

# qolistings.sty Package

Peter Brookes Chambers

November 20, 2020

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Package Requirements . . . . .	1
<b>2</b>	<b>Usage</b>	<b>1</b>
2.1	The <code>\qoinputlisting</code> Command . . . . .	2
2.2	The <code>qolisting</code> Environment . . . . .	2
2.3	Optional Arguments . . . . .	2
<b>3</b>	<b>Examples</b>	<b>4</b>

# 1 Introduction

This package builds on the `listings` package to provide appealing and professional default styles with an easy interface for common options. It also incorporates the colour schemes defined in the `qocolours` package. However, the `qolistings` package can be used independently of `qocolours`; if `qocolours` is not used, then the default theme of `twilight` is also defined in the `qolistings` package.

If, however, `qocolours` is loaded before `qolistings` the scheme selected from `qocolours` is not overwritten; to allow for this functionality, a new command `\colourprovide` (note that this uses the U.K. spelling) is defined. This has the same syntax and behaviour as the `colorlet` command from the `xcolor` package, except that if the colour name is already in use it will not be changed. In other words, `\colourprovide` is to `\colorlet` as `\providecolor` is to `\definecolor`. Although this command is not really related to the rest of the `qolistings` package, it is public-facing and so included in this documentation.

All listings are encased in an `mdframed` environment. Then, `mdframed` handles the rendering of backgrounds and frames (if present), which produces much more consistent results than the frame and background handling in the `listings` package, especially for wrapped lines and variable line heights. By default, `tikz` rendering is used for the `mdframed` frames.

## 1.1 Package Requirements

In addition to `listings` and `mdframed`, the `qolistings` package requires several other packages to be present. These (in full) are as follows:

- `listings`
- `mdframed`
  - `tikz`
- `xcolor`
- `forarray`
- `keyval`
- `ifthen`
- `kvoptions`

## 2 Usage

The package can be loaded with `\usepackage{qolistings}`. Optional arguments can be given in a comma-separated key-value format. See section 2.3 for a list of optional arguments; those given in `blue` can be passed as optional arguments when loading the package. Unfortunately, `lst options`, `lst early options`, and `mdframed options` cannot be passed as arguments when loading the package. In addition, the `frame` option, which normally can be passed without a value, **must** be given a value when being passed to `\usepackage`. Any option given when loading the package will be used as the default value throughout the document, but can be overwritten for each listing individually, or globally using `\qolstset`.

The macro `\qolstset` takes one mandatory argument, which should be a comma-separated series of key-value pairs. These can again be found in section 2.3, this time in `yellow`. If `lst options`, `lst early options`, or `mdframed options` are to be set globally, this should be done using the `\qolstset` macro.

## 2.1 The `\qoinputlisting` Command

This command is analogous to the `\lstinputlisting` command provided by the `listings` package. It takes one mandatory argument, which should be the name of the file to include, and one optional argument which should be a series of comma-separated key-value pairs (parsed by the `keyval` package). The list of optional arguments can be found in section 2.3.

## 2.2 The `qolisting` Environment

This environment is analogous to the `listing` environment provided by the `listings` package. It takes one optional argument, which should be a series of comma-separated key-value pairs. The list of optional arguments can be found in section 2.3.

## 2.3 Optional Arguments

The optional arguments for the commands in the `\qolistings` package are listed here in yellow, followed by the allowable values, then the default value. A value of “-” indicates that the key can be given with no values, in which case it is given a value of “true”. The corresponding package options, where appropriate, are given in blue.

`lst options` *any* `{}`

This option takes as its argument a list of key-value pairs to be passed to the `listings` package, which should be encased in single braces only (for example, `lst options = {stringstyle = \color{red}, commentstyle = \tiny}`). Encasing these in double braces will cause errors as this will not be correctly expanded before being passed to the `keyval` package by `listings`.

These key-value pairs must also be valid options for a style in the `listings` package. These are applied **after** the default style (and any other options), and so if a key is given a value in both `lst options` and the default style, the value given in `lst options` is used.

`lst early options` *any* `{}`

This option is almost identical to the `lst options` option, except that it is applied **before** the default style (and subsequently before `lst options`), and so key-value pairs given in `lst early options` have the lowest priority. This doesn’t currently have many direct applications, but should allow for more compatibility should this package be expanded.

`style` *any* `qolistingsmain`  
`style`

The style to be used for listings. This must be a valid name for a defined style. If `style` is passed as a package argument, then the style must be defined before the first listing. The default style of `qolistingsmain` is the style defined in the `qolistings` package.

`latex comments` *-, true, false* `false`  
`latexcomments`

If true, then comments encased by the `escapeinside` tokens are rendered as normal L<sup>A</sup>T<sub>E</sub>X code rather than printed verbatim. By default, the escape tokens are “(” and “)”, though these can of course be changed by passing `{escapeinside = {<token1>}{<token2>}}` to the `lst options` key.

<code>latex maths</code>	<i>-, true, false</i>	<i>false</i>
<code>latexmaths</code>		
<code>latex math</code>	<i>-, true, false</i>	<i>false</i>
<code>latexmath</code>		

If true (either with the U.S. or U.K. spelling), anything encased in dollar signs will be rendered as L<sup>A</sup>T<sub>E</sub>X maths rather than printed verbatim. Note that this applies to the entire file, not just to comments, and is independent of the `latex comments` option. This means that any \$ anywhere in the file must be accompanied by a corresponding closing \$, and the contents **must** be valid input for a L<sup>A</sup>T<sub>E</sub>X maths environment. (Be especially careful of escape sequences! Even strings written to be valid L<sup>A</sup>T<sub>E</sub>X when printed or passed to plotting tools may not be if backslashes need to be escaped.)

<code>latex</code>	<i>-, true, false</i>
<code>latex</code>	

This flag simply sets both `latex comments` and `latex maths` to its own value. This is processed before the `latex comments` and `latex maths` options, so their values will override the `latex` flag.

<code>first line</code>	<i>positive integer</i>	<i>1</i>
-------------------------	-------------------------	----------

For `\qoinputlisting`, the first line of the file to print. This option has no effect on the `qolisting` environment.

<code>last line</code>	<i>positive integer</i>	<i>9999</i>
------------------------	-------------------------	-------------

For `\qoinputlisting`, the last line of the file to print. This option has no effect on the `qolisting` environment.

<code>first number</code>	<i>positive integer</i>	<i>1</i>
---------------------------	-------------------------	----------

The line number from which to start counting.

<code>mdframed options</code>	<i>any</i>	<i>{}</i>
-------------------------------	------------	-----------

This option takes as its argument a list of key-value pairs to be passed to the `mdframed` package. Again, these should be encased in a single brace only.

These key-value pairs must also be valid options for a style in the `mdframed` package. These are applied last, after any other key-value pairs are handled for the `mdframed` environment.

<code>background</code>	<i>-, true, false</i>	<i>false</i>
<code>background</code>		

If true, the listing is given a background colour of `BackgroundColour`.

rounded corners  
roundedcorners

-, true, false

false

If true, the listing is given rounded corners with radius 0.5em. This has no effect if both **background** and **frame** are false.

frame  
frame

-, true, false, ltrb

false

This option handles the frame of the listing. If true, then the listing is given a frame on all four sides. This option can also take any combination of “l”, “t”, “r”, and “b”, which correspond to a left, top, right, and bottom edge frame. Any combination of these (in any order) will draw a frame on each edge given. For example, **frame** = **blr** will draw a frame on all but the top edge. If **frame** is passed as a package argument, its value **cannot** be left blank; you must pass **frame** = **true** (or similar).

Also note that no checks are in place to prevent unacceptable values from being passed to **frame**; with the exception of the words “true” and “false”, any string passed as the value of **frame** will be searched for an instance of the letters “l”, “t”, “r”, and “b”, and frames will be rendered accordingly. For example, accidentally passing the value “fasle” will result in a single edge drawn on the left of the listing.

### 3 Examples

For the purposes of examples, a python file **exampleCode.py** is used. This file consists of, verbatim and without formatting,

```
1 import example # This will be a normal comment in all QoListings styles
2
3 def exampleFunc(arg1): # (*This will be rendered in \LaTeX{} in appropriate styles*)
4     # Only some of this comment will be (*rendered in \LaTeX*), but the rest will be
5     normal
6     return arg1.upper()
7     # For some styles, maths such as  $\int\limits_0^{\infty} \text{e}^{-x^2} \text{d}x$ 
8     # will be rendered in (*\LaTeX*) regardless of whether the rest of the comment is.
9
10 print("This can cause (*problems*).")
11 print(r"The maths in this string will be rendered in LaTeX even though it's not a
12     comment:  $\int\limits_0^{\infty} \text{e}^{-x^2} \text{d}x$ ")
```

All the following examples use the `\qoinputlisting` command, but exactly the same behaviour would be achieved using the `qolisting` environment and directly inputting the contents of the file; they will (in theory) be rendered identically.

Before the following examples, we will use `\qolstset` to set the language to Python:

```
1 \qolstset{lst options = {language = Python}}
```

With no other options set, `qolisting` will render the code as the following:

```
1 \qoinputlisting{exampleCode.py}
```

```
1  IMPORT example  # This will be a normal comment in all  QoListings styles
2
3  DEF exampleFunc(arg1):  # (*This will be rendered in \LaTeX{} in appropriate styles*)
4      # Only some of this comment will be (*rendered in \LaTeX*), but the rest will be
      normal
5      RETURN arg1.upper()
6      # For some styles, maths such as  $\int\limits_0^{\infty} \text{e}^{-x^2} \text{d}x$ 
      will be rendered in (*\LaTeX*) regardless of whether the rest of the comment is.
7
8  PRINT("This can cause (*problems*).")
9  PRINT(r"The maths in this string will be rendered in LaTeX even though it's not a comment:
       $\int\limits_0^{\infty} \text{e}^{-x^2} \text{d}x$ ")
```

Note that with no frame or background, the listing is aligned on the left with the main body text. When a frame or background is specified, the frame is instead aligned at the margins with the main body text, and the margin of the line numbers is adjusted such that their position remains unchanged

We may add a background and frame quite easily:

```
1 \qoinputlisting[background, frame]{exampleCode.py}
```

```
1  IMPORT example  # This will be a normal comment in all  QoListings styles
2
3  DEF exampleFunc(arg1):  # (*This will be rendered in \LaTeX{} in appropriate styles*)
4      # Only some of this comment will be (*rendered in \LaTeX*), but the rest will be
      normal
5      RETURN arg1.upper()
6      # For some styles, maths such as  $\int\limits_0^{\infty} \text{e}^{-x^2} \text{d}x$ 
      x$ will be rendered in (*\LaTeX*) regardless of whether the rest of the comment is.
7
8  PRINT("This can cause (*problems*).")
9  PRINT(r"The maths in this string will be rendered in LaTeX even though it's not a
      comment:  $\int\limits_0^{\infty} \text{e}^{-x^2} \text{d}x$ ")
```

We may also wish to add rounded corners:

```
1 \qoinputlisting[background, frame, rounded corners]{exampleCode.py}
```

```
1  IMPORT example # This will be a normal comment in all QoListings styles
2
3  DEF exampleFunc(arg1): # (*This will be rendered in \LaTeX{} in appropriate styles*)
4      # Only some of this comment will be (*rendered in \LaTeX*), but the rest will be
      normal
5      RETURN arg1.upper()
6      # For some styles, maths such as  $\int\limits_0^\infty \text{e}^{-x^2} \text{d}x$ 
      x$ will be rendered in (*\LaTeX*) regardless of whether the rest of the comment is.
7
8  PRINT("This can cause (*problems*).")
9  PRINT(r"The maths in this string will be rendered in LaTeX even though it's not a
      comment:  $\int\limits_0^\infty \text{e}^{-x^2} \text{d}x$ ")
```

We can also use the frame options to convincingly split a file and add some commentary in the middle:

```
1 \qoinputlisting[background, frame = ltr, rounded corners, first line = 1, last line =
      6]{exampleCode.py}
2 \lipsum[1]
3 \qoinputlisting[background, frame = ltr, rounded corners, first line = 7, first number =
      7]{exampleCode.py}
```

```
1  IMPORT example # This will be a normal comment in all QoListings styles
2
3  DEF exampleFunc(arg1): # (*This will be rendered in \LaTeX{} in appropriate styles*)
4      # Only some of this comment will be (*rendered in \LaTeX*), but the rest will be
      normal
5      RETURN arg1.upper()
6      # For some styles, maths such as  $\int\limits_0^\infty \text{e}^{-x^2} \text{d}x$ 
      x$ will be rendered in (*\LaTeX*) regardless of whether the rest of the comment is.
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

```
7
8  PRINT("This can cause (*problems*).")
9  PRINT(r"The maths in this string will be rendered in LaTeX even though it's not a
      comment:  $\int\limits_0^\infty \text{e}^{-x^2} \text{d}x$ ")
```



Note that it is important to use the `qolisting`s options `first line`, `last line`, and `first number` instead of passing the corresponding options `firstline`, `lastline`, and `firstnumber` to `listings` via `lst options`. **This will not work**, as `listings` will ignore these options when processing the key-value pairs. Unfortunately, `first line` and `last line` will have no effect in the `qolisting` environment, though it is difficult to imagine a situation in which these would be necessary.

Now let's explore escaping to  $\text{\LaTeX}$ . We can use the options `latex comments` to render anything contained within the default escape tokens (`(*` and `*)`) as  $\text{\LaTeX}$ . This can be quite handy for describing formulae used in code.

```
1 \qoinputlisting[background, frame, rounded corners, latex comments]{exampleCode.py}
```

```
1  IMPORT example # This will be a normal comment in all QoListings styles
2
3  DEF exampleFunc(arg1): # This will be rendered in \text{\LaTeX} in appropriate styles
4      # Only some of this comment will be rendered in \text{\LaTeX}, but the rest will be normal
5      RETURN arg1.upper()
6      # For some styles, maths such as  $\int_0^\infty \text{e}^{-x^2} \text{d}x$  will be rendered in \text{\LaTeX} regardless of whether the rest of the comment is.
7
8  PRINT("This can cause problems.")
9  PRINT(r"The maths in this string will be rendered in \text{\LaTeX} even though it's not a
    comment:  $\int_0^\infty \text{e}^{-x^2} \text{d}x$ ")
```

Alternatively, we can choose to escape to  $\text{\LaTeX}$  only for maths using the `latex maths` (the U.S. spelling `latex math` is also allowed).

```
1 \qoinputlisting[background, frame, rounded corners, latex maths]{exampleCode.py}
```

```
1  IMPORT example # This will be a normal comment in all QoListings styles
2
3  DEF exampleFunc(arg1): # (*This will be rendered in \LaTeX{} in appropriate styles*)
4      # Only some of this comment will be (*rendered in \LaTeX*), but the rest will be
    normal
5      RETURN arg1.upper()
6      # For some styles, maths such as  $\int_0^\infty \text{e}^{-x^2} \text{d}x$  will be rendered in (*\LaTeX*) regardless
    of whether the rest of the comment is.
7
8  PRINT("This can cause (*problems*).")
9  PRINT(r"The maths in this string will be rendered in \LaTeX even though it's not a
    comment:  $\int_0^\infty \text{e}^{-x^2} \text{d}x$ ")
```

If we wish to render both comments and maths in  $\text{\LaTeX}$ , we could specify both `latex comments` and `latex maths`, or we could simply pass the option `latex`:

```
1 \qoinputlisting[background, frame, rounded corners, latex]{exampleCode.py}
```

```
1  IMPORT example # This will be a normal comment in all QoListings styles
2
3  DEF exampleFunc(arg1): # This will be rendered in LATEX in appropriate styles
4      # Only some of this comment will be rendered in LATEX, but the rest will be normal
5      RETURN arg1.upper()
6
7      # For some styles, maths such as  $\int_0^{\infty} e^{-x^2} dx$  will be rendered in LATEX regardless of
8      whether the rest of the comment is.
9
10     PRINT("This can cause problems.")
11     PRINT(r"The maths in this string will be rendered in LaTeX even though it's not a
12         comment:  $\int_0^{\infty} e^{-x^2} dx$ ")
```

Let's suppose we want to highlight a particular listing by changing the frame colour. We can do this by passing the relevant options to `mdframed`:

```
1 \qoinputlisting[background, frame, rounded corners, mdframed options = {linecolor =
   Accent3}]{exampleCode.py}
```

```
1  IMPORT example # This will be a normal comment in all QoListings styles
2
3  DEF exampleFunc(arg1): # (*This will be rendered in \LaTeX{} in appropriate styles*)
4      # Only some of this comment will be (*rendered in \LaTeX*), but the rest will be
5      normal
6      RETURN arg1.upper()
7
8      # For some styles, maths such as  $\int_0^{\infty} e^{-x^2} dx$  will be rendered in
9      (*\LaTeX*) regardless of whether the rest of the comment is.
10
11     PRINT("This can cause (*problems*).")
12     PRINT(r"The maths in this string will be rendered in LaTeX even though it's not a
13         comment:  $\int_0^{\infty} e^{-x^2} dx$ ")
```

Another point worthy of note is that, since both `mdframed` and `listings` packages handle page breaks well, `qolisting`s can also break in a consistent and well-behaved manner at page breaks.

```
1 \qoinputlisting[background, frame, rounded corners, mdframed options = {linecolor =
   Accent3}]{exampleCode.py}
```

```
1  IMPORT example # This will be a normal comment in all QoListings styles
2
3  DEF exampleFunc(arg1): # (*This will be rendered in \LaTeX{} in appropriate styles*)
4      # Only some of this comment will be (*rendered in \LaTeX*), but the rest will be
5      normal
6      RETURN arg1.upper()
```

```
6      # For some styles, maths such as  $\int\limits_0^{\infty} \text{e}^{-x^2} \text{d}x$ 
      x$ will be rendered in ( $\text{\LaTeX}$ ) regardless of whether the rest of the comment is.
7
8  PRINT("This can cause (*problems*).")
9  PRINT(r"The maths in this string will be rendered in LaTeX even though it's not a
      comment:  $\int\limits_0^{\infty} \text{e}^{-x^2} \text{d}x$ ")
```