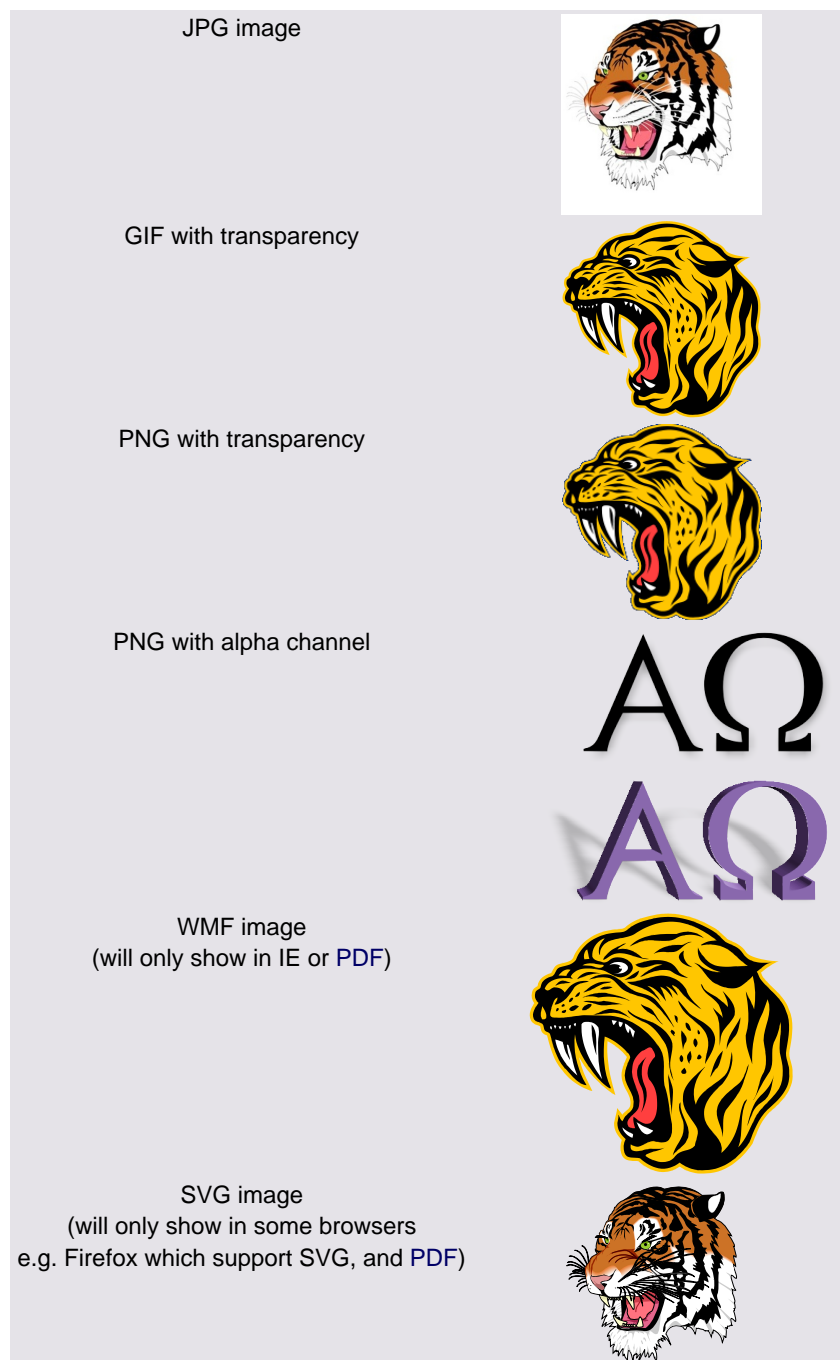


# Images

Images are supported by mPDF: GIF, PNG, JPG, WMF, SVG, BMP and generated images from **PHP** scripts.

Transparent GIF images are supported, and so are interlaced and transparent PNG files, including transparency using alpha channel.



**Debugging Errors:** If you are having trouble with images not displaying correctly, set the variable `$mpdf->showImageErrors = true`.

Images are handled in mPDF as in-line elements.

Unlike the HTML specification, the width and height attributes of IMG will additionally take different dimensions e.g. mm, pt etc.

```
 // 90 pixels, just like HTML
```

```
 // non-standard support for dimensions
```

```
 // Can also use CSS
```

Images for which the size is not defined are output at a default dpi set in the `config.php` file:

```
$this->img_dpi = 96;
```

In addition, many CSS style properties are supported including *vertical-align*, as well as some custom attributes such as *opacity* and *rotate*. (See Supported CSS and HTML attributes)

Images can be used in:

- In-line images, including in HTML headers & footers
- Watermarks
- CSS background-image

### Size constraint

When writing HTML, image size is automatically constrained to current margins and page position. An extra parameter added to end of the `Image()` function allows you to override this:

```
$mpdf->Image('files/images/frontcover.jpg',0,0,210,297,'jpg','',true, false);
// the last "false" allows a full page picture
```

This is useful for e.g. a cover page for your document.

This can be achieved using HTML & CSS like this:

```
<div style="position: absolute; left:0; right: 0; top: 0; bottom: 0;">

</div>
```

### Float and Text wrapping

mPDF partially supports the CSS style `float` with `IMG` elements. Text will wrap around these images, with certain limitations:

- only 1 `float:right` and 1 `float:left` image are allowed at a time i.e. you cannot overlap 2 right or 2 left
- the containing HTML element is extended at the bottom if necessary to enclose the last image (unlike your browser)
- the float is ignored if the image is too wide, inside a table, columns are on, or `div page-break-inside: avoid` is set
- disabled when columns are being used

If ignored, the image is output as a normal in-line element.

### Performance

JPG images are quickest and most efficient.

PNG images with no alpha channel and not interlaced are next best.

GIF images are extremely slow if you do not have the appropriate GD library installed (above 2.0.8).

GIF images with GD library installed are quite fast.

PNG images which are interlaced or have alpha channel transparency, and GIF images use the GD library if it is installed.

Any process that requires the GD library uses a large amount of memory - to create a GD image in memory can use up to 10 x the file size (e.g. a 690K GIF file read into `imagecreatefromstring()` used about 5MB of PHP memory).

### Dynamically generated Images

You can define a script which generates an image just as you can in HTML:

```

```

### Embedded Image data

Embedded image data can be used either in `<img>` elements or in `CSS:background`. gif, png and jpeg are supported.

```
[HTML]


[CSS stylesheet]
div { background: url(data:image/gif;base64,....) no-repeat 4px center; }
```

**Image data as a Variable**

A PHP variable containing image data can be passed directly to mPDF. You need to allocate the data to a class variable  
(you can make any name up) e.g.

```
$mpdf->myvariable = file_get_contents('alpha.png');
```

The image can then be referred to by use of "var:varname" instead of a file name, either in src="" or direct to Image() function e.g.

```
$html = '';  
$mpdf->WriteHTML($html);
```

```
$mpdf->Image('var: myvariable',0,0);
```

**SVG images**

mPDF partially supports SVG images, including as embedded HTML e.g.:

```
<p>This is an embedded SVG image:</p>  
<svg>.....</svg>
```

All units with no dimension are taken as pixels.

**Size of SVG image**

The viewBox attribute of the <svg> element is used (if present) to determine the intrinsic size of the image.

If viewBox is not present, a width and/or height be specified e.g. width="400" height="200"

NB preserveAspectRatio is not supported.

If none of these are present, the intrinsic size will be set as the width of containing block (height = width).

**Partially supported**

- embedded images (but not embedded SVG/WMF [vector] images)
- tspan, tref
- <use ../>
- gradient on text (fill)
- <clipPath>
- embedded <style> elements\*
- automatic font selection for text\*

\* As from mPDF 6.0 there is limited support for CSS classes and for automatic font selection (see the defined constants at the top of classes/svg.php file).

**Not supported**

- filters
- <marker>
- DOM
- <pattern>
- textlength, lengthadjust, toap or textPath
- gradient on stroke

Note: SVG images can be embedded within the HTML code. They may be a useful way to deal with some presentation issues not supported by HTML/CSS e.g. Text used with a transformation, or text used as a clipPath e.g.

```
<body>  
<div>  
...  
<svg width="100%" height="100%" viewBox="0 0 480 360">  
  <text x="100" y="-100" transform="rotate(60)" font-size="35">ROTATE</text>  
  <rect id="test-frame" x="1" y="1" width="478" height="358" fill="none" stroke="#000000"/>  
</svg>  
...  
<svg width="100%" height="100%" viewBox="0 0 480 360">  
  <defs>  
    <clipPath id="sample" clipPathUnits="userSpaceOnUse">  
      <text x="45" y="270" font-size="100" font-family="Impact">Clip Test</text>  
    </clipPath>  
  </defs>  
  <image xlink:href="bluesquidj.png" preserveAspectRatio="none" x="20" y="170" width="410"  
height="160" clip-path="url(#sample)"/>  
  <rect id="test-frame" x="1" y="1" width="478" height="358" fill="none" stroke="#000000"/>  
</svg>  
...
```

```
</div>  
</body>
```

Printed on Wed 05 Aug 2015 12:09:06 GMT +0100 (DST)