

The fancyqr package

Florian Sihler



Version v2.2 – 2024/11/27

`fancyqr` is a simple package to create fancy qr-codes with the help of the `qrcode`-package. You can use the `\fancyqr`-macro just like the normal `\qrcode`.¹

If you do want to hide a center square (e.g. because you want to embed an image) you can use `\FancyQrDoNotPrintSquare{<x>}{<y>}` to hide a rectangle with radius `x` and `y` set from the center (with `\FancyQrDoNotPrintRadius{<factor>}` you can apply a rounding to this!). If you choose this option, the default `\FancyQrRoundCut` that rounds cut corners can be changed with `\FancyQrHardCut`. At the moment, there are five additional styles for the qr-code: `flat`, `frame`, `blobs`, `glitch`, and `dots`, that you can load (locally) by using `\FancyQrLoad{<name>}`. The default style is named `default` and can be 'reset' by `\FancyQrLoad{default}` or `\FancyQrLoadDefault`.

All of the extra qr-options (you can set all of them with `\fancyqrset{<keys>}`) are showcased in Table 1. The defaults are set like this:

```
\fancyqrset{image padding=0,gradient=true,
            gradient angle=135,r color=teal,l color=purple}
```

Consider the following example (uses `fontawesome` for the symbol, but you can use include images, ...):

```
\fancyqr [image=\huge\faGithub,image padding=1,color=black
          !90!gray]{https://github.com/EagleoutIce/fancyqr}
```

Which produces:



¹`\fancyqr[<qr-options>]{<url>}`

Option	Type	Default	Explanation
<code>classic</code>	boolean	<code>false</code>	Use the classic qr-code style (black with flat rectangles, this loads the <code>flat</code> style).
<code>color</code>	color		Disables the <code>gradient</code> and sets the color accordingly.
<code>compensate</code>	length	<code>0.15pt</code>	Compensating overlap to add to avoid glitches.
<code>gradient</code>	boolean	<code>true</code>	Toggle the color gradient
<code>gradient angle</code>	angle	<code>135</code>	Change the gradient angle.
<code>image</code>	L ^A T _E X		Automatically center an image (you have to care for the size and maybe adjust the <code>version</code> and <code>level</code> to keep the qr-code readable). ²
<code>image padding</code>	number		Additionally hide blocks (x & y) around the image.
<code>image x padding</code>	number	<code>0</code>	Additionally hide blocks (x) around the image.
<code>image y padding</code>	number	<code>0</code>	Additionally hide blocks (y) around the image.
<code>l color</code>	color	<code>purple</code>	Set the top left gradient color.
<code>left color</code>	color		Alias for <code>l color</code> .
<code>level</code>	L/M/Q/H	<code>M</code>	<code>qr</code> code option affecting error correction (low, medium, quartile, high).
<code>padding</code>	flag		<code>qr</code> code option adding sufficient additional space around the qr-code.
<code>r color</code>	color	<code>teal</code>	Set the bottom right gradient color.
<code>random color</code>	colors		Allow to set a random color pool to pick from.
<code>right color</code>	color		Alias for <code>r color</code> .
<code>size</code>	length		Alias for <code>qr</code> code's <code>height</code> option.
<code>tight</code>	flag		<code>qr</code> code option adding no additional space around the qr-code.
<code>version</code>	$[0..40] \in \mathbb{N}$	<code>0</code>	<code>qr</code> code option affecting the size (tries to be as small as possible).
<code>width</code>	length		Alias for <code>qr</code> code's <code>height</code> option.

Table 1: Overview of special qr-options.

²The package will automatically calculate the required `\FancyQrDoNotPrintSquare` (you have to make sure that the qr-code still has enough information to be readable). Therefore, the image will not scale with the qr-code.