i>                      | U+3D0-3EF         |
| 9         | 2 | Cyrillic   | U+400-4FF         |
| 10        | 4 | Armenian   | U+530-58F         |
| 11        | 8 | Hebrew   | U+590-5FF         |
| 12        | 1 | <i>Hebrew Extended-A</i><br><i>Hebrew Extended-B</i> | ?? what ranges ?? |
| 13        | 2 | Arabic   | U+600-69F         |
| 14        | 4 | <i>Arabic Extended</i>                               | U+670-6FF         |
| 15        | 8 | Devanagari   | U+900-97F         |
| 16        | 1 | Bengali  | U+980-9FF         |
| 17        | 2 | Gurmukhi   | U+A00-A7F         |
| 18        | 4 | Gujarati   | U+A80-AFF         |
| 19        | 8 | Oriya  | U+B00-B7F         |
| 20        | 1 | Tamil  | U+B80-BFF         |
| 21        | 2 | Telugu   | U+C00-C7F         |
| 22        | 4 | Kannada  | U+C80-CFF         |
| 23        | 8 | Malayalam  | U+D00-D7F         |
| 24        | 1 | Thai   | U+E00-E7F         |
| 25        | 2 | Lao  | U+E80-EFF         |
| 26        | 4 | Georgian   | U+10A0-10EF       |
| 27        | 8 | <i>Georgian Extended</i>                             | U+10F0-10FF ??    |
| 28        | 1 | Hangul Jamo  | U+1100-11FF       |
| 29        | 2 | Latin Extended Additional                            | -                 |
| 30        | 4 | Greek Extended                                       | U+1F00-1FFF       |
| 31        | 8 | General Punctuation                                  | U+2000-206F       |
| 32        | 1 | Superscripts and Subscripts                          | -                 |
| 33        | 2 | Currency Symbols                                     | U+20A0-20CF       |
| 34        | 4 | Combining Marks for Symbols                          | U+20D0-20FF       |
| 35        | 8 | Letterlike Symbols                                   | U+2100-214F       |
| 36        | 1 | Number Forms   | U+2150-218F       |
| 37        | 2 | Arrows   | U+2190-21FF       |
| 38        | 4 | Mathematical Operators                               | U+2200-22FF       |
| 39        | 8 | Miscellaneous Technical                              | U+2300-23FF       |
| 40        | 1 | Control Pictures                                     | U+2400-243F       |
| 41        | 2 | Optical Character Recognition                        | U+2440-245F       |
| 42        | 4 | Enclosed Alphanumerics                               | U+2460-24FF       |

| Block | Add | Block name                      | Unicode range |
|-------|-----|---------------------------------|---------------|
| 43    | 8   | Box Drawing                     | U+2500-257F   |
| 44    | 1   | Block Elements                  | U+2580-259F   |
| 45    | 2   | Geometric Shapes                | U+25A0-25FF   |
| 46    | 4   | Miscellaneous Symbols           | U+2600-26FF   |
| 47    | 8   | Dingbats                        | U+2700-27BF   |
| 48    | 1   | CJK Symbols and Punctuation     | U+3000-303F   |
| 49    | 2   | Hiragana                        | U+3040-309F   |
| 50    | 4   | Katakana                        | U+30A0-30FF   |
| 51    | 8   | Bopomofo                        | U+3100-312F   |
| 52    | 1   | Hangul Compatibility Jamo       | U+3130-318F   |
| 53    | 2   | CJK Miscellaneous               | ??            |
| 54    | 4   | Enclosed CJK Letters and Months | U+3200-32FF   |
| 55    | 8   | CJK compatibility               | U+3300-33FF   |
| 56    | 1   | Hangul                          | U+AC00-D7FF   |
| 59    | 8   | CJK Unified Ideographs          | U+4E00-9FFF   |
| 60    | 1   | Private Use Area                | U+E000-F8FF   |
| 61    | 2   | CJK Compatibility Ideographs    | U+F900-FAFF   |
| 62    | 4   | Alphabetic Presentation Forms   | U+FB00-FB4F   |
| 63    | 8   | Arabic Presentation Forms-A     | U+FB50-FDFF   |
| 64    | 1   | Combining Half Marks            | U+FE20-FE2F   |
| 65    | 2   | CJK compatibility Forms         | U+FE30-FE4F   |
| 66    | 4   | Small Form Variants             | U+FE50-FE6F   |
| 67    | 8   | Arabic Presentation Forms-B     | U+FE70-FEFF   |
| 68    | 1   | Halfwidth and Fullwidth Forms   | U+FF00-FFEF   |
| 69    | 2   | Specials                        | U+FFF0-FFFD   |

El sistema de campo de bits de TrueType tiene el problema que está ligado a Unicode 1.1 y es incapaz de cubrir la expansión de Unicode -no puede representar el tibetano por ejemplo, u otras escrituras introducidas con Unicode 2.0 o revisiones posteriores-.

### C.5 Generación automática de descriptores

Las herramientas de autor deben permitir a los autores de hojas de estilo agregar y editar los descriptores de fuentes. En algunos casos, sin embargo, las herramientas de autor pueden ayudar examinando las fuentes instaladas localmente y automáticamente generar descriptores para las fuentes referenciadas en la hoja de estilo. Esta también es una función que puede ser llevada a cabo por herramientas que formen subconjuntos o conviertan fuentes preparadas para su transmisión dinámica.

Esta tabla sugiere donde puede encontrarse tal información, para los formatos de fuente más comunes.

| Descriptor      | Type 1                                    | TrueType and OpenType  | TrueType GX <a href="#">[TRUETYPEGX]</a> |
|-----------------|---|--|--|
| 'ascent'        | 'Ascender' in AFM/PFM file                | 'Ascender' in 'hhea' table or (preferably) 'sTypoAscender' in 'OS/2' table | 'horizontalBefore' in 'fmtx' table       |
| 'baseline'      |   |  | bsln table, see <a href="#">note</a>     |
| 'bbox'          | FontBBox, font dictionary                 | 'xMin', 'xMax', 'yMin' and 'yMax' entries of the 'head' table              |  |
| 'cap-height'    | CapHeight in AFM/PFM file                 |  |  |
| 'descent'       | 'Descender' in the AFM/PFM file.          |  |  |
| 'mathline'      |   |  | bsln table                               |
| 'font-family'   | FamilyName, fontinfo dictionary           | name table   |  |
| 'stemh'         | StdHW, private dictionary of AFM/PFM file |  |  |
| 'stemv'         | /StdVW, private dictionary                | cvt table  |  |
| 'topline'       |   |  | bsln table                               |
| 'unicode-range' | cmap file                                 | OS/2 table, ver <a href="#">Apéndice C</a>                                 |  |
| 'units-per-em'  | FontMatrix, font dictionary               | unitsPerEm, head table.  |  |
| 'widths'        |   | hmtx table   |  |

- Dentro de bsln table, la ideographic centered baseline puede ser usada para la alineación de caracteres predominantemente ideográficos y la ideographic low baseline es más adecuada para caracteres ideográficos en una ejecución de caracteres predominantemente latinos, griegos o cirílicos.

# Apéndice D: La gramática de CSS2

Este apéndice es normativo.

**Nota de la traducción:**  
Este apéndice es normativo, por lo que recuerdo a los programadores que la única versión aceptada está en <http://www.w3.org/TR/1998/REC-CSS2-19980512> por lo que deberán recurrir a ella si desean asegurarse el cumplimiento de las condiciones de [conformidad](#).

La gramática de abajo define la sintaxis de CSS2. Sin embargo, es en cierto sentido un superconjunto de CSS2 puesto que impone restricciones semánticas adicionales no expresadas en esta gramática. Una aplicación del usuario conformada debe también adherir a las [reglas de análisis con compatibilidad futura](#), la [notación de propiedades y valores](#) y la unidad de notación. Además, el lenguaje del documento puede imponer restricciones, por ej., HTML impone restricciones a los posibles valores del atributo "class".

## D.1 Gramática

La gramática de abajo es LL(1) (pero observe que la mayoría de la AU no debe utilizarla directamente, debido a que no expresa las [convenciones de análisis](#), sólo la sintaxis de CSS2). El formato de las producciones está optimizado para el uso humano y se usa alguna notación abreviadas más allá de Yacc (ver [YACC](#)):

- \*: 0 o más
- +: 1 o más
- ?: 0 o 1
- |: separa alternativas
- []: agrupa

Las producciones son:

```
stylesheet
: [ CHARSET_SYM S* STRING S* ';' ]?
  [S|CDO|CDC]* [ import [S|CDO|CDC]* ]*
  [ [ ruleset | media | page | font_face ] [S|CDO|CDC]* ]*
;
import
: IMPORT_SYM S*
  [STRING|URI] S* [ medium [ ',' S* medium]* ]? ';' S*
;
media
: MEDIA_SYM S* medium [ ',' S* medium ]* '{' S* ruleset* '}' S*
;
medium
: IDENT S*
;
page
: PAGE_SYM S* IDENT? pseudo_page? S*
  '{' S* declaration [ ';' S* declaration ]* '}' S*
;
pseudo_page
: ':' IDENT
;
font_face
: FONT_FACE_SYM S*
  '{' S* declaration [ ';' S* declaration ]* '}' S*
;
operator
: '/' S* | ',' S* | /* vacío */
;
combinator
: '+' S* | '>' S* | /* vacío */
;
unary_operator
: '-' | '+'
;
property
: IDENT S*
;
ruleset
: selector [ ',' S* selector ]*
  '{' S* declaration [ ';' S* declaration ]* '}' S*
```

```

;
selector
: simple_selector [ combinator simple_selector ]*
;
simple_selector
: element_name? [ HASH | class | attrib | pseudo ]* S*
;
class
: '.' IDENT
;
element_name
: IDENT | '*'
;
attrib
: '[' S* IDENT S* [ [ '=' | INCLUDES | DASHMATCH ] S*
[ IDENT | STRING ] S* ]? ']'
;
pseudo
: ':' [ IDENT | FUNCTION S* IDENT S* ')' ]
;
declaration
: property ':' S* expr prio?
| /* vacío */
;
prio
: IMPORTANT_SYM S*
;
expr
: term [ operator term ]*
;
term
: unary_operator?
[ NUMBER S* | PERCENTAGE S* | LENGTH S* | EMS S* | EXS S* | ANGLE S* |
TIME S* | FREQ S* | function ]
| STRING S* | IDENT S* | URI S* | RGB S* | UNICODERANGE S* | hexcolor
;
function
: FUNCTION S* expr ')' S*
;
/*
* Hay una restricción en el color que debe tener
* 3 o 6 dígitos hexadecimales (es decir, [0-9a-fA-F])
* después de "#"; ej., "#000" está bien, pero "#abcd" no.
*/
hexcolor
: HASH S*
;

```

## D.2 Scanner de léxico

La siguiente es la definición de los comandos (tokenizer) escrito en notación Flex (ver [FLEX](#)). Es sensible a la diferencia entre mayúsculas y minúsculas.

Las dos apariciones de "\377" representan el número de carácter más alto que la versión actual de Flex puede manejar (decimal 255). Debe leerse como "\4177777" (decimal 1114111), que es el punto de código más alto posible en Unicode/ISO-10646.

```

%option case-insensitive

h      [0-9a-f]
nonascii [\200-\377]
unicode  \\{h}{1,6}[ \t\r\n\f]?
escape  {unicode}||\\[ ~\200-\377]
nmstart  [a-z]||{nonascii}||{escape}
nmchar   [a-zA-Z0-9]||{nonascii}||{escape}
string1  \"([ \t !#$%&(-~]||\\{nl}|\'|{nonascii}||{escape}))*\"
string2  \'([ \t !#$%&(-~]||\\{nl}|\\"|{nonascii}||{escape}))*\'

ident    {nmstart}{nmchar}*
name     {nmchar}+
num       [0-9]+|[0-9]*\".\"[0-9]+
string    {string1}||{string2}
url       ([!#$%&*~]||{nonascii}||{escape})*

```

```
w      [ \t\r\n\f]*
nl      \n|\r\n|\r|\f
range  \{1,6\}|{h}(\{0,5\}|{h}(\{0,4\}|{h}(\{0,3\}|{h}(\{0,2\}|{h}(\{??\{h\}))))))

%%

[ \t\r\n\f]+      {return S;}

\/\*[^\*]*\*+([^\/][^\*]*\*+)*\/ /* ignore comments */

"<!--"      {return CDO;}
"-->"      {return CDC;}
"~="      {return INCLUDES;}
"|="      {return DASHMATCH;}

{string}      {return STRING;}

{ident}      {return IDENT;}

"#"{name}      {return HASH;}

"@import"      {return IMPORT_SYM;}
"@page"      {return PAGE_SYM;}
"@media"      {return MEDIA_SYM;}
"@font-face"      {return FONT_FACE_SYM;}
"@charset"      {return CHARSET_SYM;}
@"{ident}"      {return ATKEYWORD;}

"!{w}important"      {return IMPORTANT_SYM;}

{num}em      {return EMS;}
{num}ex      {return EXS;}
{num}px      {return LENGTH;}
{num}cm      {return LENGTH;}
{num}mm      {return LENGTH;}
{num}in      {return LENGTH;}
{num}pt      {return LENGTH;}
{num}pc      {return LENGTH;}
{num}deg      {return ANGLE;}
{num}rad      {return ANGLE;}
{num}grad      {return ANGLE;}
{num}ms      {return TIME;}
{num}s      {return TIME;}
{num}Hz      {return FREQ;}
{num}kHz      {return FREQ;}
{num}{ident}      {return DIMEN;}
{num}%      {return PERCENTAGE;}
{num}      {return NUMBER;}

"url("{w}{string}{w}")"      {return URI;}
"url("{w}{url}{w}")"      {return URI;}
{ident}"("      {return FUNCTION;}

U\+{range}      {return UNICODERANGE;}
U\+{h}{1,6}-{h}{1,6}      {return UNICODERANGE;}

.      {return *yytext;}
```

### D.3 Comparación de los comandos en CSS2 y CSS1

Hay algunas diferencias en la sintaxis especificada en la recomendación CSS1 ([\[CSS1\]](#)) y la de más arriba. La mayoría de éstas se deben a los nuevos comandos en CSS2 que no existían en CSS1. Otras son debido a que la gramática ha si reescrita para que sea más legible. Sin embargo, hay algunos cambios incompatibles, en lo que se consideró eran errores en la sintaxis de CSS1. Están explicados abajo.

- Las hojas de estilo CSS1 sólo podían estar en una codificación de 1 byte por carácter, tal como ASCII e ISO-8859-1. CSS2 no tiene tal limitación. En la práctica, había algunas pequeñas dificultades en extrapolar los comandos de CSS1, y algunas AU han aceptado la codificación de 2 bytes.
- CSS1 sólo permitía cuatro dígitos hexadecimales después de la barra invertida (\) para referirse a los caracteres de Unicode, CSS2 [permite seis](#). Además, CSS2 permite una carácter de espacio en blanco para delimitar la secuencia de escape. Ej., de acuerdo a CSS1, la cadena "\abcedf" tiene 3 letras (\abcd, e, y f), de acuerdo a CSS2 tiene solamente una (\abcedf).
- El carácter de tabulador (ASCII 9) no estaba permitido en las cadenas. Sin embargo, como las cadenas en CSS1 sólo eran usadas para los nombres de las fuentes y para los URL, la única forma de que esto

pueda conducir a la incompatibilidad entre CSS1 y CSS2 es que una hoja de estilo contenga una familia de fuentes con un tabulador en su nombre.

- De modo similar, los saltos de línea ([escapadas con una barra invertida](#)) no estaban permitidas en las cadenas en CSS1.
- CSS2 analiza un número seguido inmediatamente por un identificador como un comando DIMEN (es decir, una unidad desconocida), CSS1 lo analizaba como un número y un identificador. Eso significa que en CSS1, la declaración 'font: 10pt/1.2serif' era correcta, como lo era 'font: 10pt/12pt serif'; en CSS2, se requiere un espacio antes de "serif". (Algunas AU han aceptado el primer ejemplo, pero no el segundo.)
- En CSS1, un nombre de clase podía comenzar con un dígito (".55ft"), a menos que fuera una dimensión (".55in"). En CSS2, tales clases son analizadas como dimensiones desconocidas (para permitir futuros agregados de nuevas unidades). Para convertir ".55ft" en una clase válida, CSS2 requiere que el primer dígito sea escapado (".\35 5ft" ).



# Apéndice E. Referencias

## E.1 Referencias normativas

### [COLORIMETRY]

"Colorimetry, Second Edition", Publicación CIE 15.2-1986, ISBN 3-900-734-00-3.  
Disponible en <http://www.hike.te.chiba-u.ac.jp/ikeda/CIE/publ/abst/15-2-86.html>.

### [CSS1]

"Cascading Style Sheets, level 1", H. W. Lie y B. Bos, Diciembre 17 de 1996.  
Disponible en <http://www.w3.org/TR/REC-CSS1-961217.html>.

### [FLEX]

"Flex: The Lexical Scanner Generator", Version 2.3.7, ISBN 1882114213.

### [HTML40]

"HTML 4.0 Specification", D. Raggett, A. Le Hors, I. Jacobs, Julio 8 de 1997.  
Disponible en <http://www.w3.org/TR/REC-html40/>. La recomendación define tres definiciones del tipo de documento: Strict, Transitional y Frameset, todas al alcance en la recomendación.

### [IANA]

"Assigned Numbers", STD 2, RFC 1700, USC/ISI, J. Reynolds y J. Postel, Octubre 1994.  
Disponible en <ftp://ftp.internic.net/rfc/rfc1700.txt>.

### [ICC32]

"ICC Profile Format Specification, version 3.2", 1995.  
Disponible en <ftp://sgigate.sgi.com/pub/icc/ICC32.pdf>.

### [ISO8879]

ISO 8879:1986 "Information Processing -- Text and Office Systems -- Standard Generalized Markup Language (SGML)", ISO 8879:1986.  
Por la lista de las entidades SGML, consultar <ftp://ftp.ifi.uio.no/pub/SGML/ENTITIES/>.

### [ISO10646]

"Information Technology - Universal Multiple- Octet Coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane", ISO/IEC 10646-1:1993. La especificación actual también toma en consideración las primeras cinco enmiendas a ISO/IEC 10646-1:1993. Los documentos de utilidad [roadmap of the BMP](#) y [roadmap of plane 1](#) muestran qué escrituras se asientan en qué rangos numéricos.

### [PNG10]

"PNG (Portable Network Graphics) Specification, Version 1.0 specification", T. Boutell ed., Octubre 1 de 1996.  
Disponible en <http://www.w3.org/pub/WWW/TR/REC-png-multi.html>.

### [RFC1808]

"Relative Uniform Resource Locators", R. Fielding, Junio 1995.  
Disponible en <ftp://ds.internic.net/rfc/rfc1808.txt>.

### [RFC2045]

"Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", N. Freed y N. Borenstein, Noviembre 1996.  
Disponible en <ftp://ftp.internic.net/rfc/rfc2045.txt>. Observe que esta RFC deja obsoletas a RFC1521, RFC1522 y RFC1590.

### [RFC2068]

"HTTP Version 1.1 ", R. Fielding, J. Gettys, J. Mogul, H. Frystyk Nielsen y T. Berners-Lee, Enero 1997.  
Disponible en <ftp://ftp.internic.net/rfc/rfc2068.txt>.

### [RFC2070]

"Internationalization of the HyperText Markup Language", F. Yergeau, G. Nicol, G. Adams, y M. Dürst, Enero 1997.  
Disponible en <ftp://ds.internic.net/rfc/rfc2070.txt>.

### [RFC2119]

"Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, Marzo 1997.  
Disponible en <ftp://ds.internic.net/rfc/rfc2119.txt>.

### [RFC2318]

"The text/css Media Type", H. Lie, B. Bos, C. Lilley, Marzo 1998.  
Disponible en <ftp://ds.internic.net/rfc/rfc2318.txt>.

### [RFC1738]

"Uniform Resource Locators", T. Berners-Lee, L. Masinter y M. McCahill, Diciembre 1994.  
Disponible en <ftp://ds.internic.net/rfc/rfc1738.txt>.

### [SRGB]

"Proposal for a Standard Color Space for the Internet - sRGB", M. Anderson, R. Motta, S. Chandrasekar, M. Stokes.  
Disponible en <http://www.w3.org/Graphics/Color/sRGB.html>.

### [UNICODE]

"The Unicode Standard: Version 2.0", The Unicode Consortium, Addison-Wesley Developers Press, 1996. Por bidireccionalidad, ver también las correcciones en <http://www.unicode.org/unicode/uni2errata/bidi.htm>. Para mayor información, consulte la home page del Unicode Consortium en <http://www.unicode.org/>. La última versión de Unicode. Para mayor información, consulte la home page del Unicode Consortium en <http://www.unicode.org/>.

### [URI]

"Uniform Resource Identifiers (URI): Generic Syntax and Semantics", T. Berners-Lee, R. Fielding, L. Masinter, Noviembre 18 de 1997.  
Disponible en <http://www.ics.uci.edu/pub/ietf/uri/draft-fielding-uri-syntax-01.txt>. Este es un trabajo en etapa de desarrollo que se espera actualice [RFC1738] y [RFC1808].

### [XML10]

"Extensible Markup Language (XML) 1.0" T. Bray, J. Paoli, C.M. Sperberg-McQueen, editores, Febrero 10 de 1998.  
Disponible en <http://www.w3.org/TR/REC-xml/>.

### [YACC]

"YACC - Yet another compiler compiler", S. C. Johnson, Technical Report, Murray Hill, 1975.

## E.2 Referencias informativas

### [CHARSETS]

Valores registrados de juegos de caracteres. Descargue una lista de valores registrados de juegos de caracteres de <ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets>.

### [DOM]

"Document Object Model Specification", L. Wood, A. Le Hors, Octubre 9 de 1997.  
Disponible en <http://www.w3.org/TR/WD-DOM/>

### [ISO10179]

ISO/IEC 10179:1996 "Information technology -- Processing languages -- Document Style Semantics and Specification Language (DSSSL)"  
Disponible en <http://occam.sjf.novell.com:8080/dsssl/dsssl96>.

### [GAMMA]

"Gamma correction on the Macintosh Platform", C. A. Poynton.  
Disponible en [ftp://ftp.inforamp.net/pub/users/poynton/doc/Mac/Mac\\_gamma.pdf](ftp://ftp.inforamp.net/pub/users/poynton/doc/Mac/Mac_gamma.pdf).

### [HTML32]

"HTML 3.2 Reference Specification", Dave Raggett, Enero 14 de 1997.  
Disponible en <http://www.w3.org/TR/REC-html32.html>.

### [INFINIFONT]

Ver <http://www.fonts.com/hp/infinifont/moredet.html>.

### [ISO9899]

ISO/IEC 9899:1990 Lenguajes de programación -- C.

### [MONOTYPE]

Ver [http://www.monotype.com/html/oem/uni\\_scrmod.html](http://www.monotype.com/html/oem/uni_scrmod.html).

### [NEGOT]

"Transparent Content Negotiation in HTTP", K. Holtman, A. Mutz, Marzo 9 de 1997.  
Disponible en <http://gewis.win.tue.nl/~koen/conneg/draft-ietf-http-negotiation-01.html>.

### [OPENTYPE]

Ver <http://www.microsoft.com/OpenType/OTSpec/tablist.htm>.

### [PANOSE]

Por información acerca de las métricas de clasificación de PANOSE, consulte <http://www.fonts.com/hp/panose/greybook> y los siguientes capítulos: [Latin Text](#), [Latin Script](#), [Latin Decorative](#) y [Latin Pictorial](#).

Los números Panose para algunas fuentes están disponibles en línea y pueden ser [solicitados](#).

### [PANOSE2]

Ver <http://www.w3.org/Fonts/Panose/pan2.html> Panose-2 is not limited to Latin typefaces.

### [POSTSCRIPT]

"The PostScript Language Reference Manual", Segunda Edición, Adobe Systems, Inc., Addison-Wesley Publishing Co., Diciembre 1990.

### [RFC1630]

"Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web", T. Berners-Lee, Junio 1994.  
Disponible en <ftp://ftp.internic.net/rfc/rfc1630.txt>.

### [RFC1766]

"Tags for the Identification of Languages", H. Alvestrand, Marzo 1995.  
Disponible en <ftp://ftp.internic.net/rfc/rfc1766.txt>.

### [RFC1866]

"HyperText Markup Language 2.0", T. Berners-Lee y D. Connolly, Noviembre 1995.  
Disponible en <ftp://ds.internic.net/rfc/rfc1866.txt>.

### [RFC1942]

"HTML Tables", Dave Raggett, Mayo 1996.  
Disponible en <ftp://ds.internic.net/rfc/rfc1942.txt>.

### [TRUETYPEGX]

Ver <http://fonts.apple.com/TTRefMan/index.html> por detalles acerca de TrueType GX de Apple Computer, incluyendo [descriptions of the added tables](#) y [font quality specifications](#).

### [W3CSTYLE]

Recurso del W3C sobre hojas de estilo en la web.  
Examine en <http://www.w3.org/pub/WWW/Style>.

### [WAI-PAGEAUTH]

"WAI Accesibility Guidelines: Page Authoring" para el diseño de documentos accesibles están disponibles en: <http://www.w3.org/TR/WD-WAI-PAGEAUTH>.

## Apéndice F. Índice de propiedades

| Nombre   | Valores   | Valor inicial                    | Se aplica a<br>(Por defecto: todos)        | Se hereda? | Porcentajes<br>(Por defecto: N/A)     | Grupos de medios |
|--|---|----------------------------------|--|------------|---------------------------------------|------------------|
| 'azimuth'  | <ángulo>   [[ left-side   far-left   left   center-left   center   center-right   right   far-right   right-side ]    behind ]   leftwards   rightwards   inherit | center                           |  | si         |                                       | auditivos        |
| 'background'   | ['background-color'    'background-image'    'background-repeat'    'background-attachment'    'background-position']   inherit                                   | XX                               |  | no         | permitidos en 'background-position'   | visuales         |
| 'background-attachment'  | scroll   fixed   inherit  | scroll                           |  | no         |                                       | visuales         |
| 'background-color'   | <color>   transparent   inherit   | transparent                      |  | no         |                                       | visuales         |
| 'background-image'   | <uri>   none   inherit  | none                             |  | no         |                                       | visuales         |
| 'background-position'  | [ [<porcentaje>   <medida> ]{1,2}   [ [top   center   bottom]    [left   center   right] ] ]   inherit  | 0% 0%                            | elementos reemplazados y a nivel de bloque | no         | referidos al tamaño de la propia caja | visuales         |
| 'background-repeat'  | repeat   repeat-x   repeat-y   no-repeat   inherit  | repeat                           |  | no         |                                       | visuales         |
| 'border'   | [ 'border-width'    'border-style'    <color> ]   inherit   | ver propiedades individuales     |  | no         |                                       | visuales         |
| 'border-collapse'  | collapse   separate   inherit   | collapse                         | elementos 'table' e 'inline-table'         | si         |                                       | visuales         |
| 'border-color'   | <color>{1,4}   transparent   inherit  | ver propiedades individuales     |  | no         |                                       | visuales         |
| 'border-spacing'   | <medida> <medida>?   inherit  | 0                                | elementos 'table' e 'inline-table'         | si         |                                       | visuales         |
| 'border-style'   | <border-style>{1,4}   inherit   | ver propiedades individuales     |  | no         |                                       | visuales         |
| 'border-top'<br>'border-right'<br>'border-bottom'<br>'border-left'                         | [ 'border-top-width'    'border-style'    <color> ]   inherit   | ver propiedades individuales     |  | no         |                                       | visuales         |
| 'border-top-color'<br>'border-right-color'<br>'border-bottom-color'<br>'border-left-color' | <color>   inherit   | el valor de la propiedad 'color' |  | no         |                                       | visuales         |
| 'border-top-style'<br>'border-right-style'<br>'border-bottom-style'<br>'border-left-style' | <border-style>   inherit  | none                             |  | no         |                                       | visuales         |
| 'border-top-width'<br>'border-right-width'<br>'border-bottom-width'<br>'border-left-width' | <border-width>   inherit  | medium                           |  | no         |                                       | visuales         |
| 'border-width'   | <border-width>{1,4}   inherit   | ver propiedades individuales     |  | no         |                                       | visuales         |

| Nombre              | Valores   | Valor inicial                        | Se aplica a<br>(Por defecto: todos)                            | Se hereda?             | Porcentajes<br>(Por defecto: N/A)              | Grupos de medios       |
|---------------------|---|--------------------------------------|--|------------------------|--|------------------------|
| 'bottom'            | <medida>   <porcentaje>   auto   inherit  | auto                                 | elementos posicionados   | no                     | referidos a la altura del bloque de contención | visuales               |
| 'caption-side'      | top   bottom   left   right   inherit   | top                                  | elementos 'table-caption'                                      | si                     |  | visuales               |
| 'clear'             | none   left   right   both   inherit  | none                                 | elementos a nivel de bloque                                    | no                     |  | visuales               |
| 'clip'              | <forma>   auto   inherit  | auto                                 | elementos reemplazados y a nivel de bloque                     | no                     |  | visuales               |
| 'color'             | <color>   inherit   | depende de la aplicación del usuario |  | si                     |  | visuales               |
| 'content'           | [ <cadena>   <uri>   <contador>   attr(X)   open-quote   close-quote   no-open-quote   no-close-quote ]+   inherit  | cadena vacía                         | pseudo-elementos :before y :after                              | no                     |  | todos                  |
| 'counter-increment' | [ <identificador> <entero>? ]+   none   inherit   | none                                 |  | no                     |  | todos                  |
| 'counter-reset'     | [ <identificador> <entero>? ]+   none   inherit   | none                                 |  | no                     |  | todos                  |
| 'cue'               | [ 'cue-before'    'cue-after' ]   inherit   | XX                                   |  | no                     |  | auditivos              |
| 'cue-after'         | <uri>   none   inherit  | none                                 |  | no                     |  | auditivos              |
| 'cue-before'        | <uri>   none   inherit  | none                                 |  | no                     |  | auditivos              |
| 'cursor'            | [ [ <uri> ,]* [ auto   crosshair   default   pointer   move   e-resize   ne-resize   nw-resize   n-resize   se-resize   sw-resize   s-resize   w-resize   text   wait   help ] ]   inherit  | auto                                 |  | si                     |  | visuales, interactivos |
| 'direction'         | ltr   rtl   inherit   | ltr                                  | todos los elementos, pero ver el texto                         | si                     |  | visuales               |
| 'display'           | inline   block   list-item   run-in   compact   marker   table   inline-table   table-row-group   table-header-group   table-footer-group   table-row   table-column-group   table-column   table-cell   table-caption   none   inherit | inline                               |  | no                     |  | todos                  |
| 'elevation'         | <ángulo>   below   level   above   higher   lower   inherit   | level                                |  | si                     |  | auditivos              |
| 'empty-cells'       | show   hide   inherit   | show                                 | elementos 'table-cell'   | si                     |  | visuales               |
| 'float'             | left   right   none   inherit   | none                                 | todos menos los elementos posicionados y el contenido generado | no                     |  | visuales               |
| 'font'              | [ [ 'font-style'    'font-variant'    'font-weight' ]? 'font-size' [ / 'line-height' ]? 'font-family' ]   caption   icon   menu   message-box   small-caption   status-bar   inherit  | ver propiedades individuales         |  | si                     | permitidos en 'font-size' y 'line-height'      | visuales               |
| 'font-family'       | [[ <nombre-de-la-familia>   <familia-genérica> ],]* [ <nombre-de-la-familia>   <familia-genérica> ]   inherit   | depende de la aplicación del usuario |  | si                     |  | visuales               |
| 'font-size'         | <tamaño-absoluto>   <tamaño-relativo>   | medium                               |  | si, el valor computado | referidos al tamaño de la                      | visuales               |

| Nombre   | Valores   | Valor inicial | Se aplica a<br>(Por defecto: todos)   | Se hereda? | Porcentajes<br>(Por defecto: N/A)                 | Grupos de medios    |
|--|---|---------------|---|------------|---|---------------------|
|  | <medida>   <porcentaje>   inherit   |               |   | se hereda  | fuelle del elemento padre                         |                     |
| 'font-size-adjust'   | <número>   none   inherit   | none          |   | si         |   | visuales            |
| 'font-stretch'   | normal   wider   narrower   ultra-condensed   extra-condensed   condensed   semi-condensed   semi-expanded   expanded   extra-expanded   ultra-expanded   inherit   | normal        |   | si         |   | visuales            |
| 'font-style'   | normal   italic   oblique   inherit   | normal        |   | si         |   | visuales            |
| 'font-variant'   | normal   small-caps   inherit   | normal        |   | si         |   | visuales            |
| 'font-weight'  | normal   bold   bolder   lighter   100   200   300   400   500   600   700   800   900   inherit  | normal        |   | si         |   | visuales            |
| 'height'   | <medida>   <porcentaje>   auto   inherit  | auto          | todos los elementos menos los elementos de línea no reemplazados, columnas de tabla y grupos de columna | no         | ver texto   | visuales            |
| 'left'   | <medida>   <porcentaje>   auto   inherit  | auto          | elementos posicionados  | no         | referidos al ancho del bloque de contención       | visuales            |
| 'letter-spacing'   | normal   <medida>   inherit   | normal        |   | si         |   | visuales            |
| 'line-height'  | normal   <número>   <medida>   <porcentaje>   inherit   | normal        |   | si         | referidos al tamaño de fuente del propio elemento | visuales            |
| 'list-style'   | [ 'list-style-type'    'list-style-position'    'list-style-image' ]   inherit  | XX            | elementos con 'display: list-item'  | si         |   | visuales            |
| 'list-style-image'   | <uri>   none   inherit  | none          | elementos con 'display: list-item'  | si         |   | visuales            |
| 'list-style-position'  | inside   outside   inherit  | outside       | elementos con 'display: list-item'  | si         |   | visuales            |
| 'list-style-type'  | disc   circle   square   decimal   decimal-leading-zero   lower-roman   upper-roman   lower-greek   lower-alpha   lower-latin   upper-alpha   upper-latin   hebrew   armenian   georgian   cjk-ideographic   hiragana   katakana   hiragana-iroha   katakana-iroha   none   inherit | disc          | elementos con 'display: list-item'  | si         |   | visuales            |
| 'margin'   | <margin-width>{1,4}   inherit   | XX            |   | no         | referidos al ancho del bloque de contención       | visuales            |
| 'margin-top'<br>'margin-right'<br>'margin-bottom'<br>'margin-left' | <margin-width>   inherit  | 0             |   | no         | referidos al ancho del bloque de contención       | visuales            |
| 'marker-offset'  | <medida>   auto   inherit   | auto          | elementos con 'display: marker'   | no         |   | visuales            |
| 'marks'  | [ crop    cross ]   none   inherit  | none          | contexto de página  | N/A        |   | visuales, paginados |
| 'max-height'   | <medida>   <porcentaie>   | none          | todos los   | no         | referidos a la                                    | visuales            |

| Nombre   | Valores   | Valor inicial                        | Se aplica a<br>(Por defecto: todos)                                       | Se hereda? | Porcentajes<br>(Por defecto: N/A)                   | Grupos de medios       |
|--|---|--------------------------------------|---|------------|---|------------------------|
|  | none   inherit  |                                      | elementos menos los elementos de línea no reemplazados y tablas           |            | altura del bloque de contención                     |                        |
| 'max-width'  | <medida>   <porcentaje>   none   inherit                            | none                                 | todos los elementos menos los elementos de línea no reemplazados y tablas | no         | referidos al ancho del bloque de contención         | visuales               |
| 'min-height'   | <medida>   <porcentaje>   inherit                                   | 0                                    | todos los elementos menos los elementos de línea no reemplazados y tablas | no         | referidos a la altura del bloque de contención      | visuales               |
| 'min-width'  | <medida>   <porcentaje>   inherit                                   | UA dependent                         | todos los elementos menos los elementos de línea no reemplazados y tablas | no         | referidos al ancho del bloque de contención         | visuales               |
| 'orphans'  | <entero>   inherit  | 2                                    | elementos a nivel de bloque   | si         |   | visuales, paginados    |
| 'outline'  | [ 'outline-color'    'outline-style'    'outline-width' ]   inherit | ver propiedades individuales         |   | no         |   | visuales, interactivos |
| 'outline-color'  | <color>   invert   inherit  | invert                               |   | no         |   | visuales, interactivos |
| 'outline-style'  | <border-style>   inherit  | none                                 |   | no         |   | visuales, interactivos |
| 'outline-width'  | <border-width>   inherit  | medium                               |   | no         |   | visuales, interactivos |
| 'overflow'   | visible   hidden   scroll   auto   inherit                          | visible                              | elementos reemplazados y a nivel de bloque                                | no         |   | visuales               |
| 'padding'  | <padding-width>{1,4}   inherit                                      | XX                                   |   | no         | referidos al ancho del bloque de contención         | visuales               |
| 'padding-top'<br>'padding-right'<br>'padding-bottom'<br>'padding-left' | <padding-width>   inherit   | 0                                    |   | no         | referidos al ancho del bloque de contención         | visuales               |
| 'page'   | <identificador>   auto  | auto                                 | elementos a nivel de bloque   | si         |   | visuales, paginados    |
| 'page-break-after'   | auto   always   avoid   left   right   inherit                      | auto                                 | elementos a nivel de bloque   | no         |   | visuales, paginados    |
| 'page-break-before'  | auto   always   avoid   left   right   inherit                      | auto                                 | elementos a nivel de bloque   | no         |   | visuales, paginados    |
| 'page-break-inside'  | avoid   auto   inherit  | auto                                 | elementos a nivel de bloque   | si         |   | visuales, paginados    |
| 'pause'  | [ [<tiempo>   <porcentaje>]{1,2} ]   inherit                        | depende de la aplicación del usuario |   | no         | ver descripciones de 'pause-before' y 'pause-after' | auditivos              |
| 'pause-after'  | <tiempo>   <porcentaje>   inherit                                   | depende de la aplicación del usuario |   | no         | ver texto   | auditivos              |
| 'pause-before'   | <tiempo>   <porcentaje>   inherit                                   | depende de la aplicación del usuario |   | no         | ver texto   | auditivos              |
| 'pitch'  | <frecuencia>   x-low   low   medium   high   x-high   inherit       | medium                               |   | si         |   | auditivos              |
| 'pitch-range'  | <número>   inherit  | 50                                   |   | si         |   | auditivos              |



| Nombre              | Valores   | Valor inicial  | Se aplica a<br>(Por defecto: todos)                         | Se hereda?     | Porcentajes<br>(Por defecto: N/A)              | Grupos de medios                    |
|---------------------|---|--|---|----------------|--|-------------------------------------|
| 'play-during'       | <uri> mix? repeat?   auto   none   inherit  | auto   |   | no             |  | <a href="#">auditivos</a>           |
| 'position'          | static   relative   absolute   fixed   inherit  | static   | todos los elementos, pero no al contenido generado          | no             |  | <a href="#">visuales</a>            |
| 'quotes'            | [<cadena> <cadena>]+   none   inherit   | depende de la aplicación del usuario                             |   | si             |  | <a href="#">visuales</a>            |
| 'richness'          | <número>   inherit  | 50   |   | si             |  | <a href="#">auditivos</a>           |
| 'right'             | <medida>   <porcentaje>   auto   inherit  | auto   | elementos posicionados                                      | no             | referidos al ancho del bloque de contención    | <a href="#">visuales</a>            |
| 'size'              | <medida>{1,2}   auto   portrait   landscape   inherit   | auto   | contexto de página  | N/A            |  | <a href="#">visuales, paginados</a> |
| 'speak'             | normal   none   spell-out   inherit   | normal   |   | si             |  | <a href="#">auditivos</a>           |
| 'speak-header'      | once   always   inherit   | once   | elementos que tienen la información de títulos en una tabla | si             |  | <a href="#">auditivos</a>           |
| 'speak-numeral'     | digits   continuous   inherit   | continuous   |   | si             |  | <a href="#">auditivos</a>           |
| 'speak-punctuation' | code   none   inherit   | none   |   | si             |  | <a href="#">auditivos</a>           |
| 'speech-rate'       | <número>   x-slow   slow   medium   fast   x-fast   faster   slower   inherit                               | medium   |   | si             |  | <a href="#">auditivos</a>           |
| 'stress'            | <número>   inherit  | 50   |   | si             |  | <a href="#">auditivos</a>           |
| 'table-layout'      | auto   fixed   inherit  | auto   | elementos 'table' e 'inline-table'                          | no             |  | <a href="#">visuales</a>            |
| 'text-align'        | left   right   center   justify   <cadena>   inherit  | depende de la aplicación del usuario y la dirección de escritura | elementos a nivel de bloque                                 | si             |  | <a href="#">visuales</a>            |
| 'text-decoration'   | none   [ underline    overline    line-through    blink ]   inherit   | none   |   | no (ver texto) |  | <a href="#">visuales</a>            |
| 'text-indent'       | <medida>   <porcentaje>   inherit   | 0  | elementos a nivel de bloque                                 | si             | referidos al ancho del bloque de contención    | <a href="#">visuales</a>            |
| 'text-shadow'       | none   [<color>    <medida> <medida> <medida>? ,]* [<color>    <medida> <medida> <medida>? ]   inherit      | none   |   | no (ver texto) |  | <a href="#">visuales</a>            |
| 'text-transform'    | capitalize   uppercase   lowercase   none   inherit   | none   |   | si             |  | <a href="#">visuales</a>            |
| 'top'               | <medida>   <porcentaje>   auto   inherit  | auto   | elementos posicionados                                      | no             | referidos a la altura del bloque de contención | <a href="#">visuales</a>            |
| 'unicode-bidi'      | normal   embed   bidi-override   inherit  | normal   | todos los elementos, pero ver el texto                      | no             |  | <a href="#">visuales</a>            |
| 'vertical-align'    | baseline   sub   super   top   text-top   middle   bottom   text-bottom   <porcentaje>   <medida>   inherit | baseline   | elementos a nivel de línea y 'table-cell'                   | no             | referidos a 'line-height' del propio elemento  | <a href="#">visuales</a>            |
| 'visibility'        | visible   hidden   collapse   inherit   | inherit  |   | no             |  | <a href="#">visuales</a>            |
| 'voice-family'      | [[<voz-específica>   <voz-genérica> ],]* [<voz-específica>   <voz-  | depende de la aplicación del usuario                             |   | si             |  | <a href="#">auditivos</a>           |



| Nombre         | Valores   | Valor inicial | Se aplica a<br>(Por defecto: todos)  | Se hereda? | Porcentajes<br>(Por defecto: N/A)           | Grupos de medios                    |
|----------------|---|---------------|--|------------|---|-------------------------------------|
|                | genérica> ]   inherit   |               |  |            |   |                                     |
| 'volume'       | <número>   <porcentaje>   silent   x-soft   soft   medium   loud   x-loud   inherit | medium        |  | si         | referidos al valor heredado                 | <a href="#">auditivos</a>           |
| 'white-space'  | normal   pre   nowrap   inherit   | normal        | elementos a nivel de bloque  | si         |   | <a href="#">visuales</a>            |
| 'widows'       | <entero>   inherit  | 2             | elementos a nivel de bloque  | si         |   | <a href="#">visuales, paginados</a> |
| 'width'        | <medida>   <porcentaje>   auto   inherit  | auto          | todos los elementos menos elementos de línea no reemplazados, filas de tabla y grupos de filas | no         | referidos al ancho del bloque de contención | <a href="#">visuales</a>            |
| 'word-spacing' | normal   <medida>   inherit   | normal        |  | si         |   | <a href="#">visuales</a>            |
| 'z-index'      | auto   <entero>   inherit   | auto          | elementos posicionados   | no         |   | <a href="#">visuales</a>            |

# Apéndice G. Índice de descriptores

| Nombre           | Valores   | Valor inicial                        |
|------------------|---|--------------------------------------|
| 'ascent'         | <número>  | indefinido                           |
| 'baseline'       | <número>  | 0                                    |
| 'bbox'           | <número>, <número>, <número>, <número>  | indefinido                           |
| 'cap-height'     | <número>  | indefinido                           |
| 'centerline'     | <número>  | indefinido                           |
| 'definition-src' | <uri>   | indefinido                           |
| 'descent'        | <número>  | indefinido                           |
| 'font-family'    | [ <nombre-de-la-familia>   <familia-genérica> ] [, [<nombre-de-la-familia>   <familia-genérica> ]]*   | depende de la aplicación del usuario |
| 'font-size'      | all   <medida> [, <medida>]*  | all                                  |
| 'font-stretch'   | all   [ normal   ultra-condensed   extra-condensed   condensed   semi-condensed   semi-expanded   expanded   extra-expanded   ultra-expanded ] [, [ normal   ultra-condensed   extra-condensed   condensed   semi-condensed   semi-expanded   expanded   extra-expanded   ultra-expanded ] ]* | normal                               |
| 'font-style'     | all   [ normal   italic   oblique ] [, [normal   italic   oblique] ]*   | all                                  |
| 'font-variant'   | [normal   small-caps] [, [normal   small-caps]]*  | normal                               |
| 'font-weight'    | all   [normal   bold   100   200   300   400   500   600   700   800   900] [, [normal   bold   100   200   300   400   500   600   700   800   900]]*  | all                                  |
| 'mathline'       | <número>  | indefinido                           |
| 'panose-1'       | [<entero>]{10}  | 0 0 0 0 0 0 0 0 0 0                  |
| 'slope'          | <número>  | 0                                    |
| 'src'            | [ <uri> [formato(<cadena> [, <cadena>]*)]   <nombre-de-la-fuente> ] [, <uri> [formato(<cadena> [, <cadena>]*)]   <nombre-de-la-fuente> ] ]*   | indefinido                           |
| 'stemh'          | <número>  | indefinido                           |
| 'stemv'          | <número>  | indefinido                           |
| 'topline'        | <número>  | indefinido                           |
| 'unicode-range'  | <rango-u> [, <rango-u>]*  | U+0-7FFFFFFF                         |
| 'units-per-em'   | <número>  | indefinido                           |
| 'widths'         | [<rango-u> ]? [<número> ]+ [, [<rango-u> ]? <número> ]+ ]   | indefinido                           |
| 'x-height'       | <número>  | indefinido                           |