

# Amplia Imágenes RWD

---

## Amplia Imágenes RWD

- img: usando width en CSS
- img y srcset
- picture
- img, srcset y dpi
- Enlaces

## img: usando width en CSS

---

Ya se ha presentado el uso básico de las imágenes desde el punto de vista de la semántica y obviando la presentación: indicar en la etiqueta `img` los 4 atributos básicos, entre ellos la altura y la anchura en píxeles de la imagen:

```

```

En algunas situaciones deseamos que el tamaño de la imagen se adapte al espacio disponible ocupándolo completamente. En esas situaciones desde CSS podemos modificar su presentación:

```
<style>
    img.css {
        width: 100%; /* novedad */
        height: auto;
    }
</style>
</head>
<body>
    <p></p>
    <p></p>
```

En el navegador se muestra si el ancho disponible es mayor que el de la imagen (podemos tener problemas de resolución)



O si modificamos el ancho disponible por debajo del ancho de la imagen



Desde CSS es posible controlar que el ancho máximo y mínimo con `max-width` y `min-width`. Observe en el código de ejemplo que hay una tercera imagen que crece con el ancho disponible hasta que alcanza un tamaño máximo de 360px y un mínimo de 150px.

Puede probar este ejemplo en el fichero `basico.html`.

## img y srcset

Hay varios posible usos del atributo `srcset` con imágenes. El más utilizado es proponer al navegador la misma imagen pero en distintas resoluciones y que el navegador utilice la más adecuada en función del ancho disponible.

En el ejemplo `dog_srcset.html` se presentan una imagen con distintas resoluciones. El código será similar a este:

```



```

El nuevo atributo `sizes` con el valor `100%` indica que la imagen se presente al 100% del ancho disponible (se pueden encontrar un uso más avanzados usando expresiones mediaQuery como las de CSS). Y en el atributo `srcset` se van especificando cada una de las imágenes junto con el ancho que le corresponden. Se sigue manteniendo el atributo `src` por si el navegador no soporta `srcset`.

Para poder observar lo que ocurre debemos activar la herramientas de desarrollo (F12 y observar la pestaña de Network) para apreciar cómo se cargan los distintos ficheros de la misma imagen a medida que cambia el ancho disponible)


Document - Chromium

Document x +

127.0.0.1:3000/ejemplo\_dog/dog\_01.html

Aplicaciones

uso de srcset con width



DevTools is now available in Spanish! Always match Chrome's language Switch DevTools

Elements Console Sources Network Performance Memory

Filter ☐ Invert ☐ Hide data URLs All Fetch/XHR JS CSS Img

Name	Status
dog_01.html	200
__vscode_livepreview_injected_script	200
dog_c_scale,w_200.jpg	200
127.0.0.1	101


Document - Chromium

Document x +

← → ↻ ⓘ 127.0.0.1:3000/ejemplo\_dog/dog\_01.html

Aplicaciones

uso de srcset con width



DevTools is now available in Spanish! Always match Chrome's language

Elements Console Sources Network Performance

⏏ ⛔ 🔍 ☐ Preserve log ☒ Disable cache No throttling

Filter ☐ Invert ☐ Hide data URLs All Fetch

☐ 3rd-party requests

10 ms 20 ms 30 ms 40 ms

Name	Status
127.0.0.1	101
dog_01.html	200
__vscode_livepreview_injected_script	200
dog_c_scale,w_591.jpg	200


Document - Chromium

Document x +

← → ↻ ⓘ 127.0.0.1:3000/ejemplo\_dog/dog\_01.html

Aplicaciones

uso de srcset con width



DevTools is now available in Spanish! Always match Chrome's language

Elements Console

⏏ ⛔ 🔍 ☐ Preserve

Filter ☐ Ir

All Fetch/XHR JS CSS Img Me

☐ Blocked Requests ☐ 3rd-party

10 ms 20 ms 30 ms

Name
dog_01.html
__vscode_livepreview_inject...
dog_c_scale,w_879.jpg
127.0.0.1

## picture

La idea de `picture` es similar a los elementos audio o vídeo: permitir la misma imagen en distintos formatos, de tal forma que si el navegador no soporta un formato intente el siguiente. Siempre debe existir al final un etiqueta `img` por si el navegador no soporta `picture` o los formatos ofrecidos.

```
<picture>
  <source type="image/svg+xml" srcset="logo.xml">
  <source type="image/webp" srcset="logo.webp">
  
</picture>
```

También es posible usarla para lo que llaman "*art direction*": seleccionar una versión distinta de la imagen (cambiando por ejemplo el foco) en función de por ejemplo la orientación o el ancho disponible. Una cosa a destacar es que en el caso de `picture` el atributo `media` no es una sugerencia (como `srcset` en `img`) sino una directiva que el navegador debe respetar.

```
<picture>
  <source media="(min-width: 650px)" srcset="images/kitten-large.png">
  <source media="(min-width: 465px)" srcset="images/kitten-medium.png">
  <!-- img tag for browsers that do not support picture element -->
  
</picture>
```

Este ejemplo está extraído de <https://googlechrome.github.io/samples/picture-element/>

Puede ver ejemplos de estos usos en `picture_art_direction.html` y `picture_tipòs.html`

## img, srcset y dpi

En el fichero `fairy_srcset_dpi.html` puede observar como le indicamos que imagen descargar en función de la densidad de píxeles del dispositivo.

```

```

En el fichero de ejemplo se le indica cómo observar dicho comportamiento activando en el navegador las herramientas de desarrollo, etc.

## Enlaces

- Picture y el gato: <https://googlechrome.github.io/samples/picture-element/>
- Bastante completo a nuestro nivel: MDN Responsive images: [https://developer.mozilla.org/en-US/docs/Learn/HTML/Multimedia\\_and\\_embedding/Responsive\\_images](https://developer.mozilla.org/en-US/docs/Learn/HTML/Multimedia_and_embedding/Responsive_images)
- Avanzado con mucho detalle: <https://cloudfour.com/thinks/responsive-images-101-definitions/>

