

El nuevo elemento <picture> de HTML5 para crear imágenes responsive

El nuevo elemento <picture> de HTML5 permite describir con todo detalle cómo deben cargarse las imágenes de tu sitio web. Ya no serán necesarios los *hacks* de CSS o JavaScript para gestionar las imágenes responsive de los diseños web. Además, los usuarios se aprovecharán de las ventajas de cargar solamente las imágenes optimizadas para el dispositivo que están utilizando, lo que es especialmente útil para usuarios con móviles y conexiones lentas a Internet.

Al margen de los nuevos atributos `srcset` y `sizes` definidos recientemente para los elementos , el nuevo elemento <picture> permite una mayor flexibilidad al especificar qué imágenes utiliza el sitio. Gracias a este elemento <picture>, será posible escribir código HTML *limpio* y semántico, dejando que el navegador haga todo el trabajo de seleccionar la mejor imagen para cada situación.

La elección del mejor archivo de imagen depende de muchos factores:

Elección basada en el diseño gráfico

¿El dispositivo es un móvil en vertical o es un monitor panorámico?
Carga la mejor imagen optimizada para el tamaño de la pantalla.

Elección basada en la densidad de píxeles

¿Se trata de un dispositivo de alta resolución? Carga las imágenes de alta resolución.

Elección basada en cómo se visualizará la imagen

¿La imagen debe ocupar siempre un tamaño determinado de la ventana del navegador? Carga las imágenes en función del tamaño de la ventana del navegador.

Elección basada en el formato de la imagen

¿Soporta el navegador formatos de imagen con mucha mayor compresión que los tradicionales? Carga un formato de imagen alternativo, como por ejemplo WebP.

Seleccionando la imagen en función de criterios artísticos

El uso más habitual del elemento <picture> consiste en elegir la mejor imagen exclusivamente en función de criterios artísticos. En vez de diseñar una única imagen que se escala para ajustarse al tamaño de la ventana del

navegador, se pueden diseñar diferentes imágenes en función de su tamaño.



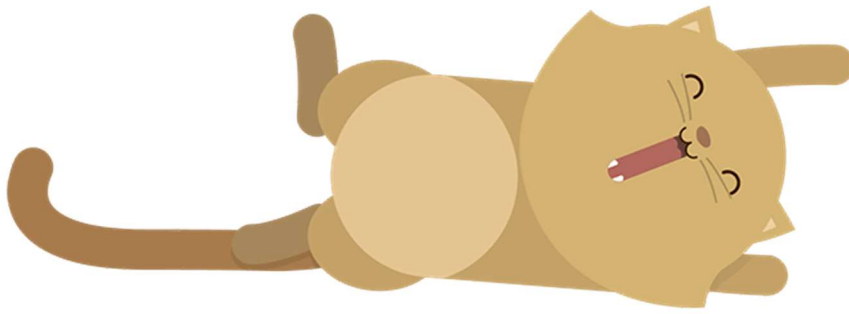
Izquierda: la misma imagen se escala para todos los tamaños de ventana. **Derecha:** diferentes imágenes en función del tamaño de la ventana del navegador.

Mejorando el rendimiento al cargar las imágenes

Cuando se utiliza el nuevo elemento `<picture>` o el elemento `` con los atributos `srcset` y `sizes`, el navegador solamente descarga la imagen adecuada para el navegador y las condiciones de acceso del usuario (tamaño del navegador, densidad en píxeles de la pantalla, formatos soportados por el navegador, etc.) La ventaja de que este comportamiento sea nativo del navegador es que se pueden aprovechar todas las funcionalidades de los navegadores, como la caché de contenidos y la precarga de imágenes.

El elemento en acción

Como sabes, Internet se inventó para mostrar fotos de gatitos, así que vamos a utilizar el elemento `<picture>` en acción mostrando cómo se ajusta nuestro gato al espacio disponible en el navegador.



[Abrir la demo en una nueva pestaña del navegador.](#) Para ver el elemento `<picture>` en acción, asegúrate de utilizar el navegador Chrome 38 y redimensiona la ventana del navegador para observar cómo cambia la imagen mostrada.

Esta demo es muy básica porque es una primera toma de contacto con las posibilidades del nuevo elemento `<picture>`. Sigue leyendo para conocer todas sus posibilidades.

La sintaxis del elemento `<picture>`

El siguiente código HTML y CSS muestra todo lo necesario para crear la anterior demo:

```
<style>
  img {display: block; margin: 0 auto;}
</style>
<picture>
  <source media="(min-width: 650px)" srcset="images/kitten-stretching.png">
  <source media="(min-width: 465px)" srcset="images/kitten-sitting.png">
  
</picture>
```

Como puedes ver, no se utiliza ni código JavaScript ni ninguna otra librería externa. El bloque de código CSS se utiliza para aplicar unos estilos básicos a la imagen y de nuevo puedes ver que no se utilizan *hacks* ni *media queries*. Cuando el navegador soporta el elemento `<picture>`, tu único trabajo consiste en definir todas las imágenes *responsive* que tienes disponibles y es el navegador el que se encarga de seleccionar la mejor alternativa.

Uso de `<picture>` con los elementos `<source>`

El elemento `<picture>` no define ningún atributo propio, pero puedes conseguir comportamientos muy avanzados cuando utilizas elementos `<picture>` para encerrar a varios `<source>`.

El elemento `<source>`, que se utiliza para cargar elementos multimedia como audios y vídeos, se ha actualizado para que también soporte la carga de imágenes. Para ello, se le han añadido los siguientes atributos:

Atributo `srcset` (obligatorio)

Indica la ruta de la imagen a la que se hace referencia (ejemplo `srcset="kitten.png"`).

También se puede indicar una lista de rutas separadas por comas y que incluyan el sufijo que indica la densidad de píxeles (ejemplo `srcset="kitten.png, kitten@2X.png 2x"`). Para la densidad de píxeles normales de 1 no hace falta añadir el descriptor 1x.

Atributo `media` (opcional)

Permite indicar cualquier *media query* que sea válido en el selector `@media` de CSS (ejemplo `media="(max-width: 30em)"`).

Atributo `sizes` (opcional)

Acepta cualquier valor que describa la anchura de la imagen (ejemplo `sizes="100vw"`) o un *media query* que defina la anchura de la imagen (ejemplo `sizes="(max-width: 30em) 100vw"`).

También se puede indicar una lista de *media queries* separadas por comas y que describan varias anchuras de la imagen (ejemplo `sizes="(max-width: 30em) 100vw, (max-width: 50em) 50vw, calc(33vw - 100px)"`). En este caso se utiliza por defecto el último de los valores definidos.

Atributo `type` (opcional)

Acepta como valor cualquier tipo MIME estándar (ejemplo `type="image/webp"` o `type="image/vnd.ms-photo"`).

Lee la sección [Seleccionando diferentes formatos de imagen](#) para saber cómo utilizarlo en la práctica.

El navegador utiliza el valor de todos los atributos anteriores para determinar qué imagen se debe cargar de entre todas las variantes definidas. Ten en cuenta que **el orden de las etiquetas es muy importante**, ya que el navegador siempre utilizará el primer elemento `<source>` cuyas condiciones cumpla el navegador e ignorará el resto de elementos `<source>`.

Añade un elemento `` al final

El elemento `` también se ha actualizado para poder utilizarlo dentro del elemento `<picture>` a modo de salvaguarda en el caso de que

el navegador no soporte <picture> o ninguna de las condiciones de los elementos <source> se cumplan.

Añadir un elemento dentro del elemento <picture> es **obligatorio**. Si no lo haces, **el navegador no mostrará ninguna imagen**.

La imagen definida por el elemento será la que utilizará el elemento <picture> por defecto cuando no se puede mostrar ninguna otra de las imágenes definidas. Coloca el elemento como último elemento hijo de <picture>, ya que los navegadores ignoran cualquier elemento <source> que se encuentre después de la etiqueta . Si defines un texto alternativo para la imagen mediante el atributo alt, asegúrate de añadir ese atributo en la etiqueta , no en <source> o <picture>.

Seleccionando la imagen en función de la densidad de píxeles

Utiliza los descriptores 1x, 1.5x, 2x y 3x para añadir soporte para pantallas de alta densidad de píxeles, como por ejemplo las de los *smartphones*. El atributo srcset que permite indicar estos descriptores ahora se soporta tanto en el elemento como en los elementos <source>.

El siguiente ejemplo muestra cómo soportar las pantallas de tipo 1x, 1.5x y 2.x:

```
<picture>
  <source media="(min-width: 650px)" srcset="images/kitten-stretching.png, images/kitten-
stretching@1.5x.png 1.5x, images/kitten-stretching@2x.png 2x">
  <source media="(min-width: 465px)" srcset="images/kitten-sitting.png, images/kitten-
sitting@1.5x.png 1.5x images/kitten-sitting@2x.png 2x">
  
</picture>
```

Seleccionando la imagen en función de su anchura

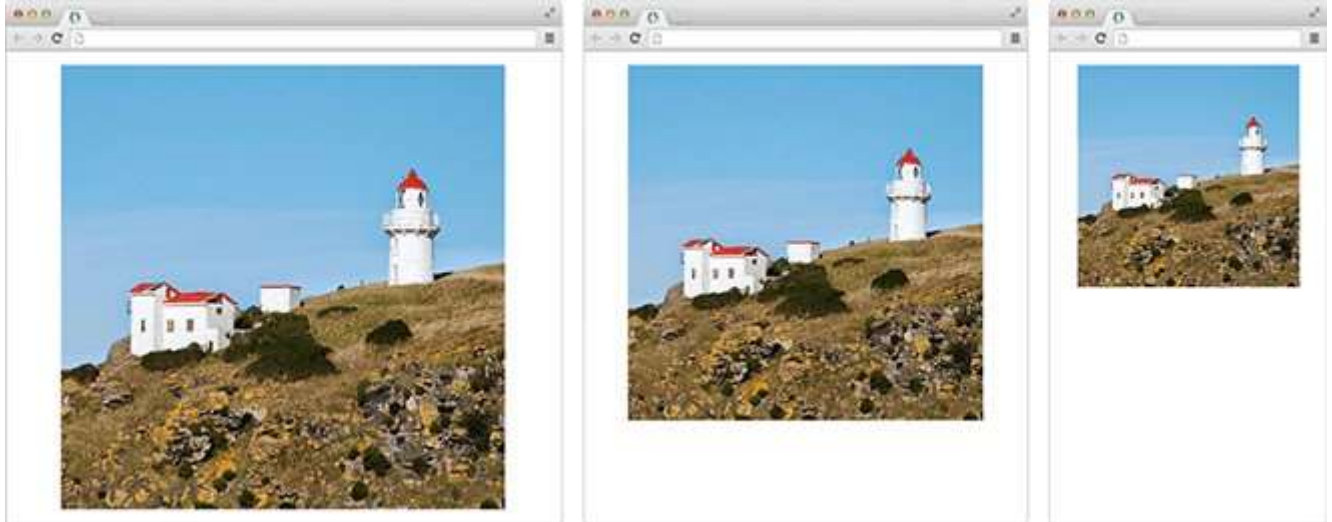
Cuando se desconoce el tamaño definitivo de una imagen, no es posible indicar el descriptor relacionado con la densidad de píxeles mencionado en la sección anterior. Así que, en vez de definir imágenes de anchura fija, se puede añadir un descriptor de su anchura para que el navegador calcule automáticamente la densidad de píxeles y así descargue la mejor imagen en cada caso.

En este ejemplo se utiliza el atributo sizes para definir que la imagen siempre ocupe el 80% de la anchura de la ventana del navegador. Además, se combina con el atributo srcset para definir cuatro versiones diferentes de la misma foto de un faro, cada una con una anchura específica: 160px, 320px, 640px y 1280px:

```

```

El navegador utiliza este descriptor de la anchura para elegir la mejor imagen en función de la anchura del navegador y de la resolución de la pantalla:



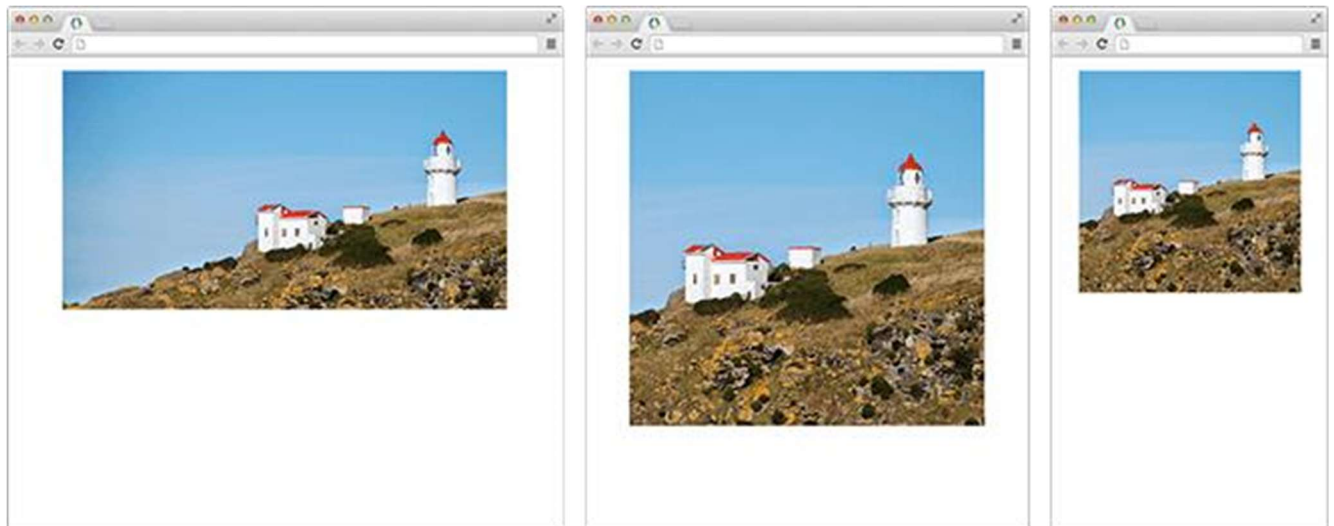
En este ejemplo, la ventana de la izquierda tiene aproximadamente 800px de ancho, por lo que el navegador carga la imagen lighthouse-640.jpg. No obstante, si el dispositivo tiene una densidad de píxeles de 2x, entonces se carga la imagen lighthouse-1280.jpg.

Al añadir `<picture>`, el atributo `sizes` se puede utilizar tanto en el elemento `` como en los elementos `<source>`:

```
<picture>
  <source media="(min-width: 800px)"
    sizes="80vw"
    srcset="lighthouse-landscape-640.jpg 640w, lighthouse-landscape-1280.jpg 1280w,
      lighthouse-landscape-2560.jpg 2560w">
  
</picture>
```

Siguiendo con este mismo ejemplo, cuando el navegador tiene una anchura de 800px o superior, se carga la imagen panorámica del faro:

es una buena alternativa, diferente de la propiedad **opacity** ya que sólo afecta a elementos individuales.



La anchura del navegador de la izquierda es mayor que 800px, por lo que se muestra la versión panorámica de la imagen.

Seleccionando diferentes formatos de imagen

El atributo `type` del elemento `<source>` se puede utilizar para cargar formatos de imagen alternativos que pueden no estar soportados por el navegador del usuario. Si por ejemplo quieres servir imágenes en [formato WebP](#) para los navegadores que lo soportan, pero al mismo tiempo mantener las imágenes JPEG para el resto de navegadores, debes utilizar lo siguiente:

`<picture>`

```
<source type="image/webp"srcset="images/butterfly.webp">
```

```

```

`</picture>`