Images in CSS and HTML

Péter Jeszenszky

Faculty of Informatics, University of Debrecen

December 1, 2025

Terms of Use

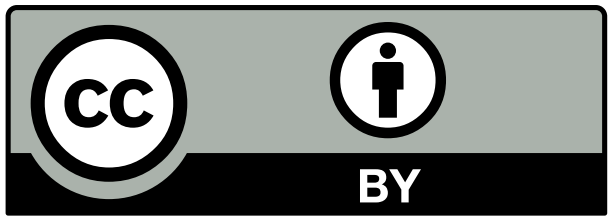

Image Statistics (1)

- Images contribute significantly to web traffic.
- Statistics show that images make up about 40% of the data transmitted over the web.
  - Consider the ratio of the Image Bytes and Total Kilobytes statistics on HTTP Archive.
- See:
  - HTTP Archive: State of the Web
  - HTTP Archive: State of Images

Image Statistics (2)

Usage statistics of image formats on the web:

- Usage statistics of image file formats for websites (W3Techs)
- Historical yearly trends in the usage statistics of image file formats for websites (W3Techs)

Image Formats

- The HTML standard does not require browsers to support any image format. However, browsers may support different image formats.
- Some notable image formats that are used and supported widely:
  - AVIF
  - JPEG
  - PNG
  - SVG
  - WebP
- Further information: Image file type and format guide (MDN)

Choosing an Image Format (1)

- Key factors to consider when choosing an image format:
  - Compression ratio
  - Is the loss of quality acceptable?
  - Encoding/decoding speed
  - Feature richness (e.g., is transparency or progressive rendering supported)
  - Browser support
- See: Addy Osmani. Using Modern Image Formats: AVIF And WebP. Sep 29, 2021.

Choosing an Image Format (2)

Features of interest to developers:

- Alpha channel (transparency): a color component in a color model that represents the extent to which the background can be seen through the color.
  - 0 or 0% represents complete transparency, and 1 or 100% represents complete opaqueness, respectively.
  - See: Alpha (alpha channel) (MDN)
- Color modes: e.g., grayscale, indexed, true color, ...
- Progressive rendering: it means to start rendering the image before it is fully downloaded; thus, a lower-resolution image is shown first.
  - For example, PNG and JPEG support it; however, AVIF currently doesn't.

Image Formats: JPEG (1)

JPEG is a lossy image compression method used primarily for encoding photographic images.

- Developer: Joint Photographic Experts Group
- Website: https://jpeg.org/
- Media type: `image/jpeg`
- File extension: `.jpeg`, `.jpg`

Image Formats: JPEG (2)

- The actual file format for storing JPEG-compressed image data is JFIF (JPEG File Interchange Format).

Image Formats: JPEG (3)

Specifications:

- ISO/IEC 10918-1:1994: Information technology—Digital compression and coding of continuous-tone still images: Requirements and guidelines
- ECMA TR/98: JPEG File Interchange Format (JFIF). June 2009.

Image Formats: JPEG (4)

- The original standard is now called JPEG 1.
  - Browser support: all major web browsers support JPEG 1 natively.
- Over the years, several enhancements to the original JPEG standard were proposed.
- The most recent such member of the family of JPEG standards is JPEG XL, which offers superior performance compared to legacy JPEG.
  - Browser support: https://caniuse.com/jpegxl

Image Formats: JPEG (5)

Progressive image rendering example:
https://nerds.sh/snippets/baseline-vs-progressive.html

- You should open DevTools and reload the page with cache disabled and simulating a slow connection.
  - Hint: the network throttling feature can be found on the Network tab.
- See: Daniel Schiau. Progressive JPEG in action! — DEMO. Feb 11, 2022.

Image Formats: PNG (1)

Portable Network Graphics (PNG) is an open, lossless raster image format that supports alpha (transparency).

- Developer: W3C
- Website: http://www.libpng.org/pub/png/
  https://www.w3.org/Graphics/PNG/
- Media type: `image/png`
- File extension: `.png`

Image Formats: PNG (2)

- Originally, it was developed to provide a patent-free replacement for the GIF image format that had controversial patent issues in the 90's.
  - Originally, the abbreviation PNG was resolved as "PNG's Not GIF".

Image Formats: PNG (3)

Specifications:

- Portable Network Graphics (PNG) Specification (Third Edition). W3C Recommendation, 24 June 2025.
- ISO/IEC 15948:2004: Information technology—Computer graphics and image processing—Portable Network Graphics (PNG): Functional specification

Image Formats: PNG (4)

- Browser support: all major web browsers support PNG natively.
  - See: https://caniuse.com/png-alpha
- Examples:
  - http://www.libpng.org/pub/png/png-sitemap.html#images

Image Formats: PNG (5)

- APNG is an extension of the PNG format proposed by Mozilla, adding support for animated images.
  - Media type: `image/apng`
  - File extension: `.apng`
  - Specification: https://wiki.mozilla.org/APNG_Specification
  - Browser support: https://caniuse.com/apng
  - Examples: https://commons.wikimedia.org/wiki/Category:Animated_PNG_files

Image Formats: WebP (1)

An open raster image format that supports both lossless and lossy compression as well as alpha (transparency) and animation.

- Developer: Google
- Website: https://developers.google.com/speed/webp
- Media type: `image/webp`
- File extension: `.webp`

Image Formats: WebP (2)

- It is optimized for fast image transfer over the network (for example, for websites).
- It is intended to replace older formats, such as JPEG or PNG, offering much better efficiency.
  - WebP lossless images are 26% smaller in size compared to PNGs.
  - WebP lossy images are 25-34% smaller than comparable JPEG images of equivalent quality.
- It uses the RIFF format as a container.
  - RIFF is also used by many other media formats, such as AVI or WAV.

Image Formats: WebP (3)

Specifications:

- James Zern, Pascal Massimino, Jyrki Alakuijala. WebP Image Format. RFC 9649, November 2024.
- RIFF (Resource Interchange File Format)

Image Formats: WebP (4)

- Browser support: all major web browsers support WebP natively.
  - See: https://caniuse.com/webp
- Examples:
  - WebP Image Galleries (Google)

Image Formats: AVIF (1)

An open raster image format that supports both lossless and lossy compression as well as alpha (transparency) and animation.

- Developer: Alliance for Open Media
- Website: https://aomedia.org/specifications/avif/
- Media type: `image/avif`
- File extension: `.avif`

Image Formats: AVIF (2)

- AVIF can provide over 50% file size savings compared to JPEG and over 30% savings compared to WebP.
- It is based on the AV1 video codec and the HEIF container format.

Image Formats: AVIF (3)

Specifications:

- Yannis Guyon (ed.), Leo Barnes (ed.), Wan-Teh Chang (ed). AV1 Image File Format (AVIF). AOM Final Deliverable, 16 October 2025.
- Peter de Rivaz, Jack Haughton. AV1 Bitstream & Decoding Process Specification. January 8, 2019.
- ISO/IEC 23008-12:2025: Information technology—High efficiency coding and media delivery in heterogeneous environments—Part 12: Image File Format (HEIF)

Image Formats: AVIF (4)

- Browser support: AVIF support has been added to major browsers in the last few years only.
  - See: https://caniuse.com/avif
- Examples:
  - https://aomediacodec.github.io/av1-avif/testFiles/Link-U/
    https://github.com/link-u/avif-sample-images

Image Formats: SVG (1)

An open image format for describing two-dimensional vector graphics in XML.

- Developer: W3C
- Website: https://www.w3.org/Graphics/SVG/
- Media type: `image/svg+xml`
- File extension: `.svg`

Image Formats: SVG (2)

- It also supports interactive graphics and animation.
- SVG content can be embedded inline within HTML documents, see the `<svg>` element.

Image Formats: SVG (3)

Specifications:

- Scalable Vector Graphics (SVG) 1.1 (Second Edition). W3C Recommendation, 16 August 2011.
- Scalable Vector Graphics (SVG) 2. W3C Candidate Recommendation, 4 October 2018.

Image Formats: SVG (4)

- Browser support: all major web browsers support SVG natively.
  - See: https://caniuse.com/svg
- Examples:
  - Bootstrap Icons (license: MIT License) https://icons.getbootstrap.com/ https://github.com/twbs/icons/
  - Material Symbols and Icons (Google Fonts)

Tools for Working with Images

Free and open source tools:

- GNU Image Manipulation Program (GIMP)
- ImageMagick

Tools for Working with Images: GIMP (1)

A free and open source image manipulation and paint program.

- Website: https://www.gimp.org/
- Repository: https://gitlab.gnome.org/GNOME/gimp
- License: GPLv3
- Platform: Linux, macOS, Windows
- Written in: C

Tools for Working with Images: GIMP (2)

- Can load and save a variety of image formats and can be used to convert between formats, including WebP and AVIF.
- Offers a variety of filters to perform image manipulations (e.g., edge detection, Gaussian blur, red eye removal, pixelization).

Tools for Working with Images: ImageMagick (1)

A free and open-source software suite used for editing and manipulating digital images on the command line.

- Website: https://imagemagick.org/
- Repository: https://github.com/ImageMagick/ImageMagick
- License: ImageMagick License (nearly identical to the Apache-2.0 license)
- Platform: Linux, macOS, Windows
- Written in: C

Supports a wide range of file formats.

Tools for Working with Images: ImageMagick (2)

Example of use:

```
convert -list format # lists supported formats

convert image.png image.avif # converts image.png to image.avif

# resizes input.png to half of its original size:
convert input.png -resize 50% output.png

# converts all PNG files in the current directory to WebP:
mogrify -format webp *.png

# resizes all PNG files in the current directory to half of their sizes:
mkdir output
mogrify -path output/ -resize 50% *.png
```

Images in CSS

The following CSS properties accept images as part of their values (the list is not exhaustive):

- `background`, `background-image`
- `content`
- `list-style`, `list-style-image`
- `cursor`

The background-image CSS Property (1)

- The background-image CSS property sets one or more background images on an element.
  - The value of the property is a comma-separated list of one or more images.
- Drawing of the background images:
  - The background images are drawn on stacking context layers on top of each other.
    - The first layer specified is drawn as if it were closest to the user.
  - The borders of the element are then drawn on top of them, and the background-color is drawn beneath them.
- Several CSS properties are provided to control the display of background images.

The background-image CSS Property (2)

The background property is a shorthand for the following CSS properties:

- background-attachment: sets whether a background image's position is fixed within the viewport, or scrolls
- background-clip: sets whether an element's background extends underneath its border box, padding box, or content box
- background-color: sets the background color
- background-image: sets one or more background images
- background-origin: sets the area within which the background images are positioned (border-box, content-box, or padding-box)
- background-position: sets the initial position for each background image
- background-repeat: sets how background images are repeated
- background-size: sets the size of the background image

The background-image CSS Property (3)

Examples (`logo.svg` refers to the official logo of the University of Debrecen that can be found here):

- ```
  body {
    background-image: url("logo.svg");
    background-repeat: repeat; /* this is the default value */
  }
  ```
- ```
  body {
    background-image: url("logo.svg");
    background-repeat: space;
  }
  ```
- ```
  body {
    background-image: url("logo.svg");
    background-position: center;
    background-repeat: no-repeat;
  }
  ```
  ```

The background-image CSS Property (4)

Examples (#fbab2c and #054434 are the two colors of the Unideb logo):

- ```
  body {
    background-image: url("logo.svg"),
      linear-gradient(#fbab2c, #054434);
    background-position: center;
    background-repeat: no-repeat;
  }
  ```
- ```
  body {
    background-attachment: fixed;
    background-image: url("logo.svg"),
      linear-gradient(#fbab2c, #054434);
    background-position: center;
    background-repeat: no-repeat;
  }
  ```

The `list-style-image` CSS Property (1)

- The `list-style-image` CSS property sets an image to be used as the list item marker.
- This property is inherited; thus, it can be set on the parent element (i.e., `<ol>` or `<ul>` in HTML) to apply to all list items.

The `list-style-image` CSS Property (2)

- Example:

```
/* CSS: */
ul.planets {
  list-style-image: url(planet.svg);
}

<!-- HTML: -->
<ul class="planets">
  <li>Mercury</li>
  <li>Venus</li>
  <li>Earth</li>
  <li>Mars</li>
</ul>
```

The cursor CSS Property

- The `cursor` CSS property sets the mouse cursor, if any, to show when the mouse pointer is over an element.
- Example:

```
form.hogwarts-form :is(input, select, textarea) {
  cursor: url("https://icons.getbootstrap.com/assets/icons/magic.svg"),
    default;
}
```

The <img> element (1)

- The <img> element embeds an image into an HTML document.
- It is a replaced element that has a display value of inline by default.
- Depending on its type, an image may have an intrinsic width and height.
  - For example, SVG images without explicitly set height or width attributes on their root <svg> elements do not have an intrinsic dimension.
- See:
  - <img>: The Image Embed element (MDN)
  - Images in HTML (MDN)
  - Images (HTML Standard)

The <img> element (2)

- Required attributes:
  - `src`: contains the URI of the image resource.
  - `alt`: contains a replacement text for the image; it is essential for accessibility.
- Optional attributes:
  - `height`/`width`: an integer without a unit that specifies the intrinsic height/width of the image in pixels.
  - `loading`: indicates how the browser should load the image, either `eager` (this is the default value) or `lazy`.

The <img> element: alt attribute (1)

- Contains a replacement text for the image.
- The user agent displays the replacement text when, for example:
  - It is not capable of displaying images, such as a text-based web browser.
  - When the image can't be loaded, e.g., due to a network error, or the file format of the image is not supported by the user agent.
  - The user chooses not to load images, e.g., to save bandwidth or to protect their privacy.

The <img> element: alt attribute (2)

- An alt attribute's value should provide clear and concise text replacement for the image's content.
- The rule of thumb to consider when writing alternative text:
  - The intent is that replacing every image with the text of its alt attribute does not change the meaning of the page.
- The purpose of providing an alternative text is very different from that of a caption.

The <img> element: alt attribute (3)

- The alt attribute's value should never contain text that could be considered the image's caption, title, or legend.
  - Instead, it is supposed to contain replacement text that could be used by users instead of the image; it is not meant to supplement the image.
  - From an accessibility viewpoint, captions and alt text have distinct roles.
    - Captions benefit even people who can see the image, whereas alt text provides the same functionality as an absent image.
    - Therefore, captions and alt text shouldn't just say the same thing, because they both appear when the image is gone.
- If an image needs a caption, use the figure and figcaption elements.

The <img> element: alt attribute (4)

- Setting the alt attribute to an empty string (i.e., alt="") indicates that the image is not a key part of the content, e.g., it serves purely decorative purposes.

The <img> element: alt attribute (5)

Example: a good alternative text

```
<p>Today I found this cat in front of our door:
  <img src="cat.jpg" alt="The cat is napping on the ground"
    width="640" height="480">
</p>
```

The <img> element: alt attribute (6)

Example: both of the following two alternative texts are bad

- ```
  Today I found this cat in front of our door:
    <img src="cat.jpg"
      alt="A cat napping on the ground"
      width="640" height="480">

  ```
- ```
  <p>Today I found this cat in front of our door:
    <img src="cat.jpg"
      alt="A photo of a cat napping on the ground"
      width="640" height="480">
  </p>
  ```

The <img> element: height/width attribute (1)

- They specify the intrinsic height/width of the image, in pixels.
- Including height and width enables the aspect ratio of the image to be calculated by the browser before the image is loaded.
  - This aspect ratio is used to reserve the space needed to display the image, reducing or even preventing a layout shift when the image is downloaded and painted to the screen.
  - Browser support: https://caniuse.com/mdn-html_elements_img_aspect_ratio_computed_from_attributes

The <img> element: height/width attribute (2)

- It is considered to be a good practice to specify the intrinsic size of the image with the `height` and `width` attributes.
  - See: Barry Pollard. Setting Height And Width On Images Is Important Again. January 11, 2022.
- The effect of the absence or presence of the `height` and `width` attributes on the rendering can be measured in Chrome DevTools, see Cumulative Layout Shift (CLS) on the Performance tab.

The <img> element: height/width attribute (3)

- Example that demonstrates the effect of the absence or presence of the `height` and `width` attributes: https://image-layout-shift.surge.sh/
  - The page should be viewed with cache disabled and simulating a slow network connection.

The <img> element: loading attribute (1)

- The loading attribute of the <img> element indicates how the browser should load the image.
- See also: Lazy loading (MDN)

The <img> element: loading attribute (2)

Valid attribute values:

- `eager`: Loads the image immediately, regardless of whether or not the image is currently within the visible viewport (this is the default value).
- `lazy`: Defers loading the image until it reaches a calculated distance from the viewport, as defined by the browser.
  - The intent is to avoid the network and storage bandwidth needed to handle the image until it's reasonably certain that it will be needed.
  - This generally improves the performance of the content in most typical use cases.
  - It is recommended to specify a preferred aspect ratio via the `width` and `height` attributes on lazy-loaded images, even if CSS sets the image's `width` and `height` properties, to prevent the page layout from shifting around after the image loads.

The <img> element: loading attribute (3)

Example:

```
<img loading="lazy" src="cat.jpg" alt="A cat napping"
     width="640" height="480">
```

The <figure> element (1)

- The <figure> HTML element represents self-contained content, potentially with an optional caption, which is specified using the <figcaption> element.
- The figure, its caption, and its contents are referenced as a single unit.
- Usually, a <figure> is an image, illustration, diagram, code snippet, etc., that is referenced in the main flow of a document, but that can be moved to another part of the document or to an appendix without affecting the main flow.
- See:
  - <figure>: The Figure with Optional Caption element
  - <figcaption>: The Figure Caption element

The <figure> element (2)

Example:

```
<p><a href="#cat-guest">Figure 1</a> shows an unexpected
  guest (i.e., a friendly cat) I found today in front of
  our door.</p>

<figure id="cat-guest">
  <img src="cat.jpg"
    alt="The cat is napping on the ground"
    width="640" height="480">
  <figcaption>Figure 1. An unexpected cat guest in front
    of our door.</figcaption>
</figure>
```

The <map> element

- The <map> HTML element is used with <area> elements to define an image map (a clickable link area).
- See:
  - <map>: The Image Map element
  - <area>: The Image Map Area element

Responsive Images (1)

Sources:

- HTML Living Standard – Images
- MDN Web Docs – Responsive images
- MDN Web Docs – <img>: The Image Embed element

Responsive Images (2)

- There are many situations in which the author might wish to use multiple image resources that the user agent can choose from:
  - Different users might have different environmental characteristics, such as
    - physical screen size,
    - screen pixel density,
    - zoom level,
    - screen orientation,
    - network speed and bandwidth cost.
  - Authors might want to show the same image content but with different rendered sizes, depending on, usually, the width of the viewport (viewport-based selection).
  - Authors might want to show different image content depending on the rendered size of the image (art direction).
- The above situations are not mutually exclusive.

Responsive Images (3)

The `srcset` and the `sizes` attributes:

- They can be specified on the `img` and the `source` elements.
- They are provided for specifying a set of alternative resources from which the user agent can choose the most appropriate one to use.

Responsive Images (4)

The `srcset` attribute:

- The value of the attribute is a comma-separated list of one or more strings (image candidate strings), each of which consists of the following whitespace-separated components:
  - A URI to an image.
  - Optionally, one of the following descriptors:
    - **Width descriptor**: a positive integer directly followed by `w`, it specifies the inherent width of the image in pixels.
    - **Pixel density descriptor**: a positive floating-point number followed by `x`.
  - If no descriptor is specified for an image candidate string, the default is `1x`.

Responsive Images (5)

The `sizes` attribute:

- The value of the attribute is a comma-separated list of one or more strings, each of which consists of the following components:
  - A media condition that must be omitted for the last item in the list.
  - A source size value (i.e., a non-negative length) that specifies the intended display width of the image.

Responsive Images (6)

The `picture` element:

- Contains **zero or more** `source` elements, followed by **exactly one** `img` element to provide multiple versions of an image for different display/device scenarios.
  - The browser will consider each child source element and choose the best match among them; if no matches are found, the URI of the `img` element's `src` attribute is selected.
  - The selected image is then presented in the space occupied by the `img` element.
- See:
  - HTML Living Standard – The picture element
  - MDN Web Docs – <picture>: The Picture element

Responsive Images (7)

Examples:

https://jeszy75.github.io/responsive-image-examples/
https://github.com/jeszy75/responsive-image-examples

Responsive Images: Device-pixel-ratio-based Selection (1)

- We consider the scenario in which the rendered size of the image is fixed!
- The user agent can choose any of the given resources depending on the user's screen's pixel density, zoom level, and possibly other factors such as the user's network conditions.

Responsive Images: Device-pixel-ratio-based Selection (2)

Example:

```
<style>
  img#cat {
    width: 320px;
  }
</style>

<img src="cat-320.jpg"
     srcset="cat-480.jpg 1.5x, cat-640.jpg 2x"
     alt="A cat napping"
     id="cat">
```

Responsive Images: Viewport-based selection (1)

- Multiple images are provided that only vary in their size.
- The user agent will calculate the effective width of each image from the specified `w` descriptors and the specified rendered size in the `sizes` attribute.
- It can then choose any of the given resources depending on the user's screen's pixel density, zoom level, and possibly other factors such as the user's network conditions.

Responsive Images: Viewport-based Selection (2)

Example:

```
<img srcset="cat-320.jpg 320w,
             cat-640.jpg 640w,
             cat-800.jpg 800w,
             cat-1024.jpg 1024w"
     sizes="(max-width: 768px) 320px,
            (max-width: 992px) 640px
            (max-width: 1200px) 800px,
            1024px"
     src="cat-640.jpg"
     alt="A cat napping">
```

Responsive Images: Art Direction-based Selection

- The `picture` element and the `source` element, together with the `media` attribute, can be used to provide multiple images that vary the image content.
  - For example, a smaller image might be a cropped version of a bigger image.
- Example:

```
<picture>
  <source media="(min-width: 1200px)" srcset="cat-desktop.jpg">
  <source media="(min-width: 768px)" srcset="cat-tablet.jpg">
  <img src="cat-mobile.jpg" alt="A cat napping">
</picture>
```

  - The rendered size of the image varies depending on which resource is chosen.
  - To specify dimensions that the user agent can use before having downloaded the image, CSS can be used.

Responsive Images: Image Format-based Selection

- The `type` attribute on the `source` element can be used to provide multiple images in different formats.
- Example:

```
<picture>
  <source srcset="cat.avif" type="image/avif">
  <source srcset="cat.webp" type="image/webp">
  <img src="cat.jpg" alt="A cat napping"
       width="640" height="480">
</picture>
```