

# CSS Display Module Level 3

Este módulo describe cómo se genera el árbol de cuadros de formato CSS a partir del árbol de elementos de documento y define la propiedad 'display' que lo controla.

CSS es un lenguaje para describir la representación de documentos estructurados (como HTML y XML) en pantalla, en papel, etc.

Este documento fue elaborado por un grupo que opera en el marco de la Política de Patentes del W3C. W3C mantiene una lista pública de las divulgaciones de patentes realizadas en relación con los resultados del grupo; esa página también incluye instrucciones para divulgar una patente. Una persona que tenga conocimiento real de una patente que la persona crea que contiene reclamaciones esenciales debe divulgar la información de conformidad con la sección 6 de la Política de Patentes de la W3C.

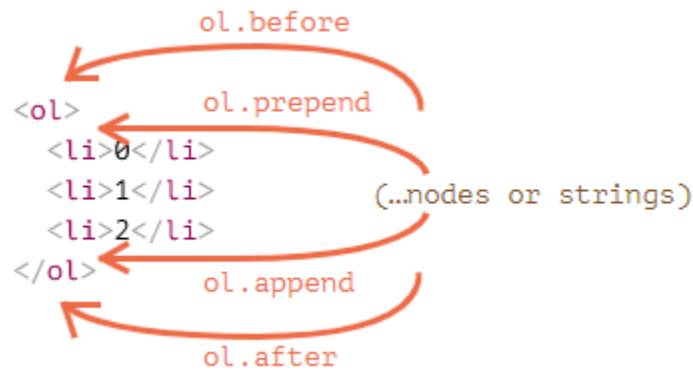
Se dispone de un informe preliminar de aplicación. Se añadirán más pruebas durante el período CR.

Las siguientes características están en riesgo y pueden descartarse durante el período CR:

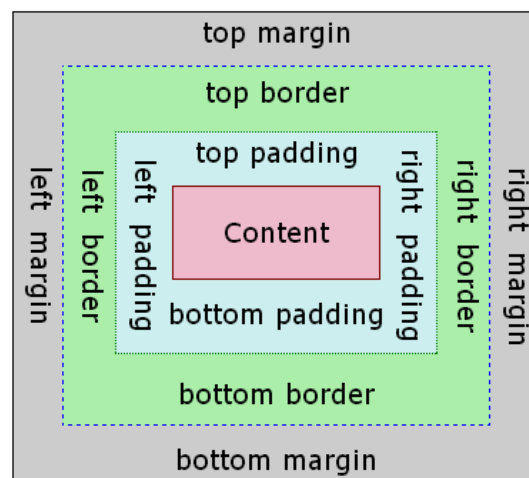
- Aplicación de '::first-letter' en presencia de run-ins
- 'pantalla: run-in'
- Todos los valores multi-palabra clave de 'display'

"En riesgo" es un término de arte del Proceso W3C, y no implica necesariamente que la característica esté en peligro de ser retirada o retrasada. Esto significa que el GT cree que la característica puede tener dificultades para ser implementada interoperable mente en tiempo y forma, y marcarla como tal permite al WG abandonar la característica si es necesario al realizar la transición a la etapa de Rec propuesta, sin tener que publicar primero un nuevo Rec candidato sin la característica.

CSS toma un documento de origen, organizado como un árbol de elementos y nodos de texto, y lo representa en un canvas (como la pantalla, un pedazo de papel o una secuencia de audio). (Algunos documentos de origen comienzan a partir de árboles más complejos, como el DOM, que pueden tener nodos de comentario y otros tipos de cosas. Para los propósitos de CSS, todos estos tipos adicionales de nodos se ignoran, como si no existieran.)



Para ello, genera una estructura intermedia, el árbol de cuadros, que representa la estructura de formato del documento representado. Cada cuadro del árbol de cuadros representa su elemento correspondiente (o pseudoelemento) en el espacio y/o el tiempo en el lienzo, mientras que cada ejecución de texto en el árbol de cuadros también representa el contenido de sus nodos de texto correspondientes.



Para crear el árbol de cuadros, CSS primero usa en cascada y herencia, para asignar un valor calculado para cada propiedad CSS a cada elemento y nodo de texto en el árbol de origen. (Véase [CSS3-CASCADE].)

A continuación, para cada elemento, CSS genera cero o más cuadros según lo especificado por la propiedad 'display' de ese elemento. Normalmente, un elemento genera un único cuadro, el cuadro principal, que se representa a sí mismo y contiene su contenido en el árbol de cuadros. Sin embargo, algunos valores de 'display' (por ejemplo, 'display: list-item') generan más de una caja (por ejemplo, un cuadro de bloque principal y un cuadro de marcador secundario). Y algunos valores (como 'ninguno' o 'contenido') hacen que el elemento y/o sus descendientes no generen ningún cuadro en absoluto. Los cuadros a menudo se refieren por su tipo de "visualización", por ejemplo, un cuadro generado por un elemento con 'display: block' se denomina "cuadro de bloque" o simplemente un "bloque".

```

1 <div class="container display-none-example">
2   <div class="circle color1"></div>
3   <div class="circle color2"></div>
4   <div class="circle color3"></div>
5 </div>

```

### CSS Border Styles and Colors

A un cuadro se le asignan los mismos estilos que a su elemento generador, a menos que se indique lo contrario. En general, las propiedades heredadas se asignan al cuadro principal y, a continuación, heredan a través del árbol de cuadros a cualquier otro cuadro generado por el mismo elemento. Las propiedades no heredadas predeterminadas para aplicar al cuadro principal, pero cuando el elemento genera varios cuadros, a veces se definen para aplicarse a un cuadro diferente: por ejemplo, las propiedades 'border' aplicadas a un elemento de tabla se aplican a su cuadro de cuadrícula de tabla, no a su cuadro contenedor de tabla principal. Si el proceso de cálculo del valor altera los estilos de esos cuadros y se solicita el estilo del elemento (por ejemplo, a través de `getComputedStyle()`), el elemento refleja, para cada propiedad, el valor del cuadro al que se aplicó esa propiedad.

De forma similar, cada secuencia contigua de nodos de texto hermano genera una ejecución de texto que contiene su contenido de texto, al que se le asignan los mismos estilos que los nodos de texto generadores. Sin embargo, si la secuencia no contiene texto, no genera una ejecución de texto.

Al construir el árbol de cuadros, los cuadros generados por un elemento son descendientes del cuadro principal de cualquier elemento antecesor. En el caso general, el cuadro primario directo del cuadro principal de un elemento es el cuadro principal de su elemento antecesor más cercano que genera un cuadro; sin embargo, hay algunas excepciones, como para los cuadros "run-in", los tipos de visualización (como tablas) que generan varios cuadros de contenedor y la intervención de cuadros anónimos.

Descargo de Responsabilidad: No tomo crédito de nada escrito en este documento esta información fue extraída de w3 y resumida para el beneficio de otros.

Un cuadro anónimo es un cuadro que no está asociado a ningún elemento. Los cuadros anónimos se generan en determinadas circunstancias para arreglar el árbol de cuadros cuando se requiere una estructura anidada determinada que no es proporcionada por los cuadros generados a partir del árbol de elementos. Por ejemplo, un cuadro de celda de tabla requiere un tipo determinado de cuadro primario (el cuadro de fila de tabla) y generará un cuadro de fila de tabla anónimo alrededor de sí mismo si su elemento primario no es un cuadro de fila de tabla. (Véase [CSS2] § 17.2.1.) A diferencia de los cuadros generados por elementos, cuyos estilos heredan estrictamente a través del árbol de elementos, los cuadros anónimos (que solo existen en el árbol de cuadros) heredan a través de su parentesco de árbol de cuadros.

## Interacciones del módulo

Este módulo reemplaza y amplía la definición de la propiedad 'display' definida en la sección [CSS2] 9.2.4.

Ninguna de las propiedades de este módulo se aplica a los pseudoelementos ::first-line o ::first-letter.

## Definiciones de valor

Esta especificación sigue las convenciones de definición de propiedades CSS de [CSS2] utilizando el valor

definición de [CSS-VALUES-3]. Los tipos de valor no definidos en esta especificación se definen en CSS Values & Units [CSS-VALUES-3]. La combinación con otros módulos CSS puede expandir las definiciones de estos tipos de valor.

Además de los valores específicos de la propiedad enumerados en sus definiciones, todas las propiedades definidas en esta especificación también aceptan las palabras clave de todo el CSS como su valor de propiedad. Para la legibilidad no se han repetido explícitamente.

## Modos de diseño de cuadro: la propiedad 'display'

<i>Name:</i>	<b><i>'display'</i></b>
<i>Value:</i>	[ <a href="#">&lt;display-outside&gt;</a>    <a href="#">&lt;display-inside&gt;</a> ]   <a href="#">&lt;display-list-item&gt;</a>   <a href="#">&lt;display-internal&gt;</a>   <a href="#">&lt;display-box&gt;</a>   <a href="#">&lt;display-legacy&gt;</a>
<i>Initial:</i>	inline
<i>Applies to:</i>	<a href="#">all elements</a>
<i>Inherited:</i>	no
<i>Percentages:</i>	n/a
<i>Computed</i>	see prose in a variety of specs

in fue

Se espera que los agentes de usuario admitan esta propiedad en todos los medios, incluidos los no visuales.

La propiedad 'display' define el tipo de visualización de un elemento, que consta de las dos cualidades básicas de cómo genera un elemento cuadros:

el tipo de visualización interno, que define (si es un elemento no reemplazado) el tipo de formato contexto que genera, dictando cómo se establecen sus cajas descendientes. (La visualización interna de un elemento reemplazado está fuera del ámbito de CSS.)

el tipo de visualización externa, que dicta cómo el propio cuadro principal participa en el diseño de flujo.

Las ejecuciones de texto no tienen ningún tipo de visualización.

Algunos valores de "visualización" tienen efectos secundarios adicionales: como 'list-item', que también genera un

Pseudoelemento '::marker' y 'none', que hace que todo el subárbol del elemento quede fuera del árbol de cajas.

La propiedad 'display' no tiene ningún efecto en la semántica de un elemento:

lenguaje de documentos y no se ven afectados por CSS. Aparte del valor "none", que también afecta a la salida auditiva/de voz [CSS-SPEECH-1] y a la interactividad de un elemento y sus descendientes, la propiedad 'display' sólo afecta al diseño visual: su propósito es permitir a los diseñadores libertad para cambiar el comportamiento de diseño de un elemento sin afectar a la semántica del documento subyacente.

Los valores se definen de la siguiente manera:

**<display-outside>= bloque | | en línea**

Descargo de Responsabilidad: No tomo crédito de nada escrito en este documento esta información fue extraída de w3 y resumida para el beneficio de otros.

**<display-inside>= | tabla de | de raíz de flujo | rejilla de | flexible | rubí**

**<display-listitem>= <display-outside>? && [ flujo | raíz de flujo ]? && list-item**

```

<display-internal> = table-row-group | table-header-group |
                    table-footer-group | table-row | table-
                    cell | table-column-group | table-column
                    | table-caption | ruby-base | ruby-text |
                    ruby-base-container |
                    ruby-text-container

<display-box>      = contents | none

<display-Legacy>   = inline-block | inline-table | inline-flex | inline-
                    grid

```

## **Roles de visualización externas para el diseño de flujo: las palabras clave 'block', 'inline' y 'run-in'**

Las <display-outside>keywords especifican el tipo de visualización externa del elemento, que es esencialmente su

función del cuadro principal en el diseño de flujo. Se definen de la siguiente manera:

### **'bloque'**

El elemento genera un cuadro que es de nivel de bloque cuando se coloca en el diseño de flujo. [CSS2]

### **'en línea'**

El elemento genera un cuadro que está en el nivel en línea cuando se coloca en el diseño de flujo. [CSS2]

### **'run-in'**

El elemento genera un cuadro de ejecución, que es un tipo de cuadro de nivel en línea con un comportamiento especial que intenta combinarlo en un contenedor de bloques posterior. Consulte § 3 Diseño de ejecución para obtener más información.

Si se especifica un valor <display-outside> pero se omite <display-inside>, la visualización interna del elemento tipo predeterminados a 'flow'.

Modelos de diseño de visualización interna: las palabras clave 'flow', 'flow-root', 'table', 'flex', 'grid' y 'ruby'

Las <display-inside>keywords especifican el tipo de visualización interno del elemento, que define el tipo de contexto de formato que establece su contenido (suponiendo que sea un elemento no reemplazado). Se definen de la siguiente manera:

### **'flujo'**

El elemento establece su contenido mediante el diseño de flujo (diseño de bloque y línea).

Si su tipo de visualización externa es 'inline' o 'run-in', y participa en un contexto de formato de bloque o en línea, entonces genera un cuadro en línea.

De lo contrario, genera un cuadro de contenedor de bloques.

Dependiendo del valor de otras propiedades (como 'position', 'float' o 'overflow') y si en sí mismo está participando en un contexto de formato de bloque o en línea, establece un nuevo bloque

contexto de formato para su contenido o integra su contenido en su contexto de formato primario. Ver

CSS2.1 Capítulo 9. [CSS2] Se considera que un contenedor de bloques que establece un nuevo contexto de formato de bloque tiene un tipo de visualización interno usado de 'raíz de flujo'.

### **'raíz de flujo'**

El elemento genera un cuadro de contenedor de bloques y establece su contenido mediante el diseño de flujo. Siempre establece un nuevo contexto de formato de bloque para su contenido. [CSS2]

### **'mesa'**

El elemento genera un cuadro contenedor de tabla principal que establece un contexto de formato de bloque y que contiene un cuadro de cuadrícula de tabla generado adicionalmente que establece un contexto de formato de tabla. [CSS2]

### **'flex'**

El elemento genera un cuadro de contenedor flex principal y establece un contexto de formato flexible.

[CSS3-FLEXBOX]

**'cuadrícula'**

El elemento genera un cuadro contenedor de cuadrícula principal y establece un contexto de formato de cuadrícula.

[CSS3-GRID-LAYOUT]

**'rubí'**

El elemento genera un cuadro contenedor rubí y establece un contexto de formato rubí además de integrar su contenido de nivel base en su contexto de formato primario (si está en línea) o generar un cuadro contenedor del tipo de visualización externa adecuado (si no lo es).

[CSS3RUBY]

Si se especifica un valor `<display-inside>` pero se omite `<display-outside>`, la visualización externa del elemento

tipo de valor predeterminado para 'bloquear', excepto 'ruby', que por defecto es 'inline'.

