

The background is a dark blue-grey color with a subtle pattern of small, light-colored dots arranged in a grid. Overlaid on this is a large, dark blue circle with a thick white border. Inside this circle, there are several white molecular structures, which are clusters of small circles connected by lines, resembling chemical or network diagrams. The text "DISEÑO DE INTERFACES" is centered within the circle in a bold, white, sans-serif font.

DISEÑO DE INTERFACES

Las etiquetas `<video>` y `<source>`

Usted [preparó correctamente un archivo de video](#) para la web. Le dio las dimensiones y la resolución correctas. Incluso creó archivos WebM y MP4 separados para diferentes navegadores

Para que cualquiera pueda ver su video, aún debe agregarlo a una página web. Para hacerlo correctamente, es necesario agregar dos elementos HTML: el elemento `<video>` y el elemento `<source>`. Además de los conceptos básicos sobre estas etiquetas, este artículo explica los atributos que debe agregar a esas etiquetas para crear una buena experiencia de usuario.

Siempre tiene la opción de subir su archivo a [YouTube](#) o [Vimeo](#). En muchos casos, esto es preferible que el procedimiento descrito aquí. Estos servicios se encargan de convertir el formato y el tipo de archivo por usted, además de proporcionar los medios para incrustar un video en su página web. Si necesita administrar esto usted mismo, siga leyendo.

Cómo especificar un solo archivo <#>

Aunque no se recomienda, puede utilizar el elemento video por sí mismo. Utilice siempre el atributo `type` como se muestra a continuación. El navegador lo utiliza para determinar si puede reproducir el archivo de video proporcionado. Si no puede, se muestra el texto adjunto.

```
<video src="chrome.webm" type="video/webm">
  <p>Your browser cannot play the provided video file.</p>
</video>
```

Cómo especificar varios formatos de archivos <#>

De acuerdo con los [fundamentos de los archivos multimedia](#), no todos los navegadores son compatibles con los mismos formatos de video. El elemento `<source>` permite especificar varios formatos como alternativa en caso de que el navegador del usuario no admita uno de ellos.

El siguiente ejemplo produce el video incrustado que se utiliza como ejemplo más adelante en este artículo.

```
<video controls>
  <source src="https://storage.googleapis.com/web-dev-assets/video-and-source-tags/chrome.webm" type="video/webm">
  <source src="https://storage.googleapis.com/web-dev-assets/video-and-source-tags/chrome.mp4" type="video/mp4">
  <p>Your browser cannot play the provided video file.</p>
</video>
```

[Pruébalo en Glitch](#) ([fuente](#))

Observe que en el ejemplo anterior se introdujo el atributo `controls`. Esto indica que los navegadores deben permitir al usuario controlar la reproducción de video, incluyendo el volumen, búsqueda, selección de subtítulos y pausa/reanudación de la reproducción entre otras funciones.

Siempre debe agregar un atributo `type` a las etiquetas `<source>` aunque es opcional. Esto asegura que el navegador solo puede descargar el archivo que es capaz de reproducir.

Este enfoque tiene varias ventajas si se compara con publicar diferentes HTML o scripts del lado del servidor, especialmente en los dispositivos móviles:

- Puede hacer una lista de los formatos por orden de preferencia.
- El cambio del lado del cliente reduce la latencia, ya que solo se realiza una solicitud para obtener el contenido.
- Permitir que el navegador elija un formato es más sencillo, rápido y potencialmente más confiable que utilizar una base de datos de respaldo del lado del servidor con detección como agente de usuario.
- Especificar el tipo de fuente de cada archivo mejora el rendimiento de la red, ya que el navegador puede seleccionar una fuente de video sin tener que descargar parte del video para "olfatear" el formato.

Estos problemas son especialmente importantes en contextos móviles, donde el ancho de banda y la latencia son muy importantes, y la paciencia del usuario es probablemente limitada. Omitir el atributo `type` puede afectar al rendimiento cuando hay varias fuentes con tipos no admitidos.

Hay algunas formas de profundizar en los detalles. Consulte [Un manual de medios digitales para geeks](#) para conocer más sobre cómo funcionan el video y el audio en la web. También puede utilizar [la depuración remota](#) en DevTools para comparar la actividad de la red [con atributos de tipo](#) y [sin atributos de tipo](#).

Precaución

Asegúrese de verificar los encabezados de respuesta en las herramientas de desarrollo de su navegador para [asegurarse de que su servidor informa del tipo MIME correcto](#), de lo contrario las verificaciones de tipo de fuente de video no funcionarán.

Especifique las horas de inicio y finalización

Ahorre ancho de banda y haga que su sitio tenga una mayor capacidad de respuesta: utilice fragmentos de medios para agregar tiempos de inicio y finalización al elemento de video.

Para utilizar un fragmento multimedia, agregue `#t=[start_time][,end_time]` a la URL multimedia. Por ejemplo, para reproducir el video de los segundos 5 a 10, especifique:

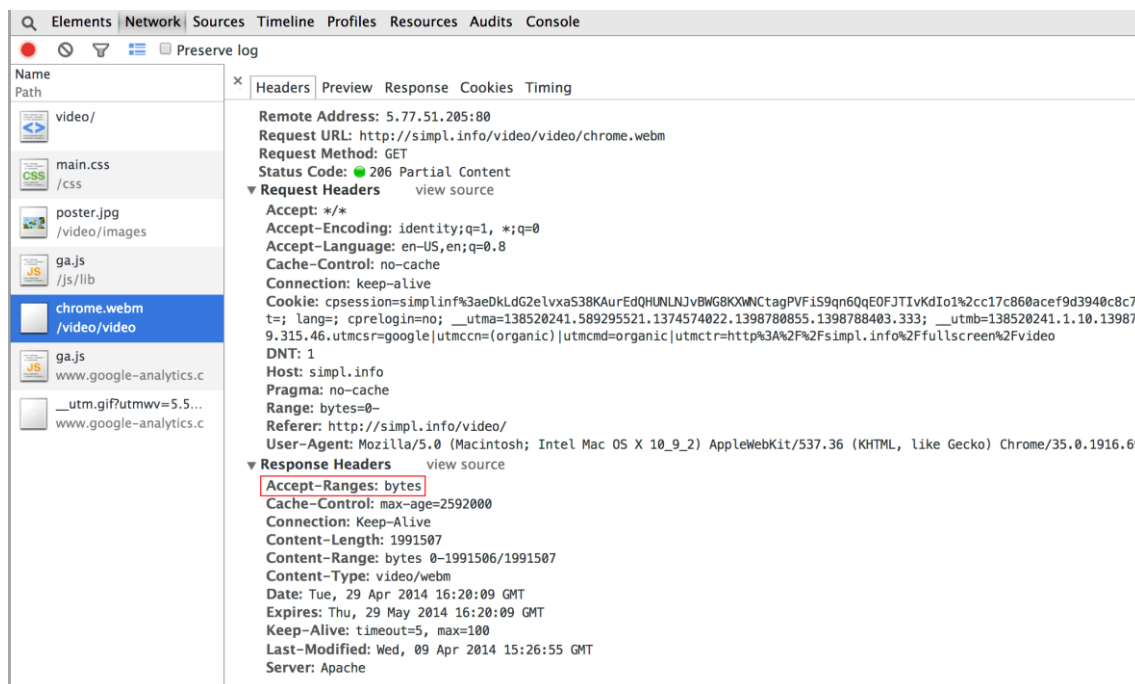
```
<source src="video/chrome.webm#t=5,10" type="video/webm">
```

También puede especificar los tiempos en `<hours>:<minutes>:<seconds>`. Por ejemplo, `#t=00:01:05` inicia el video en un minuto y cinco segundos. Para reproducir solamente el primer minuto de video, especifique `#t=,00:01:00`.

Puede utilizar esta función para ofrecer varias vistas en el mismo video, como puntos de referencia en un DVD, sin tener que codificar y distribuir varios archivos.

Para que esta función funcione, su servidor debe admitir solicitudes de rango y esa capacidad se debe habilitar. La mayoría de los servidores habilitan las solicitudes de rango de forma predeterminada. Dado que algunos servicios de alojamiento las desactivan, debe confirmar que las solicitudes de rango están disponibles para utilizar fragmentos en su sitio.

Afortunadamente, puede hacer esto desde las herramientas de desarrollo de su navegador. En Chrome, por ejemplo, se encuentra en el [panel de la red](#). Busque el encabezado `Accept-Ranges` y verifique que dice `bytes`. En la imagen, dibujé un cuadro rojo alrededor de este encabezado. Si no observa `bytes` como valor, tendrá que ponerse en contacto con su proveedor de alojamiento.



Captura de pantalla de Chrome DevTools: `Accept-Ranges: bytes`.

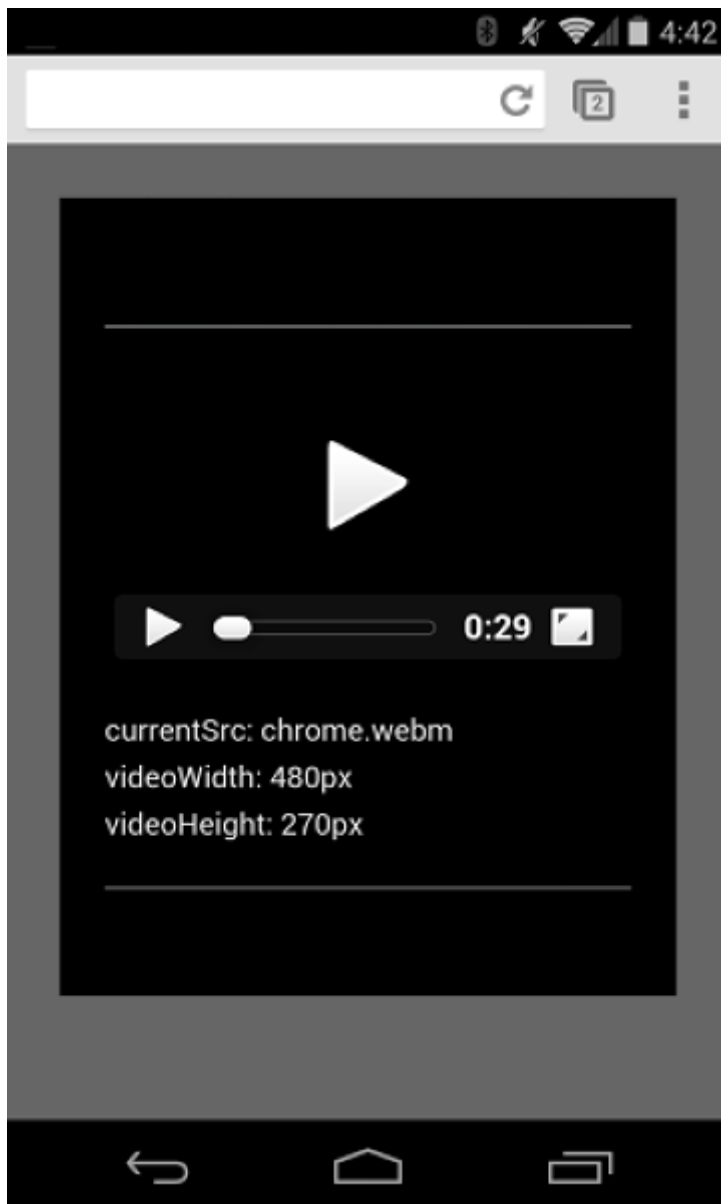
Incluya una imagen de cartel

Agregue un atributo de cartel al elemento `video` para que los espectadores se hagan una idea del contenido en cuanto se cargue el elemento, sin necesidad de descargar el video o iniciar la reproducción.

```
<video poster="poster.jpg" ...>
...
</video>
```

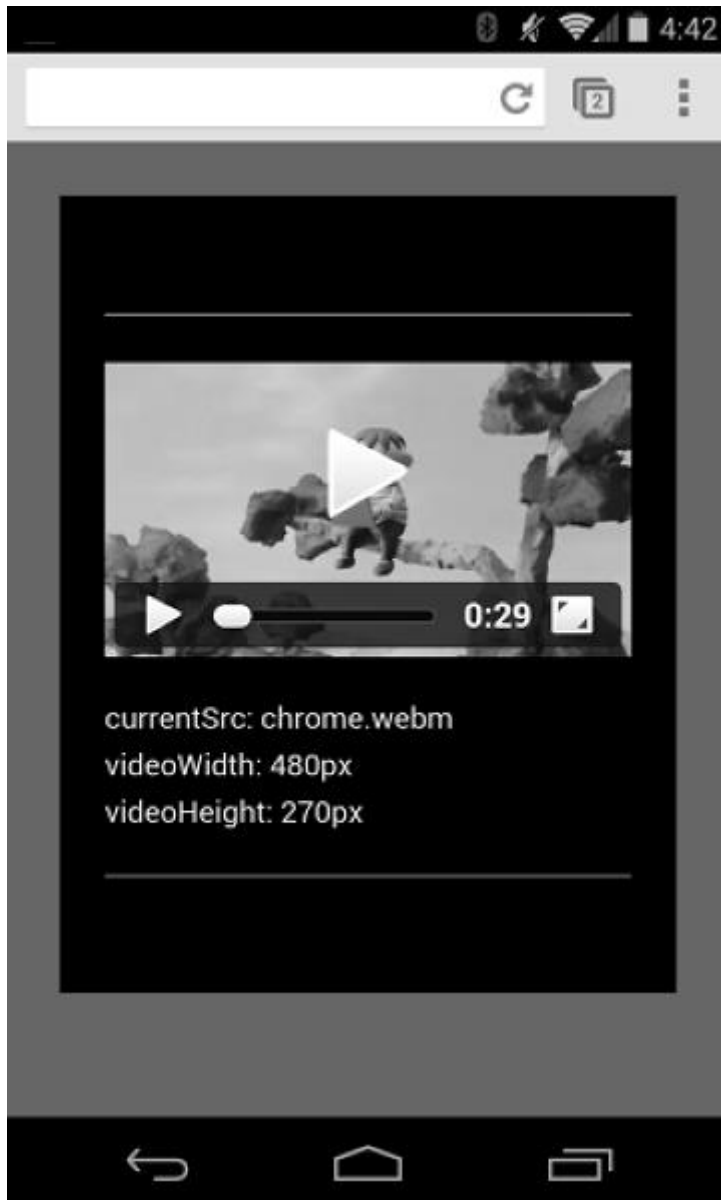
Un cartel también puede ser una alternativa si el video `src` está dañado o si ninguno de los formatos de video proporcionados es compatible. La única desventaja de las imágenes de un cartel es el hecho de solicitar un archivo adicional, que consuma algo de ancho de banda y que requiera renderización. Para obtener más información, consulte [Codificación eficiente de las imágenes](#).

No hagas



Sin un cartel de reserva, el video parece estropeado.

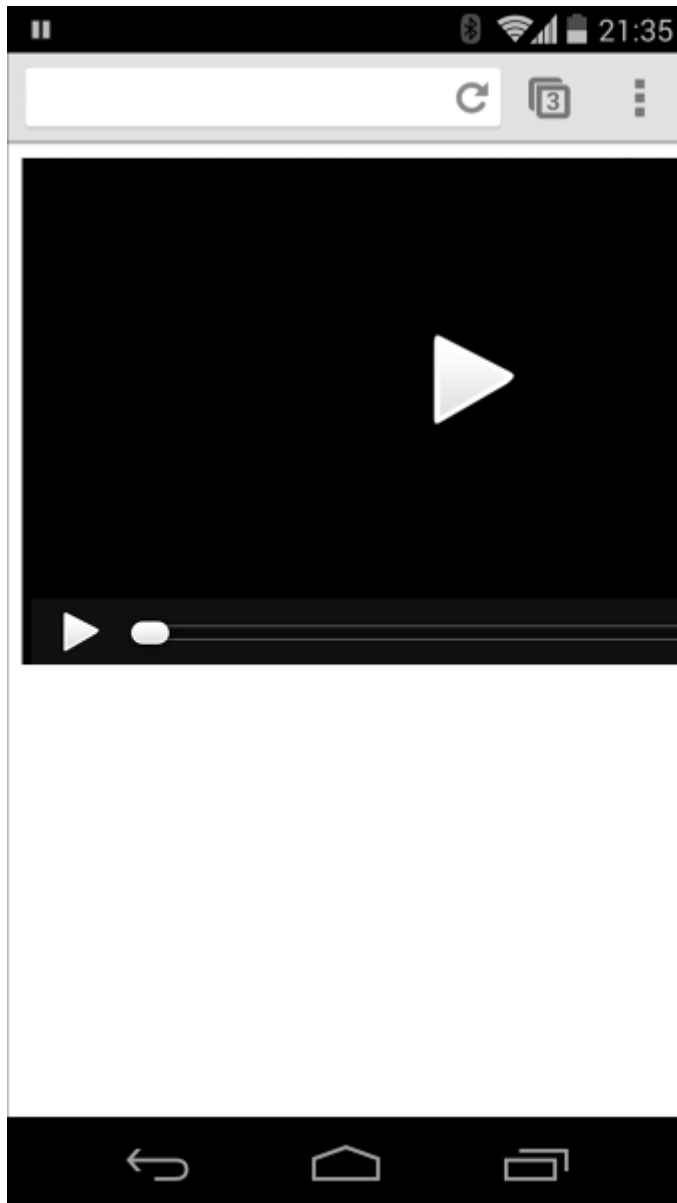
Hacer



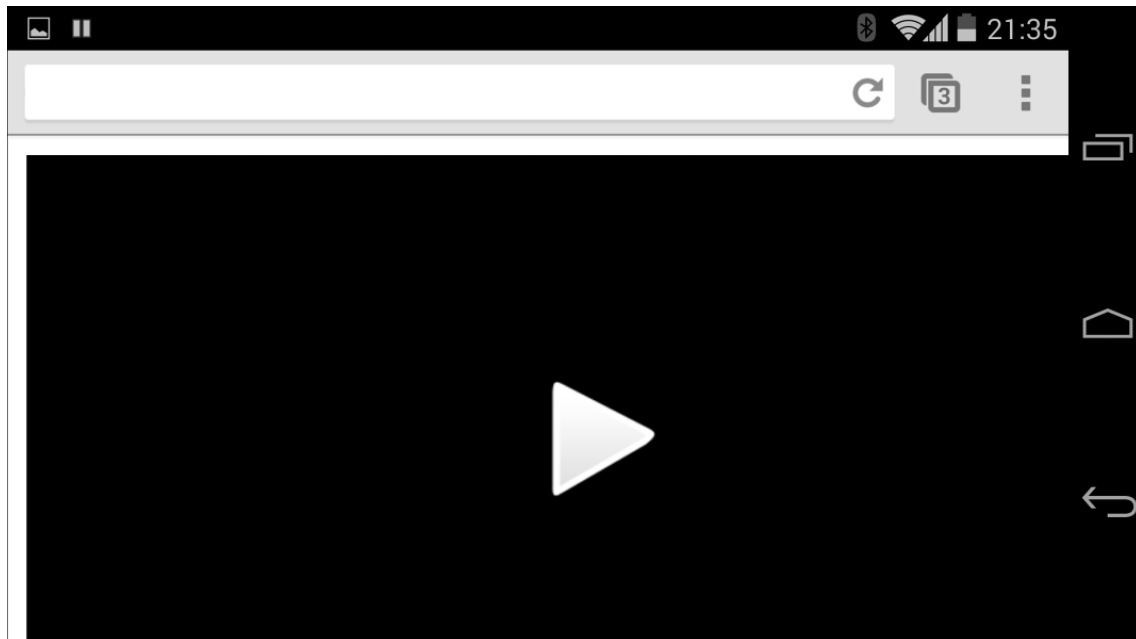
Un cartel alternativo hace que parezca que se capturó el primer fotograma.

Asegúrese de que los videos no desborden los contenedores <#>

Cuando los elementos de video son demasiado grandes para la ventana de visualización, pueden desbordar su contenedor, lo que hace imposible que el usuario vea el contenido o utilice los controles.



Captura de pantalla de Android Chrome, en vertical: el elemento de video sin estilo desborda la ventana de visualización.



Captura de pantalla de Android Chrome, paisaje: el elemento de video sin estilo desborda la ventana de visualización.

Puede controlar las dimensiones del video usando CSS. Si el CSS no satisface todas sus necesidades, las bibliotecas de JavaScript y los complementos como [FitVids](#) (que están fuera del alcance de este artículo) pueden ayudar, incluso para los videos de YouTube y otras fuentes. Lamentablemente, estos recursos pueden aumentar el [tamaño de la carga útil de su red](#), con consecuencias negativas para sus ingresos y la cartera de sus usuarios.

Para usos sencillos como los que describo aquí, utilice las [consultas de medios de CSS](#) para especificar el tamaño de los elementos según las dimensiones de la ventana de visualización; `max-width: 100%` será su aliado.

Para el contenido multimedia en iframes (como los videos de YouTube), pruebe un enfoque adaptativo (como el que [propuso John Surdakowski](#)).

Precaución

No fuerce el tamaño de los elementos para que la relación de aspecto sea diferente a la del video original. Los videos comprimidos o estirados se ven horribles.

CSS

```
.video-container {
  position: relative;
  padding-bottom: 56.25%;
  padding-top: 0;
  height: 0;
  overflow: hidden;
}
```




```
.video-container iframe,  
.video-container object,  
.video-container embed {  
    position: absolute;  
    top: 0;  
    left: 0;  
    width: 100%;  
    height: 100%;  
}
```

HTML

```
<div class="video-container">  
    <iframe src="//www.youtube.com/embed/l-BA9Ee2XuM"  
        frameborder="0" width="560" height="315">  
    </iframe>  
</div>
```

Pruébalo

Compare la [muestra adaptativa](#) con la [versión no adaptativa](#). Como podrá ver, la versión sin respuesta no es una gran experiencia para el usuario.

Orientación del dispositivo

La orientación del dispositivo no es un problema para los monitores de las computadoras de escritorio o laptops, pero es muy importante cuando se considera el diseño de la página web para dispositivos móviles y tablets.

Safari en iPhone hace un buen trabajo al cambiar entre la orientación vertical y la horizontal:

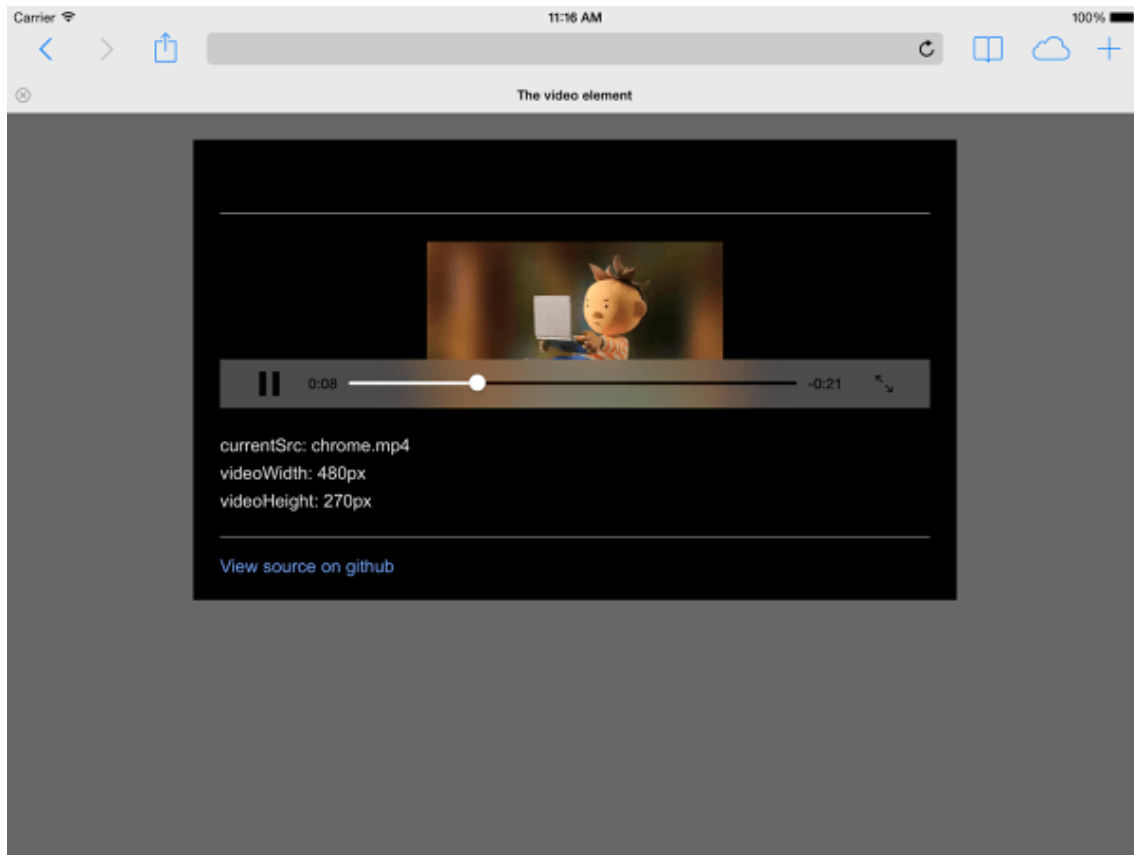


Captura de pantalla de un video que se reproduce en un iPhone con Safari, en posición vertical.



Captura de pantalla de un video que se reproduce en un iPhone con Safari, en formato horizontal.

La orientación del dispositivo en un iPad y Chrome en Android podría ser problemática. Por ejemplo, sin ninguna personalización, un video que se reproduzca en un iPad con orientación horizontal tendrá el siguiente aspecto:



Captura de pantalla de un video que se reproduce en un iPad con Safari, en formato horizontal.

La configuración del video `width: 100% O max-width: 100%` con CSS puede resolver muchos problemas de diseño en la orientación del dispositivo.

Reproducción automática

El atributo `autoplay` controla si el navegador descarga y reproduce un video inmediatamente. El funcionamiento exacto depende de la plataforma y del navegador.

- Chrome: Depende de varios factores, incluyendo entre otros, si la visualización se realiza en una computadora de escritorio y si el usuario de un dispositivo móvil agregó su sitio o aplicación a su pantalla de inicio. Para obtener más información, consulte [Prácticas recomendadas de la reproducción automática](#).
- Firefox: Bloquea todo el video y el sonido, pero ofrece a los usuarios la posibilidad de relajar estas restricciones para todos los sitios o para sitios específicos. Para obtener más información, consulte [Permitir o bloquear la reproducción automática de medios en Firefox](#).
- Safari: Históricamente requiere un gesto del usuario, pero en las versiones recientes se ha relajado ese requisito. Para más información, consulte [Nuevas políticas de <video> para iOS](#).

Incluso en las plataformas en las que es posible la reproducción automática, debe considerar si es una buena idea activarla:

- El uso de datos puede ser caro.
- Reproducir medios antes de que el usuario lo desee puede acaparar el ancho de banda y el CPU y, por tanto, retrasar la renderización de la página.
- Los usuarios pueden estar en un contexto en el que la reproducción de video o de audio es intrusiva.

Precarga

El atributo `preload` proporciona una pista al navegador sobre cuánta información o contenido debe precargar.

Valor	Descripción
<code>none</code>	El usuario podría elegir no ver el video, así que no precargará nada.
<code>metadata</code>	Los metadatos (duración, dimensiones, pistas de texto) deberían estar precargados, pero con un video con un contenido mínimo.
<code>auto</code>	Se considera deseable descargar el video completo de inmediato. Una cadena vacía produce el mismo resultado.

El atributo `preload` tiene efectos diferentes en las distintas plataformas. Por ejemplo, Chrome almacena 25 segundos de video en el escritorio, pero ninguno en iOS o Android. Esto significa que en los dispositivos móviles puede haber retrasos en el inicio de la reproducción que no se producen en el escritorio. Consulte [Reproducción rápida con precarga de audio y video](#) o [el blog de Steve Souders](#) para obtener más información.

Ahora que ya sabe cómo agregar medios de comunicación a su página web es el momento de aprender acerca de la [Accesibilidad de los medios de comunicación](#) donde agregará subtítulos a su video para las personas con discapacidad auditiva, o cuando la reproducción del audio no es una opción viable.



