

CSS media queries

Una **media query** consiste en un tipo de medio y al menos una consulta que limita las hojas de estilo utilizando características del medio como ancho, alto y color. Se entiende como un módulo CSS3 que permite adaptar la representación del contenido a características del dispositivo. Añadido en [CSS3](#), las media queries dejan que la presentación del contenido se adapte a un rango específico de dispositivos de salida sin tener que cambiar el contenido en sí.

Sintaxis

Las Media queries consisten de un **media type** y una o mas expresiones, implicando características del medio, la cual se resuelve como verdadera o falsa. El resultado de la consulta es verdadera si el tipo de medio especificado en el media query concuerda con el tipo de dispositivo que está siendo mostrado y todas las expresiones en el media query son verdaderas.

```
<!-- CSS media query on a link element -->
<link rel="stylesheet" media="(max-width: 800px)" href="example.css" />

<!-- CSS media query within a style sheet -->
<style>
@media (max-width: 600px) {
  .facet_sidebar {
    display: none;
  }
}
</style>
```

Cuando una media query es verdadera, la hoja de estilo correspondiente o reglas de estilos son aplicadas, siguiendo las reglas normales de cascada.

Cuando `not` ni `only` son usadas dentro de una query, el tipo de medio es opcional y será interpretada como `all`.

Operadores Lógicos

Se pueden redactar queries utilizando operadores lógicos, incluyendo `not`, `and`, y `only`.

Además se puede combinar múltiples queries en una lista separada por comas múltiples; si cualquiera de las queries en la lista es verdadera, la hoja de estilo asociada es aplicada. Esto es equivalente a una operación lógica "or".

and

El operador `and` es usado para colocar juntas múltiples funciones multimedia. Un query básico con el tipo de medio especificado como `all` puede lucir así:

```
1 | @media (min-width: 700px) { ... }
```

Si usted quiere aplicar ese query solo si la pantalla esta en formato horizontal, usted puede utilizar el operador `and` y colocar la siguiente cadena:

```
1 | @media (min-width: 700px) and (orientation: landscape) { ... }
```

La siguiente query solo retornara verdadero si la ventana tiene un ancho de 700px o mas y la pantalla esta en formato horizontal. Ahora si usted quiere aplicar esta opción solo si tipo de medio es TV, usted puede utilizar la siguiente cadena:

```
1 | @media tv and (min-width: 700px) and (orientation: landscape) { ... }
```

Esta query solo se aplica si el tipo de medio es TV, la ventana tiene 700px de ancho o mas y la pantalla esta en formato horizontal.

lista separada por comas

Las listas separadas por comas se comportan como el operador `or` cuando es usado en media queries. Cuando utilice una lista separada por comas y alguno de los queries retorna verdadero, el estilo o la hoja de estilos sera aplicada. Cada query en una lista separada por comas es tratado como una query individual y cualquier operador aplicado a un medio no afectara a los demás. Esto significa que cada query en una lista separada por comas puede tener como objetivo diferentes medios, tipos y estados.

Si usted quiere aplicar una serie de estilos para un equipo con un ancho mínimo de 700px o si el dispositivo esta colocado en horizontal, usted puede escribir lo siguiente:

```
1 | @media (min-width: 700px), handheld and (orientation: landscape) { ... }
```

Por lo tanto, si esta en una screen con una ventana de 800px de ancho, la declaración del medio retornara verdadero debido a la primera parte de la lista, si aplicamos esto a un dispositivo `@media all and (min-width: 700px)` podría retornar verdadero a pesar del hecho de que la pantalla falle la verificación tipo de medio del `handheld` en la segunda query. Por otra parte si estuviese en un `handheld` con un ancho de ventana de 500px, la primera parte de la lista fallaría pero la segunda parte retornara verdadero debido a la declaración de medio.

not

El operador `not` aplica a todo el query y retorna verdadero si es posible y sino retorna falso (como por ejemplo `monochrome` en un monitor a color o una ventana con un ancho mínimo de `min-width: 700px` en un monitor de 600px). Un `not` negara un query si es posible pero no a todos los query posibles si están ubicados en una lista separada por comas. El operador `not` no puede ser usado para negar un query individual, solo para un query completo. Por ejemplo, el `not` en el siguiente query es evaluado al final:

```
1 | @media not all and (monochrome) { ... }
```

Esto significa que el query es evaluado de la siguiente forma:

```
1 | @media not (all and (monochrome)) { ... }
```

... y no de esta forma:

```
1 | @media (not all) and (monochrome) { ... }
```

Otro Ejemplo:

```
1 | @media not screen and (color), print and (color)
```

Es evaluado de la siguiente forma:

```
1 | @media (not (screen and (color))), print and (color)
```

only

El operador `only` previene que navegadores antiguos que no soportan queries con funciones apliquen los estilos asignados:

```
1 | <link rel="stylesheet" media="only screen and (color)" href="Ejemplo.css">
```

Pseudo-BNF

```
media_query_list: <media_query> [, <media_query> ]*
media_query: [[only | not]? <media_type> [ and <expression> ]*
             | <expression> [ and <expression> ]*
expression: ( <media_feature> [: <value>]? )
media_type: all | aural | braille | handheld | print |
            projection | screen | tty | tv | embossed
media_feature: width | min-width | max-width
              | height | min-height | max-height
              | device-width | min-device-width | max-device-width
              | device-height | min-device-height | max-device-height
              | aspect-ratio | min-aspect-ratio | max-aspect-ratio
              | device-aspect-ratio | min-device-aspect-ratio | max-device-aspect-ratio
              | color | min-color | max-color
              | color-index | min-color-index | max-color-index
              | monochrome | min-monochrome | max-monochrome
              | resolution | min-resolution | max-resolution
              | scan | grid
```

Los queries son insensibles a las mayúsculas o minúsculas. Media queries que contengan tipos de medios desconocidos siempre retornaran falso.

 **Nota:** Los paréntesis son requeridos alrededor de las expresiones, no utilizarlos es un error.

Funciones Multimedia

La mayoría de las funciones multimedia pueden ser precedidas por "min-" o "max-" para expresar "greater or equal to" o "less than or equal to". Esto elimina la necesidad de usar los símbolos "<" y ">" los cuales podrían causar conflictos con HTML y XML. Si usted usa una función multimedia sin especificarle un valor, la expresión retornara verdadero si el valor es diferente de cero.

❏ **Nota:** Si una función multimedia no aplica al dispositivo donde el navegador esta corriendo, la expresión que contiene esa función siempre retornara falso. Por Ejemplo, cambiar la relación de aspecto en un dispositivo aural siempre resultara falso.

color

Valor: <color>

Medio: **media/visual**

Acepta prefijos min/max: si

Indica el numero de bits por componente de color del dispositivo de salida. Si el dispositivo no soporta colores, este valor es 0.

❏ **Nota:** Si el componente de color tiene diferente numero de bits por color, entonces el valor mas pequeño es utilizado. Por Ejemplo, si una pantalla utiliza 5 bits para azul y rojo y 6 bits para el verde, entonces se utilizan 5 bits por componente de color. Si el dispositivo utiliza índices de color se usara el mínimo valor de bits por color en la tabla.

Ejemplos

Para aplicar una hoja de estilo a todos los dispositivos que soporten colores:

```
1 | @media all and (color) { ... }
```

Para aplicar una hoja de estilo a dispositivos con al menos 4 bits por componente de color:

```
1 | @media all and (min-color: 4) { ... }
```

color-index

Valor: `<integer>`

Medio: `media/visual`

Acepta prefijos min/max: sí

Indica el numero de entradas en la tabla de colores para el dispositivo de salida.

Ejemplos

Para indicar que una hoja de estilo debe aplicarse a todos los dispositivos utilizando índices de color, usted puede hacer esto:

```
1 | @media all and (color-index) { ... }
```

Para aplicar una hoja de estilo a un dispositivo con un índice de al menos 256 colores:

```
1 | <link rel="stylesheet" media="all and (min-color-index: 256)" href="http://
```

aspect-ratio

Valor: `<ratio>`

Medio: `media/visual`, `media/tactile`

Acepta prefijos min/max: sí

Describe el aspecto de una zona a mostrar en el dispositivo de salida. Este valor consiste en enteros positivos separados por una barra ("/"). Esto representa la razón de aspecto de los pixeles horizontales (primer termino) a los pixeles verticales (segundo termino).

Ejemplo

Lo siguiente selecciona una hoja de estilo especial para ser aplicada donde la proporción del tamaño de la pantalla del dispositivo de salida es al menos la misma de ancho que de alto.

```
1 | @media screen and (min-aspect-ratio: 1/1) { ... }
```

Este selecciona el estilo cuando la proporción de aspecto sea 1:1 o superior. En otras palabras este estilo solo sera aplicado cuando el área de visualización sea cuadrada u horizontal.

device-aspect-ratio

Valor: **<ratio>**

Medio: **media/visual**, **media/tactile**

Acepta prefijos min/max: sí

Describe la proporción de aspecto del dispositivo de salida. Este valor consiste de dos enteros positivos separados por una barra ("/"). Este representa la proporción de aspecto de los pixeles horizontales (primer termino) a los pixeles verticales (segundo termino).

Ejemplo

El siguiente código selecciona una hoja de estilo especial para ser aplicada en pantallas panorámicas ("widescreen").

```
1 | @media screen and (device-aspect-ratio: 16/9), screen and (device-aspect-r
```

Este selecciona el estilo cuando la proporción de aspecto sea 16:9 o 16:10.

device-height

Valor: **<length>**

Medio: **media/visual**, **media/tactile**

Acepta prefijos min/max: sí

Describe la altura del dispositivo de salida (ya sea la totalidad de la pantalla o página y no el área del documento a renderizar).

device-width

Valor: **<length>**

Medio: **media/visual**, **media/tactile**

Acepta prefijos min/max: sí

Describe la anchura del dispositivo de salida (ya sea la totalidad de la pantalla o página y no el área del documento a renderizar).

Ejemplo

Para aplicar una hoja de estilo a un documento cuando este sea mostrado en una pantalla de menos de 800px de ancho, usted puede usar esto:

```
1 | <link rel="stylesheet" media="screen and (max-device-width: 799px)" />
```

grid

Valor: <integer>

Medio: all


Acepta prefijos min/max: no

Determina cuando el dispositivo de salida es un dispositivo de cuadrícula o de mapa de bits. Si el dispositivo esta basado en una cuadrícula (como una terminal TTY o una pantalla de teléfono de solo texto), el valor sera 1, de lo contrario sera 0.

Ejemplo

Para aplicar una hoja de estilo a un dispositivo portátil con una pantalla de 15 caracteres o mas estrecha:

```
1 | @media handheld and (grid) and (max-width: 15em) { ... }
```

 **Nota:** La unidad "em" tiene un significado especial para los dispositivos de cuadrícula; ya que la anchura exacta de una "em" no puede ser determinada, se asume un "em" como el ancho de una celda en la cuadrícula y el alto de una celda en la cuadrícula.

height

Valor: <length>

Medio: **media/visual**, **media/tactile**

Acepta prefijos min/max: sí

La función `height` describe la altura de la superficie a renderizar en el dispositivo de salida (como la altura de una ventana o la bandeja de papel en una impresora).

📄 **Nota:** Cuando el usuario cambia el tamaño de una ventana Firefox también cambia las hojas de estilo para utilizar la mas adecuada basándose en los valores de width y height del query.

monochrome

Valor: <integer>

Medio: **media/visual**

Acepta prefijos min/max: sí

Indica el número de bits por pixel en un dispositivo monocromático (escala de grises). Si el dispositivo no es monocromático el valor sera 0.

Ejemplos

Para aplicar una hoja de estilo a todos los dispositivos monocromáticos:

```
1 | @media all and (monochrome) { ... }
```

Para aplicar una hoja de estilo a un dispositivo monocromático con al menos 8 bits por pixel:

```
1 | @media all and (min-monochrome: 8) { ... }
```

orientation

Valor: landscape | portrait

Medio: **media/visual**

Acepta prefijos min/max: no

Indica cuando el dispositivo esta en modo landscape (el ancho de la pantalla es mayor al alto) o modo portrait (el alto de la pantalla es mayor al ancho).

Ejemplo

Para aplicar una hoja de estilo solo en orientación vertical (portrait):

```
1 | @media all and (orientation: portrait) { ... }
```

resolution

Valor: `<resolution>`

Medio: `bitmap`

Acepta prefijos min/max: sí

Indica la resolución (densidad de píxeles) del dispositivo de salida. La resolución puede ser especificada en puntos por pulgada (dpi) o en puntos por centímetros (dpcm).

Ejemplo

Para aplicar una hoja de estilo a dispositivos con al menos 300 dpi de resolución:

```
1 | @media print and (min-resolution: 300dpi) { ... }
```

Para reemplazar la vieja sintaxis (`min-device-pixel-ratio: 2`):

```
1 | @media screen and (min-resolution: 2dppx) { ... }
```

scan

Valor: `progressive` | `interlace`

Medio: `media/tv`

Acepta prefijos min/max: no

Describe el proceso de exploración de televisión de los dispositivos de salida.

Ejemplo

Para aplicar una hoja de estilo solo a televisores de exploración progresiva:

```
1 | @media tv and (scan: progressive) { ... }
```

width

Valor: `<length>`

Medio: `media/visual`, `media/tactile`

Acepta prefijos min/max: sí

La función `width` describe el ancho de la superficie a renderizar en el dispositivo de salida (como el ancho de una ventana de un documento o el ancho de la bandeja de papel en una impresora).

Nota: Cuando el usuario cambia el tamaño de una ventana Firefox también cambia las hojas de estilo para utilizar la mas adecuada basándose en los valores de `width` y `height` del query.

Ejemplos

Si usted quiere especificar una hoja de estilo para dispositivos portátiles o pantallas con un ancho de al menos 20em, usted puede usar esta query:

```
1 | @media handheld and (min-width: 20em), screen and (min-width: 20em) { ...
```

Esta query especifica una hoja de estilo para ser aplicada a un medio impreso con un ancho mayor a 8.5 pulgadas:

```
1 | <link rel="stylesheet" media="print and (min-width: 8.5in)"
2 |     href="http://foo.com/mystyle.css" />
```

Esta query especifica una hoja de estilo para ser utilizada cuando la ventana tiene un ancho entre 500 y 800 pixeles:

```
1 | @media screen and (min-width: 500px) and (max-width: 800px) { ... }
```

Media Types

Un media type puede ser declarado en la *cabecera* de un documento HTML usando el atributo "media" dentro de un elemento `<link>`. El valor específico del atributo "media" indica en que dispositivo el documento vinculado será mostrado. Media types también pueden ser declarados dentro de instrucciones de procesamiento XML, el `@import` at-rule, y el `@media` at-rule. CSS2 definen los siguientes media types:

- braille
 - embossed
 - handheld
 - print
 - projection
 - screen
 - speech
 - tty
 - tv
-

Específico de Mozilla

Mozilla ofrece varias funciones específicas de Gecko. Algunas de estas pueden ser propuestas como funciones oficiales.

Requires Gecko 1.9.2(Firefox 3.6 / Thunderbird 3.1 / Fennec 1.0)

`-moz-images-in-menus`

Valor: `<integer>`

Medio: `media/visual`

Acepta prefijos min/max: no

Si el dispositivo acepta que haya imágenes en menús, el valor es 1; de otro modo será 0. Esto corresponde a la pseudoclase: `:-moz-system-metric(images-in-menus)`.

Requires Gecko 1.9.2(Firefox 3.6 / Thunderbird 3.1 / Fennec 1.0)

`-moz-mac-graphite-theme`

Valor: `<integer>`

Medio: `media/visual`

Acepta prefijos min/max: no

Si el usuario ha configurado su dispositivo para usar la apariencia "Grafito" en Mac OS X, esto será 1. Si el usuario está usando la apariencia azul por defecto, o si no usa Mac OS X, esto será 0.

Esto corresponde a la pseudoclase: `:-moz-system-metric(mac-graphite-theme)`.

`-moz-maemo-classic`

Valor: `<integer>`

Medio: `media/visual`

Acepta prefijos min/max: no

Si el usuario esta usando Maemo con el tema original, esto sera 1; Si esta usando el nuevo tema Fremantle, sera 0.

Esto corresponde a la pseudoclase: `:-moz-system-metric(maemo-classic)`

`-moz-device-pixel-ratio`

Valor: `<number>`

Medio: `media/visual`

Acepta prefijos min/max: sí

Da el numero de pixeles del dispositivo por pixeles de CSS.

No use esta función.

En cambio use la función `resolution` con la unidad `dppx`.

`-moz-device-pixel-ratio` puede ser usada para compatibilidad con Firefox 16 o anterior; y `-webkit-device-pixel-ratio` con navegadores basados en WebKit que no soporten `dppx`.

Ejemplo:

```
1  @media (-webkit-min-device-pixel-ratio: 2), /* Navegadores basados
2      (min--moz-device-pixel-ratio: 2),      /* Navegadores anterio
3      (min-resolution: 2dppx),                /* La forma estandar *
4      (min-resolution: 192dpi)                /* compatibilidad con
```

Vea este [artículo CSSWG](#) sobre buenas practicas para compatibilidad en relación a `resolution` y `dppx`.

Nota: Esta funcion multimedia tambien esta implementada en Webkit como `-webkit-device-pixel-ratio`. Los prefijos minimos y maximos de esta funcion implementados por Gecko se llaman asi: `min--moz-device-pixel-ratio` y `max--moz-device-pixel-ratio`; y los mismos prefijos implementados por Webkit se llaman asi: `-webkit-min-device-pixel-ratio` y `-webkit-max-device-pixel-ratio`.

Requires Gecko 25.0(Firefox 25.0 / Thunderbird 25.0 / SeaMonkey 2.22)

`-moz-os-version`

Valor: `windows-xp` | `windows-vista` | `windows-win7` | `windows-win8`

Medio: `media/visual`

Acepta prefijos min/max: no

Indica que sistema operativo esta en uso actualmente. Actualmente solo es implementado en Windows. Sus posibles valores son:

- `windows-xp`
- `windows-vista`
- `windows-win7`
- `windows-win8`
- `windows-win10`

Esto se hace para permitir a los skins y a algunos códigos funcionen mejor con el sistema operativo en el que se ejecutan.

Requires Gecko 1.9.2(Firefox 3.6 / Thunderbird 3.1 / Fennec 1.0)

`-moz-scrollbar-end-backward`

Valor: `<integer>`

Medio: `media/visual`

Acepta prefijos min/max: no

El valor sera 1 si la interfaz de usuario muestra una fecha inversa al final de la barra de desplazamiento, de lo contrario el valor sera 0.

Esto corresponde a la pseudoclase: `:-moz-system-metric(scrollbar-end-backward)`.

Requires Gecko 1.9.2(Firefox 3.6 / Thunderbird 3.1 / Fennec 1.0)

`-moz-scrollbar-end-forward`

Valor: `<integer>`

Medio: `media/visual`

Acepta prefijos min/max: no

El valor sera 1 si la interfaz de usuario muestra una fecha apuntando hacia la derecha al final de la barra de desplazamiento. de lo contrario el valor sera 0.

Esto corresponde a la pseudoclase: `:-moz-system-metric(scrollbar-end-forward)`.

Requires Gecko 1.9.2(Firefox 3.6 / Thunderbird 3.1 / Fennec 1.0)

`-moz-scrollbar-start-backward`

Valor: `<integer>`

Medio: `media/visual`

Acepta prefijos min/max: no

El valor sera 1 si la interfaz de usuario muestra una fecha inversa al principio de la barra de desplazamiento, de lo contrario el valor sera 0.

Esto corresponde a la pseudoclase: `:-moz-system-metric(scrollbar-start-backward)`.

Requires Gecko 1.9.2(Firefox 3.6 / Thunderbird 3.1 / Fennec 1.0)

`-moz-scrollbar-start-forward`

Valor: `<integer>`

Medio: `media/visual`

Acepta prefijos min/max: no

El valor sera 1 si la interfaz de usuario muestra una fecha apuntando hacia la derecha al principio de la barra de desplazamiento, de lo contrario el valor sera 0.

Esto corresponde a la pseudoclase: `:-moz-system-metric(scrollbar-start-forward)`.

`-moz-scrollbar-thumb-proportional`

Requires Gecko 1.9.2(Firefox 3.6 / Thunderbird 3.1 / Fennec 1.0)

Valor: `<integer>`

Medio: `media/visual`

Acepta prefijos min/max: no

El valor sera 1 si la interfaz de usuario muestra la miniatura de la barra de desplazamiento de forma proporcional (basado en el porcentaje del documento que es

visible), de lo contrario el valor sera 0.

Esto corresponde a la pseudoclase: `:-moz-system-metric(scrollbar-thumb-proportional)`.

Requires Gecko 1.9.2(Firefox 3.6 / Thunderbird 3.1 / Fennec 1.0)

`-moz-touch-enabled`

Valor: `<integer>`

Medio: `media/visual`

Acepta prefijos min/max: no

El valor sera 1 si el dispositivo soporta eventos táctiles (una pantalla táctil), de lo contrario el valor sera 0.

Esto corresponde a la pseudoclase: `:-moz-system-metric(touch-enabled)`.

Ejemplo

Usted puede usar esto para renderizar sus botones un poco mas grandes, Por Ejemplo, si el usuario se encuentra en una pantalla táctil, esto hará los botones mas fáciles de usar con los dedos.

Requires Gecko 1.9.2(Firefox 3.6 / Thunderbird 3.1 / Fennec 1.0)

`-moz-windows-classic`

Valor: `<integer>`

Medio: `media/visual`

Acepta prefijos min/max: no

El valor sera 1 si el usuario utiliza un windows sin temas visuales (modo clasico); de lo contrario el valor sera 0.

Esto corresponde a la pseudoclase: `:-moz-system-metric(windows-classic)`.

Requires Gecko 1.9.2(Firefox 3.6 / Thunderbird 3.1 / Fennec 1.0)

`-moz-windows-compositor`

Valor: `<integer>`

Medio: `media/visual`

Acepta prefijos min/max: no

El valor sera 1 si el usuario utiliza windows con el compositor de ventanas DWM; de lo contrario el valor sera 0.

Esto corresponde a la pseudoclase: `:-moz-system-metric(windows-compositor)`.

Requires Gecko 1.9.2(Firefox 3.6 / Thunderbird 3.1 / Fennec 1.0)

`-moz-windows-default-theme`

Valor: `<integer>`

Medio: `media/visual`

Acepta prefijos min/max: no

El valor sera 1 si el usuario actualmente esta utilizando algunos de los temas por defecto de Windows (Luna, Royale, Zune, or Aero), de lo contrario el valor sera 0.

Esto corresponde a la pseudoclase: `:-moz-system-metric(windows-default-theme)`.

Requires Gecko 2.0(Firefox 4 / Thunderbird 3.3 / SeaMonkey 2.1)

`-moz-windows-theme`

Valor: `aero | luna-blue | luna-olive | luna-silver | royale | generic | zune`

Medio: `media/visual`

Acepta prefijos min/max: no

Indica cual tema de Windows esta en uso actualmente. Solo disponible para Windows. Sus posibles valores son:

- `aero`
- `luna-blue`
- `luna-olive`
- `luna-silver`
- `royale`
- `generic`
- `zune`

Esto se hace para permitir a los skins y a algunos códigos funcionen mejor con el tema utilizado por el sistema operativo en el que se ejecutan.

Específico de WebKit

`-webkit-transform-3d`

Vakir: `<integer>`

Medio: `media/visual`

Acepta prefijos min/max: no

- [Especificación](#)
- [Documentación](#)

Si está soportado, el valor es 1, si no, el valor es 0.

Ejemplo

```
@media (-webkit-transform-3d) {  
  .foo {  
    transform-style: preserve-3d;  
  }  
}  
  
@media (-webkit-transform-3d: 1) {  
  .foo {  
    transform-style: preserve-3d;  
  }  
}
```

`-webkit-transform-2d`

Valor: `<integer>`

Medio: `media/visual`

Acepta prefijos min/max: no

- [Documentación](#)

Si está soportado, el valor es 1, si no, el valor es 0.

`-webkit-transition`

Value: `<integer>`

Medio: `media/visual`

Acepta prefijos min/max: no

- Documentación

Si está soportado, el valor es 1, si no, el valor es 0.

Compatibilidad con Navegadores

 **We're converting our compatibility data into a machine-readable JSON format.** This compatibility table still uses the old format, because we haven't yet converted the data it contains. **Find out how you can help!**

Escritorio

Móvil

Feature	Chrome	Firefox (Gecko)	Internet Explorer	Opera	Safari
Soporte básico	21	3.5 (1.9.1)	9.0	9	4
grid	?	Sin soporte ^[1]	?	?	?
resolution	29	3.5 (1.9.1) ^[2] 8.0 (8.0) ^[3]	?	?	?
scan	?	Sin soporte ^[4]	?	?	?
-webkit-min-device-pixel-ratio, -webkit-max-device-pixel-ratio	(Yes)	Sin soporte ^[7]	Sin soporte	(Yes)	?
-webkit-transform-3d	(Yes) ^[5]	49 (49) ^[6]	?	(Yes) ^[5]	1.0 ^[5]
-webkit-transform-2d	Sin soporte	Sin soporte	?	(Yes) ^[5]	1.0 ^[5]
-webkit-transition	Sin soporte	Sin soporte	?	(Yes) ^[5]	1.0 ^[5]
display-mode	?	47 (47)	?	?	?

[1] grid media type no está soportado

[2] Soporta valores `<integer>`.

[3] Soporta valores `<number>`, según la especificación.

[4] tv media no está soportado

[5] Ver [WebKit bug 22494](#).

[6] Implementado por razones de compatibilidad web en Gecko 46.0 (Firefox 46.0 / Thunderbird 46.0 / SeaMonkey 2.43) tras cambiar `layout.css.prefixes.webkit` por defecto a `false` (ver [error 1239799](#)). Desde Gecko 49.0 (Firefox 49 / Thunderbird 49 / SeaMonkey 2.46) `layout.css.prefixes.webkit` por defecto es `true`.

[7] Implementado como alias para `min--moz-device-pixel-ratio` y `max--moz-device-pixel-ratio` por razones de compatibilidad web in Gecko 45.5 (Firefox 45.0 / Thunderbird 45.0 / SeaMonkey 2.42) (ver [error 1176968](#)) tras cambiar `layout.css.prefixes.webkit` y `layout.css.prefixes.device-pixel-ratio-webkit` por defecto a `false`. Desde Gecko 49.0 (Firefox 49 / Thunderbird 49 / SeaMonkey 2.46) `layout.css.prefixes.webkit` por defecto es `true`.

Vea también

- [Especificación CSS 3 media query](#)
 - Tipos de Medios
 - Usando media queries desde código
-