

[Home](#) > [All articles](#)

Conceptos básicos del diseño web responsivo

Cómo crear sitios que respondan a las necesidades y capacidades del dispositivo en el que se ven.

Feb 12, 2019 — Actualizado May 14, 2020

Available in: [English](#), [Português](#), [Русский](#), [中文](#), [日本語](#), and [한국어](#)



Pete LePage

[Twitter](#) [GitHub](#) [Glitch](#) [Homepage](#)



Rachel Andrew

[Twitter](#) [GitHub](#) [Glitch](#) [Homepage](#)

En esta página



[Establecer la ventana gráfica](#)

[Ajustar el tamaño del contenido a la ventana gráfica](#)

[Utilice consultas de medios CSS para mejorar la capacidad de respuesta](#)

[Cómo elegir los breakpoints](#)

[Ver consultas de medios con breakpoints en Chrome DevTools](#)

By Chrome DevRel dispositivos móviles para navegar por la web continúa creciendo a un ritmo astronómico, y estos dispositivos a menudo están limitados por el tamaño de la pantalla y requieren un enfoque diferente sobre cómo se presenta el contenido en la pantalla.

El diseño web responsivo, originalmente definido por [Ethan Marcotte en A List Apart](#), responde a las necesidades de los usuarios y a la de los dispositivos que utilizan. El diseño cambia según el tamaño y las capacidades del dispositivo. Por ejemplo, en un teléfono, los usuarios verían el contenido que se muestra en una vista de una sola columna mientras que una tableta puede mostrar el mismo contenido en dos columnas.

0:02 / 0:14



Existe una multitud de tamaños de pantalla diferentes en teléfonos, "phablets", tabletas, computadoras de escritorio, consolas de juegos, televisores e incluso dispositivos portátiles. Los tamaños de pantalla cambian constantemente, por lo que es importante que su sitio pueda adaptarse a cualquier tamaño de pantalla, hoy o en el futuro. Además, los dispositivos tienen distintas características con las que interactuamos con ellos. Por ejemplo, algunos de sus visitantes utilizarán una pantalla táctil. El diseño responsivo moderno considera todas estas cosas para optimizar la experiencia para todos.

By Chrome DevRel

jurar la ventana gráfica

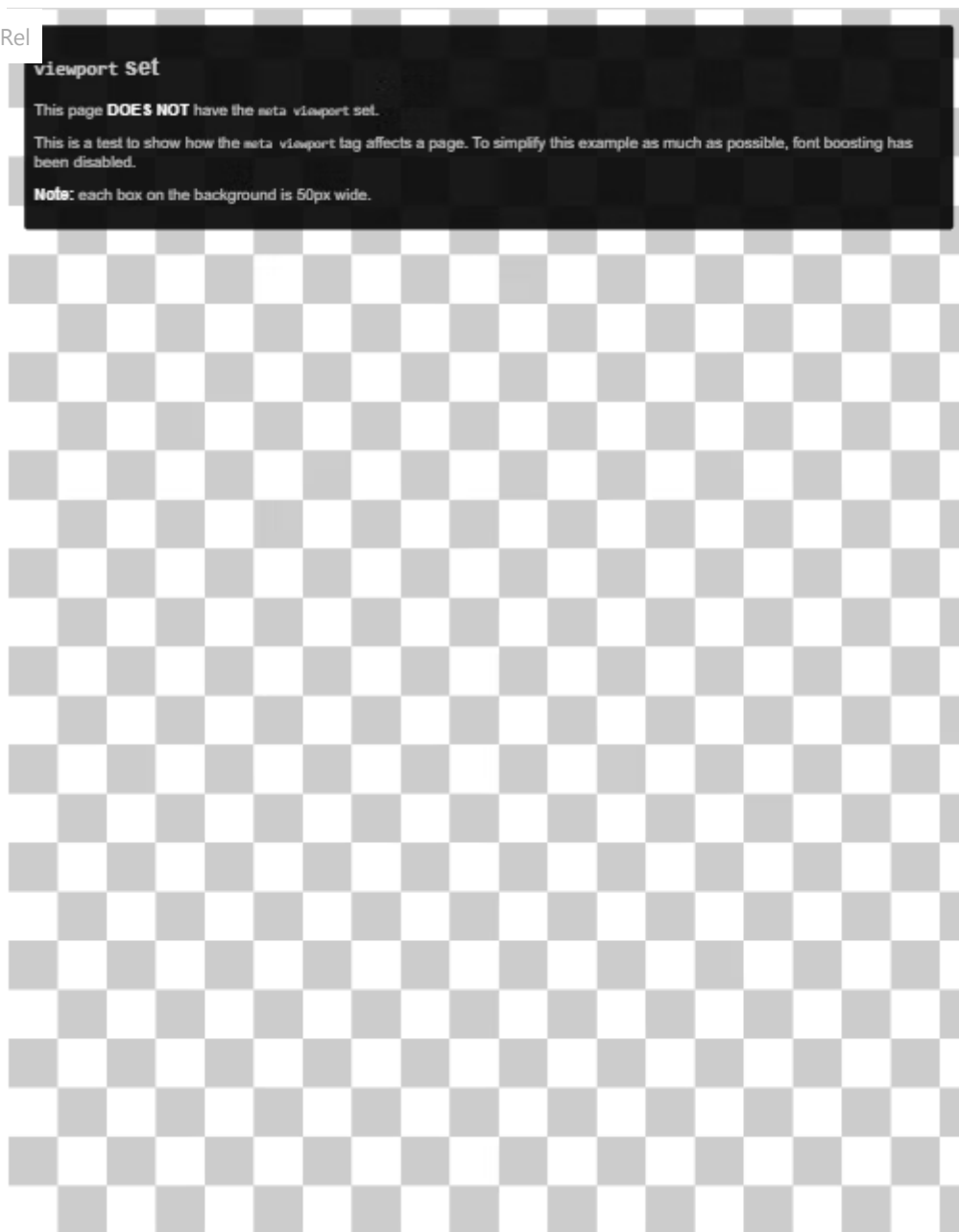
Las páginas optimizadas para una variedad de dispositivos deben incluir una meta etiqueta de ventana gráfica en el encabezado del documento. Una meta etiqueta de ventana gráfica le da al navegador instrucciones sobre cómo controlar las dimensiones y la escala de la página.

Para intentar brindar la mejor experiencia, los navegadores móviles representan la página con un ancho de pantalla de escritorio (generalmente alrededor de `980px`, aunque esto varía según los dispositivos) y luego intentan hacer que el contenido se vea mejor aumentando el tamaño de la fuente y escalando el contenido para que se ajuste al tamaño de la pantalla. Esto significa que los tamaños de fuente pueden parecer inconsistentes para los usuarios que deben de tocar dos veces o pellizcar para hacer zoom para ver e interactuar con el contenido.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    ...
    <meta name="viewport" content="width=device-width, initial-scale=1">
    ...
  </head>
  ...
```

El uso del valor de la meta etiqueta de ventana gráfica `width=device-width` indica a la página que coincida con el ancho de la pantalla en píxeles independientes del dispositivo. Un dispositivo (o densidad) de píxeles independientes es una representación de un solo píxel, el cual en una pantalla de alta densidad puede constar de muchos píxeles físicos. Esto permite que la página redistribuya el contenido para que coincida con diferentes tamaños de pantalla, ya sea renderizado en un teléfono móvil pequeño o en un monitor de escritorio grande.

By Chrome DevRel



Un ejemplo de cómo se carga la página en un dispositivo sin la meta etiqueta de la ventana gráfica. [Vea este ejemplo en Glitch](#).

By Chrome DevRel

viewport set

This page **DOES** have the meta viewport set.

This is a test to show how the meta viewport tag affects a page. To simplify this example as much as possible, font boosting has been disabled.

Note: each box on the background is 50px wide.

Un ejemplo de cómo se carga la página en un dispositivo con la meta etiqueta de la ventana gráfica. [Vea el ejemplo en Glitch](#).

By Chrome DevRel [vegadores](#) mantienen constante el ancho de la página cuando se gira al modo horizontal y hacen zoom en lugar de volver a acomodar para llenar la pantalla. Agregar el valor `initial-scale=1` indica a los navegadores que establezcan una relación 1:1 entre los píxeles del CSS y los píxeles independientes del dispositivo, independientemente de la orientación del dispositivo, y permite que la página aproveche todo el ancho.

! Precaución

Para asegurarse de que los navegadores más antiguos puedan analizar correctamente los atributos, use una coma para separarlos.

La auditoría Lighthouse de [No tiene una etiqueta <meta name="viewport"> con width o initial-scale](#) puede ayudarle a automatizar el proceso de asegurarse de que sus documentos HTML estén usando la meta etiqueta de ventana gráfica correctamente.

Asegurar una ventana gráfica accesible

Además de establecer el `initial-scale`, también puede establecer los siguientes atributos en la ventana gráfica:

`minimum-scale`

`maximum-scale`

`user-scalable`

Cuando se configuran, estos pueden deshabilitar la capacidad del usuario para hacer zoom en la ventana gráfica, lo que podría causar problemas de accesibilidad. Por lo tanto, no recomendamos utilizar estos atributos.

Adapte el contenido a la ventana gráfica

By Chrome DevRel escritorio como en los dispositivos móviles, los usuarios están acostumbrados a desplazarse por los sitios web verticalmente pero no horizontalmente, obligar al usuario a desplazarse horizontalmente o alejarse para ver toda la página resulta en una mala experiencia de usuario.

Al desarrollar un sitio móvil con una meta etiqueta de ventana gráfica, es fácil crear accidentalmente contenido de página que no encaja del todo dentro de la ventana gráfica especificada. Por ejemplo, una imagen que se muestra con un ancho más grande que la ventana gráfica puede hacer que esta se desplace horizontalmente. Debe ajustar este contenido para que se ajuste al ancho de la ventana gráfica, de modo que el usuario no necesite desplazarse horizontalmente.

La auditoría de Lighthouse del [Contenido no tiene el tamaño adecuado para la ventana gráfica](#) puede ayudarle a automatizar el proceso de detección del contenido desbordado.

Imágenes

Una imagen tiene dimensiones fijas y si es más grande que la ventana gráfica, se creará una barra de desplazamiento. Una forma habitual de solucionar este problema es dar a todas las imágenes un `max-width` del `100%`. Esto hará que la imagen se encoja para ajustarse al espacio disponible, en caso de que el tamaño de la ventana gráfica sea más pequeño que el de la imagen. Sin embargo, debido a que el `max-width`, en lugar del `width` es del `100%`, la imagen no se estirará más que su tamaño natural. Por lo general, es seguro agregar lo siguiente a su hoja de estilo para que nunca tenga problemas con las imágenes que pueden causar una barra de desplazamiento.

```
img {  
  max-width: 100%;  
  display: block;  
}
```

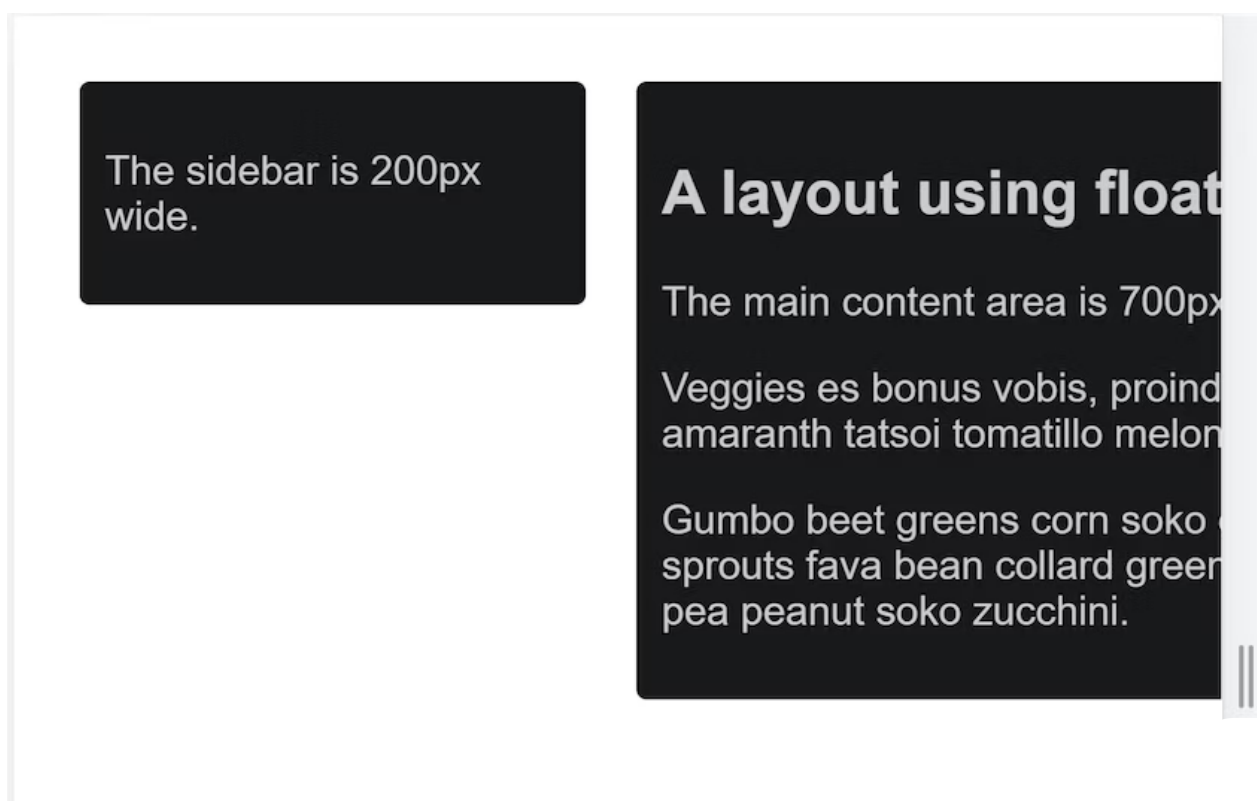
Agregue las dimensiones de la imagen al elemento `img`

By Chrome DevRel } `max-width: 100%`, está anulando las dimensiones naturales de la imagen, sin embargo, aún debe usar los atributos de `width` y `height` en su etiqueta ``. Esto se debe a que los navegadores modernos usarán esta información para reservar espacio para la imagen antes de que se cargue, esto ayudará a evitar [cambios de diseño](#) a medida que se carga el contenido.

Diseño

Dado que las dimensiones y el ancho de la pantalla en píxeles del CSS varían ampliamente entre dispositivos (por ejemplo, entre teléfonos y tabletas, e incluso entre diferentes teléfonos), el contenido no debe depender de un ancho de ventana en particular para renderizarse bien.

En el pasado, esto requería elementos de configuración que se usaban para crear un diseño basado en porcentajes. En el siguiente ejemplo, puede ver un diseño de dos columnas con elementos flotantes, con un tamaño con base en píxeles. Una vez que la ventana gráfica se vuelva más pequeña que el ancho total de las columnas, tenemos que desplazarnos horizontalmente para ver el contenido.



Un diseño flotante que utiliza píxeles. [Vea este ejemplo en Glitch](#).

By Chrome DevRel centajes para los anchos, las columnas siempre quedan como un cierto porcentaje del contenedor. Esto significa que las columnas se vuelven más estrechas, en lugar de crear una barra de desplazamiento.

The sidebar is
27% wide.

A layout using floats

The main content area is 70% wide.

Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

Gumbo beet greens corn soko endive gumbo gourd. Parsley shallot courgette tatsoi pea sprouts fava bean collard greens dandelion okra wakame tomato. Dandelion cucumber earthnut pea peanut soko zucchini.

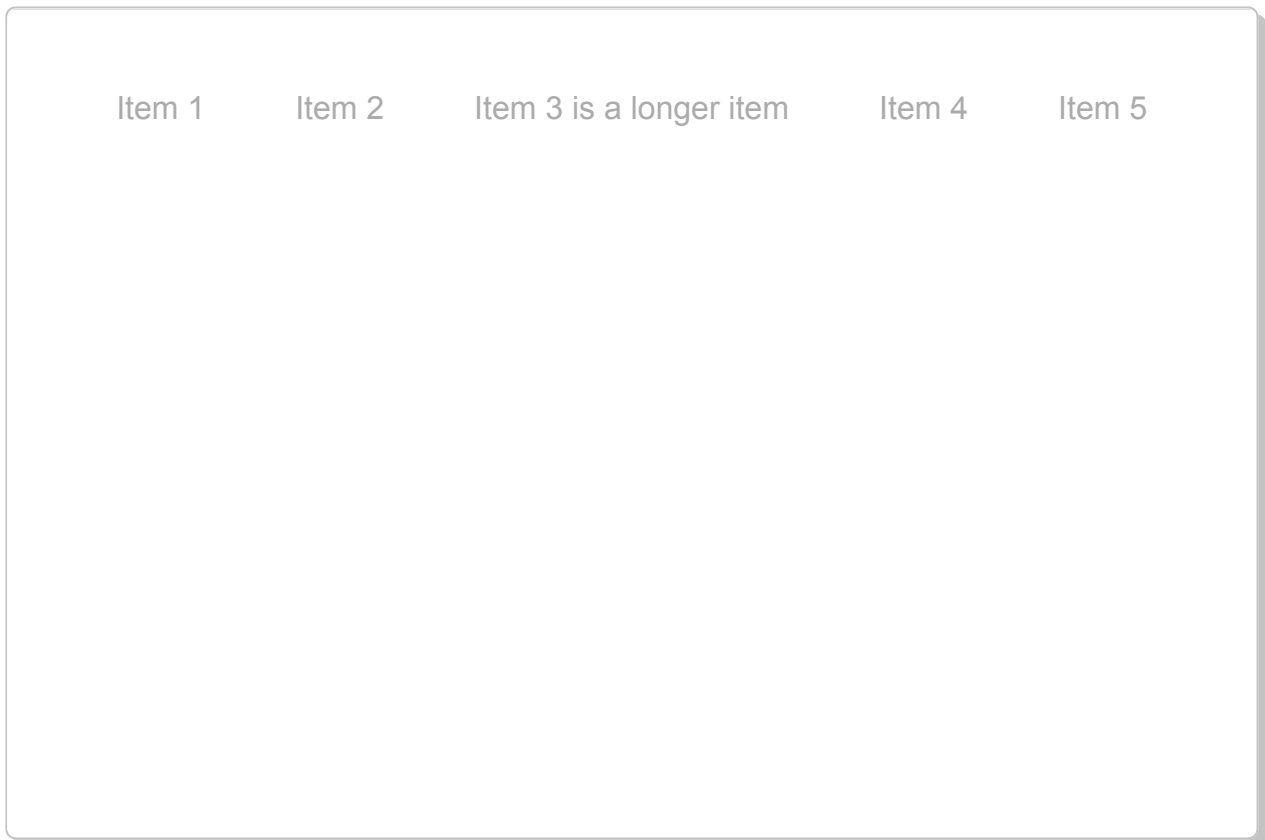
Las técnicas modernas de diseño de CSS como Flexbox, Diseño de cuadrículas y Multicol facilitan mucho la creación de estas cuadrículas flexibles.

Flexbox

Este método de diseño es ideal cuando tiene un conjunto de elementos de diferentes tamaños y le gustaría que encajen cómodamente en una fila o más, donde los elementos más pequeños ocupan menos espacio y los más grandes tienen más espacio.

```
.items {  
  display: flex;  
  justify-content: space-between;  
}
```

By Chrome DevRel | io responsivo, puede usar Flexbox para mostrar elementos como una sola fila o envueltos en varias filas a medida que disminuye el espacio disponible.



[Leer más sobre Flexbox.](#)

Diseño de cuadrículas CSS

El diseño de cuadrícula CSS permite la creación sencilla de cuadrículas flexibles. Si consideramos el ejemplo anterior, en lugar de crear nuestras columnas con porcentajes, podríamos usar el diseño de cuadrícula y la `fr`, que representa una parte del espacio disponible en el contenedor.

```
.container {  
  display: grid;  
  grid-template-columns: 1fr 3fr;  
}
```

By Chrome DevRel

The sidebar is
1fr.

A grid layout

The main content area is 3fr.

Veggies es bonus vobis, proinde vos postulo essum magis kohlrabi welsh onion daikon amaranth tatsoi tomatillo melon azuki bean garlic.

Gumbo beet greens corn soko endive gumbo gourd. Parsley shallot courgette tatsoi pea sprouts fava bean collard greens dandelion okra wakame tomato. Dandelion cucumber earthnut pea peanut soko zucchini.

La cuadrícula también se puede utilizar para crear diseños de cuadrícula regulares, con tantos elementos como quepan. El número de pistas disponibles se reducirá a medida que se reduzca el tamaño de la pantalla. En la siguiente demostración, tenemos tantas tarjetas como quepan en cada fila, con un tamaño mínimo de `200px`.

One

Two

Three

Four

Five

Six

Seven

By Chrome DevRel [sobre el diseño de cuadrículas CSS](#)

Diseño de varias columnas

Para ciertos tipos de diseño, puede usar un diseño de columnas múltiples (Multicol), que puede crear varias columnas responsivas con la propiedad de `column-width`. En la siguiente demostración, puede ver que se agregan columnas si hay espacio para otra columna de `200px`.

Gumbo beet greens corn soko endive gumbo gourd. Parsley shallot courgette tatsoi pea sprouts fava bean collard greens dandelion okra wakame tomato. Dandelion cucumber earthnut pea peanut soko zucchini.

Turnip greens yarrow ricebean rutabaga endive cauliflower sea lettuce kohlrabi amaranth water spinach avocado daikon napa cabbage asparagus winter purslane kale. Celery potato scallion desert raisin horseradish spinach carrot soko. Lotus root water spinach fennel kombu maize bamboo shoot green bean swiss chard seakale pumpkin onion chickpea gram corn pea. Brussels sprout coriander water chestnut gourd swiss chard wakame kohlrabi beetroot carrot watercress. Corn amaranth salsify bunya nuts nori azuki bean chickweed potato bell pepper artichoke.

Nori grape silver beet broccoli kombu beet greens fava bean potato quandong celery. Bunya nuts black-eyed pea prairie turnip leek lentil turnip greens parsnip. Sea lettuce lettuce water chestnut eggplant winter purslane fennel azuki bean earthnut pea sierra leone bologi leek soko chicory celtuce parsley jícama salsify.

Celery quandong swiss chard chicory earthnut pea potato. Salsify taro catsear garlic gram celery bitterleaf wattle seed collard greens nori. Grape wattle seed kombu beetroot horseradish carrot squash brussels

[Leer más sobre Multicol](#)

Utilice consultas de medios CSS para mejorar la capacidad de respuesta

En ocasiones, necesitará realizar cambios más extensos en su diseño para permitir a un cierto tamaño de pantalla fuera de lo que permiten las técnicas que se muestran arriba. Aquí es donde las consultas de medios se vuelven útiles.

Las consultas de medios son filtros simples que se pueden aplicar a los estilos CSS. Facilitan el cambio de estilos según los tipos de dispositivos que despliegan el

By Chrome DevRel > por las características de ese dispositivo, por ejemplo, ancho, alto, orientación, capacidad para desplazarse y si el dispositivo se está utilizando como pantalla táctil.

Para proporcionar diferentes estilos de impresión, debe apuntar a un *tipo* de salida para poder incluir una hoja de estilo con los estilos de impresión de la siguiente manera:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    ...
    <link rel="stylesheet" href="print.css" media="print">
    ...
  </head>
  ...
```

Alternativamente, puede incluir estilos de impresión dentro de su hoja de estilo principal mediante una consulta de medios:

```
@media print {
  /* print styles go here */
}
```

También es posible incluir hojas de estilo separadas en el archivo principal de CSS mediante el sintaxis de `@import`, `@import url(print.css) print;`, sin embargo, este uso no se recomienda por razones de rendimiento. Consulte [Evitar importaciones de CSS](#) para obtener más detalles.

Para el diseño web responsivo, generalmente consultamos las *características* del dispositivo para proporcionar un diseño diferente para pantallas más pequeñas, o cuando detectamos que nuestro visitante está usando una pantalla táctil.

By Chrome DevRel **as de medios basadas en el tamaño de la ventana gráfica**

Las consultas de medios nos permiten crear una experiencia responsiva donde se aplican estilos específicos a pantallas pequeñas, pantallas grandes y cualquier tamaño intermedio. La característica que estamos detectando aquí es el tamaño de la pantalla, con la cual podemos probar las siguientes cosas.

`width` (`min-width` `max-width`)

`height` (`min-height` `max-height`)

`orientation`

`aspect-ratio`

Testing Media Queries

Resize your browser window to find out how media queries are reporting the size and orientation of this page.

This browser window is smaller than 640px wide.

The orientation is Landscape.

Todas estas características tienen una excelente compatibilidad con los navegadores; para obtener más detalles, incluyendo la información de soporte del navegador, consulte el [ancho](#) , la [altura](#) , la [orientación](#) y [la relación de aspecto](#) en MDN.

By Chrome DevRel

La especificación incluyó pruebas para `device-width` y `device-height`. Estos han quedado obsoletos y deben evitarse. `device-width` y `device-height` fueron utilizados para obtener el tamaño actual de la ventana del dispositivo, lo que no fue útil en práctica porque puede ser diferente de la ventana gráfica que está mirando el usuario, por ejemplo, al cambiar el tamaño de la ventana del navegador.

Consultas de medios basadas en la capacidad del dispositivo

Dada la gama de dispositivos disponibles, no podemos suponer que todos los dispositivos grandes son una computadora de escritorio o una computadora laptop, o que las personas solo usan una pantalla táctil en un dispositivo pequeño. Con algunas adiciones más recientes a la especificación de consultas de medios, podemos probar características como el tipo de puntero utilizado para interactuar con el dispositivo y si el usuario puede desplazarse sobre los elementos.

`hover``pointer``any-hover``any-pointer`

Intente ver esta demostración en diferentes dispositivos, como una computadora de escritorio normal y en un teléfono o tableta.

By Chrome DevRel

Testing Media Features for pointer and hover

View this demo using different devices.

Your device allows you to hover. You are using a fine pointer, a mouse or trackpad perhaps?


Estas funciones más recientes tienen un buen soporte en todos los navegadores modernos. Obtenga más información en las páginas de MDN para [hover](#), [any-hover](#), [pointer](#), [any-pointer](#).

Usando any-hover y any-pointer

Las funciones de `any-hover` y `any-pointer` comprueban si el usuario tiene la capacidad de pasar el ratón sobre un elemento (hover) o usar ese tipo de puntero, incluso si no es la forma principal en la que interactúa con su dispositivo. Tenga mucho cuidado al usarlos. ¡Obligar a un usuario a cambiar a un ratón cuando está usando su pantalla táctil no es algo fácil! Sin embargo, `any-hover` y `any-pointer` pueden ser útiles si es importante averiguar qué tipo de dispositivo tiene un usuario. Por ejemplo, una laptop con pantalla táctil y trackpad debe coincidir con punteros gruesos y finos, además de la capacidad de situar el puntero sobre un elemento.

Cómo elegir los breakpoints

No defina los breakpoints basados en clases de dispositivos. La definición de breakpoints basados en dispositivos, productos, marcas o sistemas operativos específicos que se utilizan hoy en día puede resultar en una pesadilla de

By Chrome DevRel . En cambio, el contenido en sí debería determinar cómo se ajusta el diseño a su contenedor.

Elija los breakpoints principales comenzando con algo pequeño y partir de ahí

Primero diseñe el contenido para que se ajuste a un tamaño de pantalla pequeño, luego expanda la pantalla hasta que sea necesario un breakpoint. Esto le permite optimizar los breakpoints según el contenido y mantener la menor cantidad de breakpoints posible.

Trabajemos en el ejemplo que vimos al principio: el pronóstico del clima. El primer paso es hacer que el pronóstico se vea bien en una pantalla pequeña.

By Chrome DevRel

New York, NY

Tuesday, April 15th
Overcast



58°F

Precipitation: 100%
Humidity: 97%
Wind: 4 mph SW
Pollen Count: 36

Today		68° 36°	Pollen 36
Wednesday		50° 39°	Pollen 36
Thursday		55° 39°	Pollen 36
Friday		54° 43°	Pollen 36
Saturday		64° 46°	Pollen 36
Sunday		64° 50°	Pollen 36
Monday		61° 50°	Pollen 36

La aplicación en una anchura muy estrecha.

A continuación, cambie el tamaño del navegador hasta que haya demasiado espacio en blanco entre los elementos y el pronóstico del clima no se vea tan bien. La decisión es algo subjetiva, pero por encima de `600px` es ciertamente demasiado amplia.

By Chrome DevRel



La aplicación en un punto en el que creemos que deberíamos modificar el diseño.

Para insertar un breakpoint en `600px`, cree dos consultas de medios al final de su CSS para el componente, una que se utilice cuando el navegador tenga `600px` o menos, y otra para cuando sea más ancho que `600px`.

```
@media (max-width: 600px) {  
  
}  
  
@media (min-width: 601px) {  
  
}
```

Finalmente, refactorice el CSS. Dentro de la consulta de medios para un `max-width` de `600px`, agregue el CSS que es solo para pantallas pequeñas. Dentro de la consulta de medios para un `min-width` mínimo de `601px` agregue CSS para pantallas más grandes.

Elija breakpoints menores cuando sea necesario

Además de elegir breakpoints importantes cuando el diseño cambia significativamente, también es útil ajustarlos para cambios menores. Por ejemplo, entre los breakpoints principales puede resultar útil ajustar los márgenes o el relleno de un elemento, o aumentar el tamaño de la fuente para que se sienta más natural en el diseño.

By Chrome DevRel os optimizando el diseño de la pantalla pequeña. En este caso, aumentemos la fuente cuando el ancho de la ventana sea `360px`. De manera seguida, cuando hay suficiente espacio, podemos separar las temperaturas altas y bajas para que estén en la misma línea en lugar de una encima de la otra. Y también hagamos un poco más grandes los íconos del clima.

```
@media (min-width: 360px) {  
  body {  
    font-size: 1.0em;  
  }  
}  
  
@media (min-width: 500px) {  
  .seven-day-fc .temp-low,  
  .seven-day-fc .temp-high {  
    display: inline-block;  
    width: 45%;  
  }  
  
  .seven-day-fc .seven-day-temp {  
    margin-left: 5%;  
  }  
  
  .seven-day-fc .icon {  
    width: 64px;  
    height: 64px;  
  }  
}
```

Del mismo modo, para las pantallas grandes, es mejor limitar el ancho máximo del panel de pronóstico para que no consuma todo el ancho de la pantalla.

```
@media (min-width: 700px) {  
  .weather-forecast {  
    width: 700px;  
  }  
}
```

By Chrome DevRel

New York, NY

Tuesday, April 15th
Overcast

58 °F

Precipitation: 100%
Humidity: 97%
Wind: 4 mph SW
Pollen Count: 36

Today

Wednesday

Thursday

Friday

Saturday

Sunday

Monday

68° 36°
Pollen 36

50° 39°
Pollen 36

55° 39°
Pollen 36

54° 43°
Pollen 36

64° 46°
Pollen 36

64° 50°
Pollen 36

61° 50°
Pollen 36

Optimiza el texto para leer

La teoría clásica de la legibilidad sugiere que una columna ideal debe contener de 70 a 80 caracteres por línea (aproximadamente de 8 a 10 palabras en inglés). Por lo tanto, cada vez que el ancho de un bloque de texto supere las 10 palabras, considere agregar un breakpoint.

Bacon Ipsum

Bacon ipsum dolor sit amet pancetta salami beef ribs ribeye ham. Ribeye strip steak boudin t-bone chuck tri-tip kielbasa sirloin frankfurter shankle ball tip ham short loin hamburger pork chop. Doner jowl tail, shoulder chicken beef ribs ribeye leberkas. Tongue corned beef fatback turkey pork chop bacon. Pork meatloaf andouille doner. Swine biltong ham pork chop tongue, shankle shank sausage jowl landjaeger bresaola boudin pork. Porchetta shank beef tongue cow, pork flank shankle rump pig.

Ground round shankle tongue chicken kevin fatback pancetta boudin strip steak doner shank drumstick pork loin ball tip short ribs. Kevin rump chicken ribeye salami tail andouille. Tenderloin shoulder drumstick biltong jowl short

El texto como se lee en un dispositivo móvil.

Bacon Ipsum

Bacon ipsum dolor sit amet pancetta salami beef ribs ribeye ham. Ribeye strip steak boudin t-bone chuck tri-tip kielbasa sirloin frankfurter shankle ball tip ham short loin hamburger pork chop. Doner jowl tail, shoulder chicken beef ribs ribeye leberkas. Tongue corned beef fatback turkey pork chop bacon. Pork meatloaf andouille doner. Swine biltong ham pork chop tongue, shankle shank sausage jowl landjaeger bresaola boudin pork. Porchetta shank beef tongue cow, pork flank shankle rump pig.

Ground round shankle tongue chicken kevin fatback pancetta boudin strip steak doner shank drumstick pork loin ball tip short ribs. Kevin rump chicken ribeye salami tail andouille. Tenderloin shoulder drumstick biltong jowl short ribs capicola. Ham hock pastrami spare ribs tail doner frankfurter.

Pancetta ground round hamburger, tenderloin tail porchetta filet mignon prosciutto bacon. Prosciutto rump spare ribs boudin meatloaf doner flank chuck. Prosciutto leberkas ground round, sirloin tail flank filet mignon spare ribs ham turducken jowl swine pork loin shankle. Tri-tip rump doner andouille tail, bacon ham jerky. Rump shank pastrami, sausage brisket chicken ribeye frankfurter andouille pork chop pork pancetta. Jowl tongue filet mignon jerky, prosciutto strip steak pastrami rump.

Bresaola pork kielbasa ham hock tail cow short ribs sirloin kevin fathack

El texto como se lee en un navegador de escritorio con un breakpoint agregado para restringir la longitud de la línea.

Echemos un vistazo más profundo al ejemplo de la publicación anterior del blog. F pantallas más pequeñas, la fuente Roboto en `1em` funciona perfectamente

By Chrome DevRel o 10 palabras por línea, pero las pantallas más grandes requieren un breakpoint. En este caso, si el ancho del navegador es superior a `575px`, el ancho ideal del contenido es `550px`.

```
@media (min-width: 575px) {  
  article {  
    width: 550px;  
    margin-left: auto;  
    margin-right: auto;  
  }  
}
```

Bacon Ipsum

Bacon ipsum dolor amet prosciutto pancetta porchetta pork brisket pork belly short loin doner frankfurter, cupim shankle boudin beef salami drumstick. Tri-tip ball tip salami, jowl shank pig beef ribs frankfurter biltong. Venison pork loin pork belly tail pig strip steak corned beef porchetta cupim cow burgdoggen ribeye beef ribs. Short ribs rump pork beef. Meatball tenderloin ball tip shankle, shoulder swine filet mignon.

Brisket strip steak ball tip pork chop pancetta pork boudin filet mignon doner venison. Tri-tip capicola chicken salami pastrami bresaola tenderloin strip steak chuck. Tail prosciutto ball tip pork filet mignon frankfurter spare ribs beef ribs, strip steak kielbasa sirloin tongue. Burgdoggen salami pork chop doner venison. Ball tip venison beef ribs shank andouille turducken spare ribs pork chop biltong. Hamburger landjaeger t-bone, cupim ground round swine porchetta spare ribs brisket jerky flank pork ribeye short ribs tongue. Fatback ground round cow tail ball tip beef.

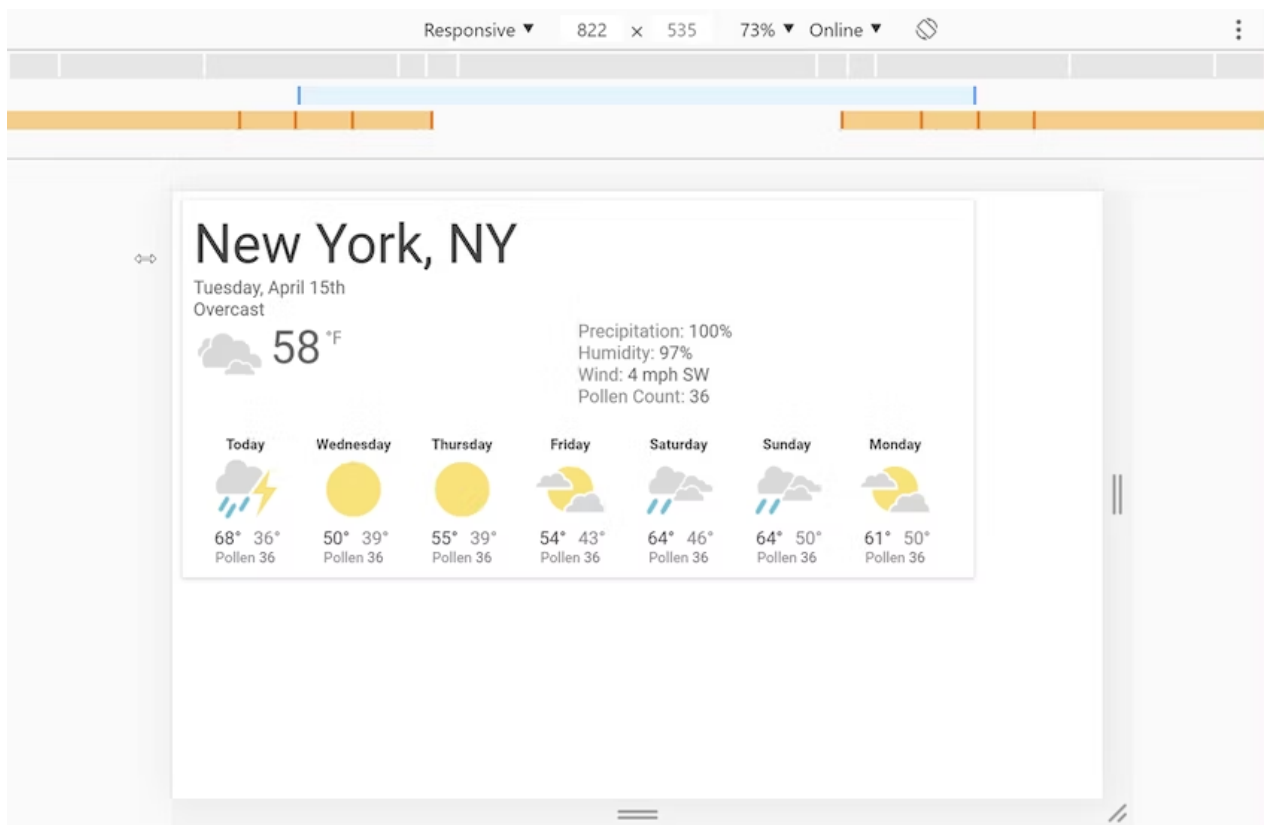
Turducken porchetta ground round pastrami, short ribs capicola short loin rump ribeye chislic. Tenderloin swine spare ribs, chicken ball tip venison beef ribs capicola ground round. Turducken kevin drumstick iowl buradodaen

Evite ocultar contenido

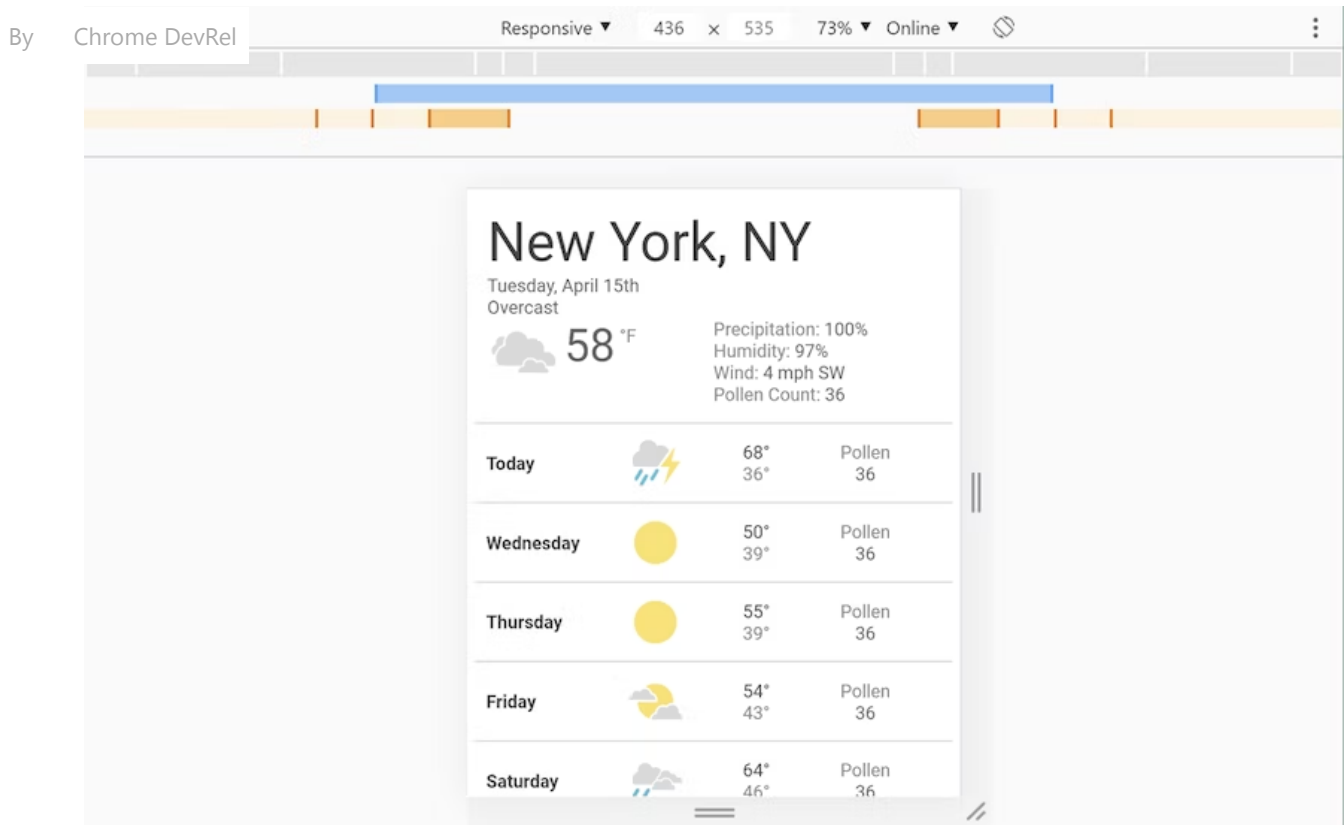
Tenga cuidado al elegir qué contenido ocultar o mostrar según el tamaño de la pantalla. No oculte el contenido simplemente porque no puede caber en la pantalla. El tamaño de la pantalla no es una indicación definitiva de lo que puede querer el usuario. Por ejemplo, eliminar el recuento de polen del pronóstico del clima podría ser un problema grave para las personas alérgicas en primavera que necesitan la información para determinar si pueden salir o no.

By Chrome DevRel Consultas de medios de breakpoints en Chrome DevTools

Una vez que haya configurado sus consultas de medios de breakpoints, querrá ver cómo se ve su sitio con ellos. Puede cambiar el tamaño de la ventana de su navegador para activar los puntos de interrupción, pero Chrome DevTools tiene una función incorporada que facilita ver cómo se ve una página en diferentes puntos de interrupción.



DevTools muestra la aplicación del pronóstico del clima a medida que se ve en un tamaño de ventana más amplio.



DevTools muestra la aplicación del pronóstico del clima a medida que se ve en un tamaño de ventana gráfica más estrecho.

Para ver su página en diferentes breakpoints:

[Abra DevTools](#) y luego active el [modo de dispositivo](#). Esto se abre en [modo responsivo](#) de forma predeterminada.

Para ver sus consultas de medios, abra el menú de modo de dispositivo y seleccione [Mostrar consultas de medios](#) para mostrar sus breakpoints como barras de colores sobre su página.

Haga clic en una de las barras para ver su página mientras esa consulta de medios está activa. Haga clic con el botón derecho en una barra para saltar a la definición de la consulta de medios.

[CSS](#)[Diseño](#)[Dispositivos móviles](#)[Experiencia del usuario \(UX\)](#)

Última actualización

[← RETURN TO ALL ARTICLES](#)



We want to help you build beautiful, accessible, fast, and secure websites that work cross-browser, and for all of your users. This site is our home for content to help you on that journey, written by members of the Chrome team, and external experts.

Contribute

[Presentar un error](#)

[Ver fuente](#)

Contenido relevante

[developer.chrome.com](#)

[Chrome Actualizaciones](#)

[Case studies](#)

[Podcasts](#)

[Shows](#)

Conectar

[Twitter](#)

[YouTube](#)



[Chrome](#)

[Firebase](#)

[Google Cloud Platform](#)

[Todos los productos](#)

Dark theme ☐

ESPAÑOL (es)

By Chrome DevRel

[Condiciones y privacidad](#)[Principios de la Comunidad](#)

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see the [Google Developers Site Policies](#).