

Guía para principiantes sobre consultas de medios

[Anterior](#)

[Descripción general: diseño CSS](#)

[próximo](#)

CSS Media Query le ofrece una forma de aplicar CSS solo cuando el navegador y el entorno del dispositivo coinciden con una regla que usted especifica, por ejemplo, "la ventana gráfica es más ancha que 480 píxeles". Las consultas de medios son una parte clave del diseño web receptivo, ya que le permiten crear diferentes diseños según el tamaño de la ventana gráfica, pero también se pueden usar para detectar otras cosas sobre el entorno en el que se ejecuta su sitio, por ejemplo, si el usuario está usando una pantalla táctil en lugar de un mouse. En esta lección, primero aprenderá sobre la sintaxis utilizada en las consultas de medios y luego pasará a usarlas en un ejemplo práctico que muestra cómo se puede hacer que un diseño simple responda.

requisitos
previos:

Conceptos básicos de HTML (estudiar [Introducción a HTML](#)) y una idea de cómo funciona CSS (estudiar [los primeros pasos](#) de CSS y [los componentes básicos](#) de CSS).

Objetivo:

Comprender cómo usar las consultas de medios y el enfoque más común para usarlas para crear diseños receptivos.

Conceptos básicos de consulta de medios

La sintaxis de consulta de medios más simple se ve así:

```
@media media-type and (media-feature-rule) {  
  /* CSS rules go here */  
}
```

Consiste en:

- Un tipo de medio, que le dice al navegador para qué tipo de medio es este código (por ejemplo, impreso o pantalla).
- Una expresión multimedia, que es una regla o prueba que se debe pasar para que se aplique el CSS contenido.
- Un conjunto de reglas CSS que se aplicarán si pasa la prueba y el tipo de medio es correcto.

Tipos de medios

Los posibles tipos de medios que puede especificar son:

- all
- print
- screen

La siguiente consulta de medios solo establecerá el cuerpo en 12 puntos si la página está impresa. No se aplicará cuando la página se cargue en un navegador.

```
@media print {  
  body {  
    font-size: 12pt;  
  }  
}
```

Nota: el tipo de medio aquí es diferente del llamado tipo MIME .

Nota: hubo una serie de otros tipos de medios definidos en la especificación de consultas de medios de nivel 3; estos han sido obsoletos y deben evitarse.

Nota: los tipos de medios son opcionales; si no indica un tipo de medio en su consulta de medios, la consulta de medios será predeterminada para todos los tipos de medios.

Reglas de funciones multimedia

Después de especificar el tipo, puede orientar una función multimedia con una regla.

Anchura y altura

La función que tendemos a detectar con más frecuencia para crear diseños receptivos (y que cuenta con una amplia compatibilidad con los navegadores) es el ancho de la ventana gráfica, y podemos aplicar CSS si la ventana gráfica está por encima o por debajo de cierto ancho, o un ancho exacto, usando el `min-width` , `max-width` y `width` funciones multimedia.

Estas funciones se utilizan para crear diseños que responden a diferentes tamaños de pantalla. Por ejemplo, para cambiar el color del texto del cuerpo a rojo si la ventana gráfica tiene exactamente 600 píxeles, usaría la siguiente consulta de medios.

```
@media screen and (width: 600px) {  
  body {
```

```
    color: red;
  }
}
```

[Abra este ejemplo](#) en el navegador o [vea la fuente](#) .

Las funciones de medios `width` (y `height`) se pueden usar como rangos y, por lo tanto, pueden tener el prefijo `min-` o `max-` para indicar que el valor dado es un mínimo o un máximo. Por ejemplo, para hacer que el color sea azul si la ventana gráfica es más estrecha que 600 píxeles, use `max-width` :

```
@media screen and (max-width: 600px) {
  body {
    color: blue;
  }
}
```

[Abra este ejemplo](#) en el navegador o [vea la fuente](#) .

En la práctica, usar valores mínimos o máximos es mucho más útil para el diseño receptivo, por lo que rara vez verá `width` o `height` usará solo.

Hay una serie de otras funciones de medios que puede probar, aunque algunas de las funciones más nuevas introducidas en los niveles 4 y 5 de la especificación de consultas de medios tienen compatibilidad limitada con el navegador. Cada función está documentada en MDN junto con la información de soporte del navegador, y puede encontrar una lista completa en [Uso de consultas de medios: funciones de medios](#) .

Orientación

Una función de medios bien admitida es `orientation` , que nos permite probar el modo vertical u horizontal. Para cambiar el color del texto del cuerpo si el dispositivo está en orientación horizontal, use la siguiente consulta de medios.

```
@media (orientation: landscape) {
  body {
    color: rebeccapurple;
  }
}
```

[Abra este ejemplo](#) en el navegador o [vea la fuente](#) .

Una vista de escritorio estándar tiene una orientación horizontal y un diseño que funciona bien en esta orientación puede no funcionar tan bien cuando se ve en un teléfono o tableta en modo vertical. La prueba de orientación puede ayudarlo a crear un diseño optimizado para dispositivos en modo vertical.

Uso de dispositivos señaladores.

Como parte de la especificación de Nivel 4, `hover` se introdujo la función multimedia. Esta característica significa que puede probar si el usuario tiene la capacidad de pasar el mouse sobre un elemento, lo que esencialmente significa que está usando algún tipo de dispositivo señalador; la pantalla táctil y la navegación con teclado no se desplazan.

```
@media (hover: hover) {  
  body {  
    color: rebeccapurple;  
  }  
}
```

[Abra este ejemplo](#) en el navegador o [vea la fuente](#) .

Si sabemos que el usuario no puede desplazarse, podríamos mostrar algunas funciones interactivas de forma predeterminada. Para los usuarios que pueden pasar el mouse, podemos elegir que estén disponibles cuando se pasa el mouse sobre un enlace.

También en el Nivel 4 está la `pointer` función multimedia. Esto toma tres valores posibles, `none` , `fine` y `coarse` . Un `fine` puntero es algo así como un mouse o un trackpad. Permite al usuario apuntar con precisión a un área pequeña. Un `coarse` puntero es su dedo en una pantalla táctil. El valor `none` significa que el usuario no tiene ningún dispositivo señalador; quizás estén navegando solo con el teclado o con comandos de voz.

El uso `pointer` puede ayudarlo a diseñar mejores interfaces que respondan al tipo de interacción que un usuario tiene con una pantalla. Por ejemplo, podría crear áreas de acceso más grandes si sabe que el usuario está interactuando con el dispositivo como una pantalla táctil.

Consultas de medios más complejas

Con todas las diferentes consultas de medios posibles, es posible que desee combinarlas o crear listas de consultas, cualquiera de las cuales podría coincidir.

Lógica "y" en consultas de medios

Para combinar funciones de medios, puede usar `and` de la misma manera que hemos usado `and` anteriormente para combinar un tipo de medio y una función. Por ejemplo, podríamos querer probar para `min-width` y `orientation` . El cuerpo del texto solo será azul si la ventana gráfica tiene al menos 600 píxeles de ancho y el dispositivo está en modo horizontal.

```
@media screen and (min-width: 600px) and (orientation: landscape) {  
  body {  
    color: blue;  
  }  
}
```

[Abra este ejemplo](#) en el navegador o [vea la fuente](#) .

Lógica "o" en consultas de medios

Si tiene un conjunto de consultas, cualquiera de las cuales podría coincidir, entonces puede separarlas con comas. En el siguiente ejemplo, el texto será azul si la ventana gráfica tiene al menos 600 píxeles de ancho O si el dispositivo está en orientación horizontal. Si alguna de estas cosas es verdadera, la consulta coincide.

```
@media screen and (min-width: 600px), screen and (orientation: landscape) {  
  body {  
    color: blue;  
  }  
}
```

[Abra este ejemplo](#) en el navegador o [vea la fuente](#) .

Lógica "no" en consultas de medios

Puede negar una consulta de medios completa utilizando el `not` operador. Esto invierte el significado de toda la consulta de medios. Por lo tanto, en el siguiente ejemplo, el texto solo será azul si la orientación es vertical.

```
@media not all and (orientation: landscape) {  
  body {  
    color: blue;  
  }  
}
```

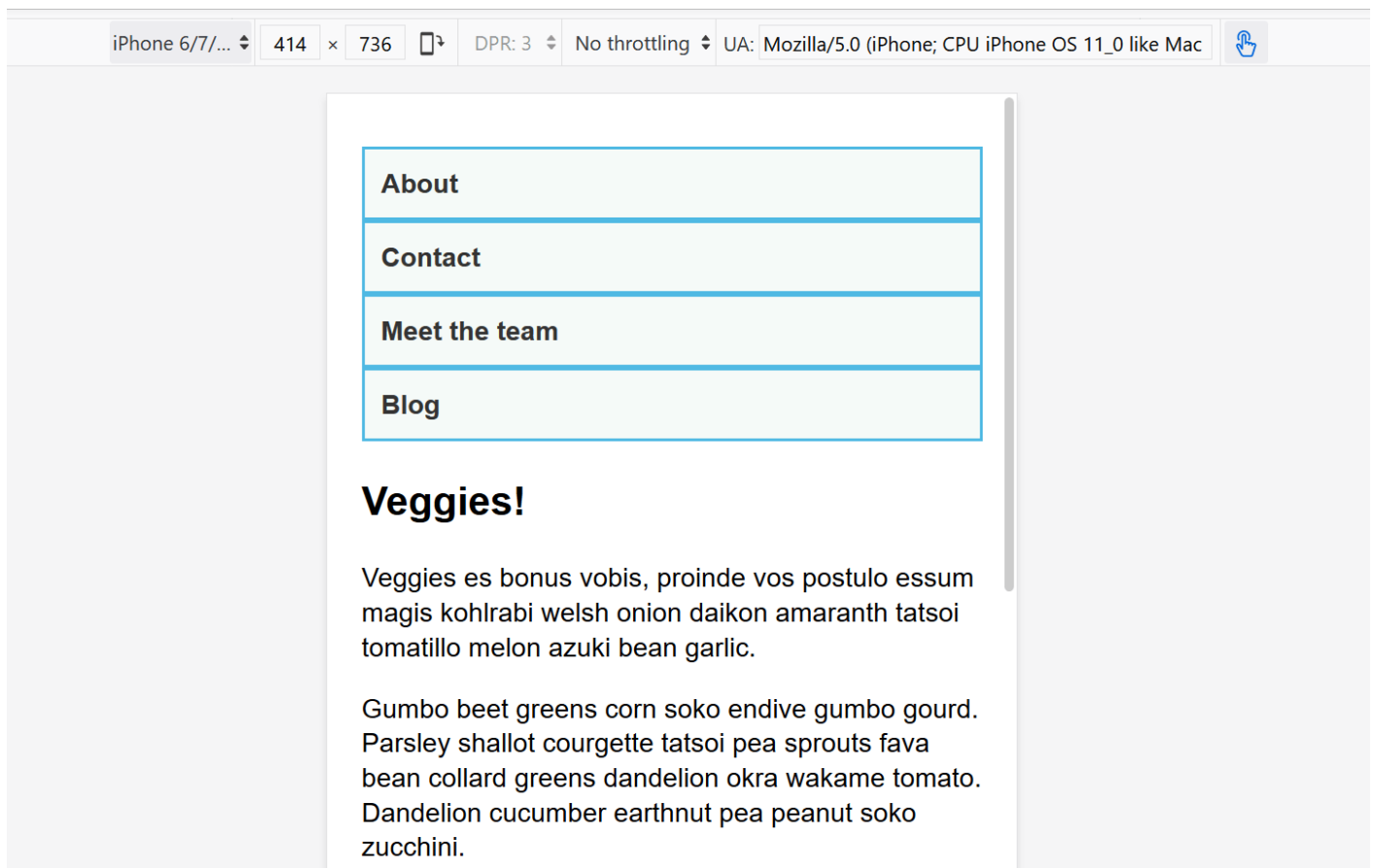
[Abra este ejemplo](#) en el navegador o [vea la fuente](#) .

Cómo elegir puntos de interrupción

En los primeros días del diseño receptivo, muchos diseñadores intentarían apuntar a tamaños de pantalla muy específicos. Se publicaron listas de los tamaños de las pantallas de los teléfonos y tabletas populares para poder crear diseños que coincidieran perfectamente con esas ventanas.

Ahora hay demasiados dispositivos, con una gran variedad de tamaños, para que eso sea factible. Esto significa que, en lugar de apuntar a tamaños específicos para todos los diseños, un mejor enfoque es cambiar el diseño al tamaño donde el contenido comienza a romperse de alguna manera. Tal vez las longitudes de las líneas se vuelvan demasiado largas, o una barra lateral enmarcada se aplaste y sea difícil de leer. Ese es el punto en el que desea utilizar una consulta de medios para cambiar el diseño a uno mejor para el espacio que tiene disponible. Este enfoque significa que no importa cuáles sean las dimensiones exactas del dispositivo que se utiliza, se atiende a todos los rangos. Los puntos en los que se introduce una consulta de medios se conocen como puntos de interrupción .

El [modo de diseño](#) receptivo en Firefox DevTools es muy útil para determinar dónde deben ir estos puntos de interrupción. Puede hacer que la ventana gráfica sea más pequeña o más grande para ver dónde se mejoraría el contenido agregando una consulta de medios y ajustando el diseño.



Aprendizaje activo: primer diseño receptivo móvil

En términos generales, puede adoptar dos enfoques para un diseño receptivo. Puede comenzar con su escritorio o la vista más amplia y luego agregar puntos de interrupción para mover las cosas a medida que la ventana gráfica se vuelve más pequeña, o puede comenzar con la vista más pequeña y agregar diseño a medida que la ventana gráfica se vuelve más grande. Este segundo enfoque se describe como el primer diseño receptivo móvil y, a menudo, es el mejor enfoque a seguir.

La vista de los dispositivos más pequeños suele ser una simple columna de contenido, tal como aparece en el flujo normal. Esto significa que probablemente no necesite hacer mucho diseño para dispositivos pequeños: ordene bien su fuente y tendrá un diseño legible por defecto.

El siguiente tutorial lo lleva a través de este enfoque con un diseño muy simple. En un sitio de producción, es probable que tenga que ajustar más cosas dentro de sus consultas de medios, sin embargo, el enfoque sería exactamente el mismo.

Tutorial: un diseño simple para dispositivos móviles primero

Nuestro punto de partida es un documento HTML con algo de CSS aplicado para agregar colores de fondo a las distintas partes del diseño.

```
* {  
  box-sizing: border-box;  
}  
  
body {
```

```

width: 90%;
margin: 2em auto;
font: 1em/1.3 Arial, Helvetica, sans-serif;
}

a:link,
a:visited {
  color: #333;
}

nav ul,
aside ul {
  list-style: none;
  padding: 0;
}

nav a:link,
nav a:visited {
  background-color: rgba(207, 232, 220, 0.2);
  border: 2px solid rgb(79, 185, 227);
  text-decoration: none;
  display: block;
  padding: 10px;
  color: #333;
  font-weight: bold;
}

nav a:hover {
  background-color: rgba(207, 232, 220, 0.7);
}

.related {
  background-color: rgba(79, 185, 227, 0.3);
  border: 1px solid rgb(79, 185, 227);
  padding: 10px;
}

.sidebar {
  background-color: rgba(207, 232, 220, 0.5);
  padding: 10px;
}

article {
  margin-bottom: 1em;
}

```

No hemos realizado cambios de diseño, sin embargo, la fuente del documento está ordenada de manera que hace que el contenido sea legible. Este es un primer paso importante y que garantiza que si un lector de pantalla leyera el contenido, sería comprensible.

```

<body>
  <div class="wrapper">
    <header>
      <nav>
        <ul>

```

```

    <li><a href="">About</a></li>
    <li><a href="">Contact</a></li>
    <li><a href="">Meet the team</a></li>
    <li><a href="">Blog</a></li>
  </ul>
</nav>
</header>
<main>
  <article>
    <div class="content">
      <h1>Veggies!</h1>
      <p>
        ...
      </p>
    </div>
    <aside class="related">
      <p>
        ...
      </p>
    </aside>
  </article>

  <aside class="sidebar">
    <h2>External vegetable-based links</h2>
    <ul>
      <li>
        ...
      </li>
    </ul>
  </aside>
</main>

<footer><p>&copy;2019</p></footer>
</div>
</body>

```

Este diseño simple también funciona bien en dispositivos móviles. Si vemos el diseño en modo de diseño receptivo en DevTools, podemos ver que funciona bastante bien como una vista móvil directa del sitio.

Abra el paso 1 en el navegador o vea la fuente .

Si desea seguir e implementar este ejemplo a medida que avanzamos, haga una copia local de step1.html en su computadora.

A partir de este punto, comience a arrastrar la vista del Modo de diseño receptivo hasta que vea que las longitudes de las líneas son cada vez más largas y tenemos espacio para que la navegación se muestre en una línea horizontal. Aquí es donde agregaremos nuestra primera consulta de medios. Usaremos ems, ya que esto significará que si el usuario ha aumentado el tamaño de su texto, el punto de interrupción se producirá en una línea de longitud similar pero con una ventana gráfica más amplia que alguien con un tamaño de texto más pequeño.

Agregue el siguiente código en la parte inferior de su step1.html CSS.

```

@media screen and (min-width: 40em) {
  article {

```



```

display: grid;
grid-template-columns: 3fr 1fr;
column-gap: 20px;
}

nav ul {
display: flex;
}

nav li {
flex: 1;
}
}

```

Este CSS nos brinda un diseño de dos columnas dentro del artículo, del contenido del artículo y la información relacionada en el elemento aparte. También hemos usado flexbox para poner la navegación en una fila.

[Abra el paso 2](#) en el navegador o [vea la fuente](#) .

Continuemos expandiendo el ancho hasta que sintamos que hay suficiente espacio para que la barra lateral también forme una nueva columna. Dentro de una consulta de medios, convertiremos el elemento principal en una cuadrícula de dos columnas. Luego, debemos eliminar el [margin-bottom](#) en el artículo para que las dos barras laterales se alineen entre sí, y agregaremos una [border](#) en la parte superior del pie de página. Por lo general, estos pequeños ajustes son el tipo de cosas que hará para que el diseño se vea bien en cada punto de interrupción.

Nuevamente, agregue el siguiente código en la parte inferior de su step1.html CSS.

```

@media screen and (min-width: 70em) {
main {
display: grid;
grid-template-columns: 3fr 1fr;
column-gap: 20px;
}

article {
margin-bottom: 0;
}

footer {
border-top: 1px solid #ccc;
margin-top: 2em;
}
}

```

[Abra el paso 3](#) en el navegador o [vea la fuente](#) .

Si observa el ejemplo final en diferentes anchos, puede ver cómo responde el diseño y funciona como una sola columna, dos columnas o tres columnas, según el ancho disponible. Este es un ejemplo muy simple de un primer diseño receptivo móvil.

La metaetiqueta de la ventana gráfica

Si observa la fuente HTML en el ejemplo anterior, verá el siguiente elemento incluido en el encabezado del documento:

```
<meta name="viewport" content="width=device-width,initial-scale=1">
```

Esta es la metaetiqueta de la ventana gráfica : existe como una forma de controlar cómo los navegadores móviles representan el contenido. Esto es necesario porque, de forma predeterminada, la mayoría de los navegadores móviles mienten sobre el ancho de su ventana gráfica. Los sitios que no responden generalmente se ven muy mal cuando se representan en una ventana de visualización estrecha, por lo que los navegadores móviles generalmente representan el sitio con un ancho de ventana de visualización más ancho que el ancho del dispositivo real de forma predeterminada (generalmente 980 píxeles) y luego reducen el resultado renderizado para que se ajuste en la pantalla.

Todo esto está muy bien, pero significa que los sitios receptivos no funcionarán como se esperaba. Si el ancho de la ventana gráfica se informa como 980 píxeles, los diseños móviles (por ejemplo, creados con una consulta de medios de `@media screen and (max-width: 600px) { ... }`) no se representarán como se esperaba.

Para remediar esto, incluya una metaetiqueta de ventana gráfica como la que se muestra arriba en su página que le dice al navegador "no renderice el contenido con una ventana gráfica de 980 píxeles; renderícelo usando el ancho real del dispositivo y establezca un nivel de escala inicial predeterminado para mejorar". consistencia." Las consultas de los medios se activarán como se esperaba.

Hay una serie de otras opciones que puede colocar dentro del `content` atributo de la metaetiqueta de la ventana gráfica; consulte [Uso de la metaetiqueta de la ventana gráfica para controlar el diseño en los navegadores móviles](#) para obtener más detalles.

¿Realmente necesitas una consulta de medios?

Flexbox, Grid y el diseño de varias columnas le brindan formas de crear componentes flexibles e incluso receptivos sin la necesidad de una consulta de medios. Siempre vale la pena considerar si estos métodos de diseño pueden lograr lo que desea sin agregar consultas de medios. Por ejemplo, es posible que desee un conjunto de tarjetas que tengan al menos 200 píxeles de ancho, con tantos de estos 200 píxeles como quepan en el artículo principal. Esto se puede lograr con el diseño de cuadrícula, sin utilizar consultas de medios en absoluto.

Esto podría lograrse usando lo siguiente:

```
<ul class="grid">
  <li>
    <h2>Card 1</h2>
    <p>...</p>
  </li>
  <li>
    <h2>Card 2</h2>
    <p>...</p>
  </li>
```

```
<li>
  <h2>Card 3</h2>
  <p>...</p>
</li>
<li>
  <h2>Card 4</h2>
  <p>...</p>
</li>
<li>
  <h2>Card 5</h2>
  <p>...</p>
</li>
</ul>
```

```
.grid {
  list-style: none;
  margin: 0;
  padding: 0;
  display: grid;
  gap: 20px;
  grid-template-columns: repeat(auto-fill, minmax(200px, 1fr));
}

.grid li {
  border: 1px solid #666;
  padding: 10px;
}
```

[Abra el ejemplo de diseño de cuadrícula](#) en el navegador o [vea el código fuente](#) .

Con el ejemplo abierto en su navegador, haga que la pantalla sea más ancha y más estrecha para ver cómo cambia el número de pistas de columna. Lo bueno de este método es que la cuadrícula no mira el ancho de la ventana gráfica, sino el ancho que tiene disponible para este componente. ¡Puede parecer extraño terminar una sección sobre consultas de medios con la sugerencia de que es posible que no necesite ninguna! Sin embargo, en la práctica encontrará que un buen uso de los métodos de diseño modernos, mejorados con consultas de medios, dará los mejores resultados.

¡Pon a prueba tus habilidades!

Has llegado al final de este artículo, pero ¿puedes recordar la información más importante? Puede encontrar una prueba para verificar que ha retenido esta información antes de continuar; consulte [Ponga a prueba sus habilidades: diseño web receptivo](#) .

Resumen

En esta lección, aprendió sobre las consultas de medios y también descubrió cómo usarlas en la práctica para crear un primer diseño receptivo móvil.

Puede usar el punto de partida que hemos creado para probar más consultas de medios. Por ejemplo, tal vez podría cambiar el tamaño de la navegación si detecta que el visitante tiene un puntero grueso, utilizando la `pointer` función de medios.

También puede experimentar agregando diferentes componentes y ver si la adición de una consulta de medios o el uso de un método de diseño como flexbox o grid es la forma más adecuada de hacer que los componentes respondan. Muy a menudo, no hay una manera correcta o incorrecta: debe experimentar y ver cuál funciona mejor para su diseño y contenido.