

---

# **lookatme.contrib.qrcode**

## **Documentation**

*Release {{VERSION}}*

**James 'd0c\_s4vage' Johnson**

**Oct 23, 2020**



---

## Contents:

---

<b>1</b>	<b>Setup</b>	<b>3</b>
<b>2</b>	<b>qrcode Codeblocks</b>	<b>5</b>
<b>3</b>	<b>qrcode-ex Codeblocks</b>	<b>7</b>
3.1	contrib . . . . .	8
<b>4</b>	<b>Indices and tables</b>	<b>11</b>
	<b>Python Module Index</b>	<b>13</b>
	<b>Index</b>	<b>15</b>



This `lookatme` extension adds QR code rendering support to codeblocks within a presentation. If a codeblock sets its language as either `qrcode` or `qrcode-ex`, this extension will handle rendering the codeblock.



# CHAPTER 1

---

## Setup

---

Install this extension with:

```
pip install lookatme.contrib.qrcode
```

Declare that it should be loaded and used in your slides' header:

```
---
title: Slide title
extensions:
  - qrcode
---

# First Slide Header
```





## CHAPTER 2

---

### qr code Codeblocks

---

If the language is set to `qr code`, the data contained within the codeblock will be converted directly to a qr code, with the caption being automatically set to the codeblock's value.

```
# A Slide

```qr code
https://github.com/d0c-s4vage/lookatme.contrib.qr code
```
```

The markdown above is rendered as the qr code below:

*A Slide*

■ A Slide



<https://github.com/d0c-s4vage/lookatme.contrib.qrcode>

2019-12-11 05:55:16.013576

## CHAPTER 3

---

### qrcode-ex Codeblocks

---

If the language is set to `qrcode-ex`, the data contained within the codeblock must YAML data that matches the schema [QrSchema](#). The YAML data defines the columns of QR codes to display and their captions:

```
columns:
- data: REQUIRED
  autocaption: true # default
  caption: null # default
```

```
# A Slide

```qrcode-ex
columns:
- data: https://github.com/d0c-s4vage/lookatme
  caption: "**lookatme** project"
- data: https://github.com/d0c-s4vage/lookatme.contrib.qrcode
  caption: Text `lookatme.contrib.qrcode`
```
```

The markdown above is rendered as the qrcode below:



## 3.1 contrib

### 3.1.1 qrcode module

Adds qrcode rendering

```
class qrcode.QrColumn(*, only: Union[Sequence[str], Set[str]] = None, exclude:
    Union[Sequence[str], Set[str]] = (), many: bool = False, context: Dict[KT,
    VT] = None, load_only: Union[Sequence[str], Set[str]] = (), dump_only:
    Union[Sequence[str], Set[str]] = (), partial: Union[bool, Sequence[str],
    Set[str]] = False, unknown: str = None)
    Bases: marshmallow.schema.Schema

    opts = <marshmallow.schema.SchemaOpts object>

class qrcode.QrSchema(*, only: Union[Sequence[str], Set[str]] = None, exclude:
    Union[Sequence[str], Set[str]] = (), many: bool = False, context: Dict[KT,
    VT] = None, load_only: Union[Sequence[str], Set[str]] = (), dump_only:
    Union[Sequence[str], Set[str]] = (), partial: Union[bool, Sequence[str],
    Set[str]] = False, unknown: str = None)
    Bases: marshmallow.schema.Schema

class Meta
    Bases: object

    render_module
        alias of YamlRender
```

```
opts = <marshmallow.schema.SchemaOpts object>
```

```
class qrcode.YamlRender
```

```
Bases: object
```

```
dump()
```

```
loads()
```

```
qrcode.add_padding(code: List[List[int]], width: int = 4) → List[List[int]]
```

Add width number of blocks of padding to the code

#### Parameters

- **code** – The qrcode on/off image
- **width** – The amount of padding to add

```
qrcode.blocks_to_squares(blocks: List[List[int]]) → List[List[List[int]]]
```

Returns a list of list of blocks of four squares at a time. E.g.

```
blocks = [
    [A, B, 1, 1],
    [C, D, 0, 1],
]

# would return the list below containing a list of rows, with each
# row being a list of squares

[ [ [A, B, C, D], [1, 1, 0, 1] ] ]
```

In the returned squares, the square coords go:

- top-left
- top-right
- bottom-left
- bottom-right

**Parameters blocks** – List of blocks to transform into squares

```
qrcode.qrcode_ex(data: dict) → urwid.container.Columns
```

Handle the structured QR data (parsed from the YAML) to render QR codes for each defined column.

```
qrcode.qrcode_raw_render(data: str) → urwid.widget.Text
```

Render the qrcode into an urwid.Text

**Parameters data** – The data to encode in a qrcode and render

```
qrcode.qrcode_render(data: str, autocaption: bool = True, caption: str = None) → urwid.widget.Widget
```

Render a qrcode.

#### Parameters

- **data** – The data to render as a qrcode
- **autocaption** – If caption should be added automatically

```
qrcode.render_code(token: Dict[KT, VT], body: urwid.widget.Widget, stack: List[urwid.widget.Widget], loop: urwid.main_loop.MainLoop)
```

Main extension function that has first-chance at handling the `render_code` function `lookatme.render.markdown_block`

This extension ignores all code blocks **except** ones with the language `qrcode` or `qrcode-ex`.

- `qrcode-ex` - Expects the contents of the code block to be yaml of the form:

```
columns:
- data: "Data to be qr encoded"
  autocaption: true/false # optional
  caption: "Manual markdown caption" # optional
```

One or more columns may be defined to render QR codes side-by-side

If `autocaption` is `True` **and** `caption` is `None`, the caption will be set to the data that is QR encoded.

`qrcode.spec_from_square` (*square: List[int]*) → list

Create a list of `urwid.Text` specs from a square

**Parameters** `square` – The square to create the specs from

`qrcode.user_warnings` ()

No warnings here

## CHAPTER 4

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`





**q**

qrcode, 8



## A

`add_padding()` (*in module qrcode*), 9

## B

`blocks_to_squares()` (*in module qrcode*), 9

## D

`dumps()` (*qrcode.YamlRender method*), 9

## L

`loads()` (*qrcode.YamlRender method*), 9

## O

`opts` (*qrcode.QrColumn attribute*), 8

`opts` (*qrcode.QrSchema attribute*), 8

## Q

`qrcode` (*module*), 8

`qrcode_ex()` (*in module qrcode*), 9

`qrcode_raw_render()` (*in module qrcode*), 9

`qrcode_render()` (*in module qrcode*), 9

`QrColumn` (*class in qrcode*), 8

`QrSchema` (*class in qrcode*), 8

`QrSchema.Meta` (*class in qrcode*), 8

## R

`render_code()` (*in module qrcode*), 9

`render_module` (*qrcode.QrSchema.Meta attribute*), 8

## S

`spec_from_square()` (*in module qrcode*), 10

## U

`user_warnings()` (*in module qrcode*), 10

## Y

`YamlRender` (*class in qrcode*), 9