`qolistings.sty` Package

Peter Brookes Chambers

December 18, 2020

Contents

1 Introduction

This package builds on the `listings` package to provide appealing and professional default styles, integrating the superior background and frame handling of the `mdframed` package. The aim is for these styles to be easily adapted to match the surrounding document, with simple options for common requirements. The QoListings package provides little additional functionality; instead, the emphasis is on making clear, aesthetically appealing listings quicker and easier to include.

The QoListings package is designed to incorporate the colour schemes defined by the QoColours package, such that if the QoColours package is used in the same document then the colours used in the default listing style of QoListings will be chosen from the QoColours scheme. However, the QoColours package is not necessary; if the QoColours package is not used, then QoListings defines a scheme identical to the default `twilight` scheme of QoColours.

2 Package Requirements

The following packages are each required for the QoListings package.

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

In addition, the following packages are required **if and only if** coloured inline backgrounds are used (see section 3.1).

- `xpatch`
- `realboxes`

3 Usage

3.1 Loading the Package

The package can be loaded with `\usepackage{qolistings}`. It takes a variety of optional arguments in a comma-separated list of key-value pairs. Most of these correspond to options which can also be set locally, and are given in section 3.3 in blue. However, there are two options which can only be passed to the package; `caps` (and the equivalent opposite `nocaps`), and `inlinebackgrounds`. Passing the options `caps` modifies the default listing style to typeset keywords in the `\scshape` style (for example, Small Capitals Shape). The default `nocaps` does not use

small capitals. The `inlinebackgrounds` option (which is true by default) uses `xpatch` to apply a colour box to all inline listings. To prevent this (and subsequently not require the `xpatch` and `realboxes` packages), one must pass `inlinebackgrounds = false` when loading the package.

All other options are given in 3.3, with any differences in behaviour to the local options noted there.

3.2 Macros and Environments

`\qolstset{}`

This macro takes one mandatory argument, which should be a comma-separated series of key-value pairs. Appropriate pairs can be found in section 3.3 in yellow. These are then set globally; any options passed to `\qolstset{}` will be applied to all subsequent listings in the document, unless overwritten (locally or globally).

`\qoinputlisting[]{}`

This macro is analogous to the `\lstinoutlisting` macro from the `listings` package. It takes one optional argument and one mandatory argument. The mandatory argument should be a path to the file to be inlcuded, and the optional argument should again be a comma-separated series of key-value pairs, given in section 3.3 in yellow. These are only applied locally to that single listing.

`qolisting` Environment

This environment is analogous to the `lstlisting` environment from the `listings` package. When beginning the environment, one optional argument can be passed to the environment. Again, this should be a comma-separated series of key-value pairs, given in section 3.3 in yellow. These are only applied locally to that single listing. Any text within the environment is then typeset verbatim as a listing.

`\qoinline||`

This macro is analogous to the `\lstinline||` macro, and typesets any text between the two delimiter tokens as a listing inline with the surrounding text. Optional arguments can be passed to this macro, however these must be valid key-value pairs for the `\lstinline` macro (**not** those given in section 3.3). If the package option `inlinebackgrounds` is true, then the inline listing will have a coloured background. Note that this will not linewrap long inline listings.

3.3 Key-Value Pairs

The following are the allowable key-value pairs for most QoListings macros and environments. The key (local and global where available) is given on the left, allowable values in the centre, and the default value on the right. A dash ("-") indicates that the key can be passed with no value, in which case it will be given a value of *true*.

`lst options`	*any*	{}

This option takes as its value any string which would be valid for the `\lstset` macro from the `listings` package. This must be surrounded by single braces only. For example,

```
lst options = {stringstyle = \color{red}, language = Python}
```

would be acceptable, but

```
lst options = {{stringstyle = \color{red}, language = Python}}
```

would not. These options are applied last, after the default style, and so can be used to overwrite anything defined in other options. The global options (set with `qolstset{}`) are applied, followed by the local options.

`lst early options` *any* {}

This option is identical to `lst options`, except that they are applied first, before the default style. Again, the global options (set with `qolstset{}`) are applied, followed by the local options.

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

This option defines the style to be applied to the listing. This can be any style defined with the `\lstdefinestyle` macro from the `listings` package, or the default style `qolistingsmain`. When this option is set using the `\qolstset` command or in the optional argument of the `\qoinputlisting` or `qolisting` environment, the style must have already been defined. When this option is passed to the package, the style must be defined before the first listing.

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

When true, this option causes any text encased in the escape delimiters (by default, these are `(*` and `*)`, but this can be changed using the apptopriate options in the `listings` package) to be rendered as LaTeX rather than being preinted verbatim. This is intended to be used for code comments, but unfortunately is not restricted to comments, and so any text encased in the escape delimiters will be rendered as LaTeX, and so **must** be valid LaTeX syntax.

`latex maths` *-, true, false* *false*
`latexmaths`
`latex math` *-, true, false* *false*
`latexmath`

When true (either in the U.K. spelling or the U.S. spelling), any text encased in dollar signs will be rendered as LaTeX maths rather than printed vevrbatim. Again, this is not restricted to comments, and so any text encased in dollar signs will be rendered as LaTeX maths, and so must be valid LaTeXsyntax.

`latex` *-, true, false* *false*
`latex`

When true, this option simply sets `latex comments` and `latex maths` to also be true.

`first line` *positive integer* *1*

This option only has an effect for the `\qoinputlisting` command. For all other commands and environments (including `\qolstset`), this will not cause an error but will have no effect. For `\qoinputlisting`, this is the first line of the file to be printed.

`last line` *positive integer* *9999*

This option only has an effect for the `\qoinputlisting` command. For all other commands and environments

(including `\qolstset`), this will not cause an error but will have no effect. For `\qoinputlisting`, this is the last line of the file to be printed.

`first number` *integer* *1*

This option only has an effect for the `\qoinputlisting` command. For all other commands and environments (including `\qolstset`), this will not cause an error but will have no effect. For `\qoinputlisting`, this is the first line number of the listing.

`mdframed options` *any* {}

This option takes as its value any string which would be valid for the `\mdframedsetup` macro from the `mdframed` package. This must be surrounded by single braces only. These options are applied last, after any other options, and so can be used to overwrite anything defined in other options. The global options (set with `qolstset{}`) are applied first, followed by the local options.

`background` *-, true, false* *false*
`background`

When true, the listing is given a coloured background using `mdframed`.

`frame` *-, true, false, ltrb* *false*
`frame`

When true, a frame will be drawn around the listing. Instead of passing `true` or `false`, a string can be passed which contains some combination of the letters `l`, `t`, `r`, and `b` (currently these must be lowercase). For each letter present an edge will be drawn on the left, top, right, and bottom of the listing respectively. For example, `frame = tb` would draw an edge at the top and bottom of the listing.

Note that currently there is no check for strings containing additional letters; that is, any string which is not exactly "`true`" or "`false`" will be searched for any of the letters `l`, `t`, `r`, and `b`. This can lead to false positives; passing the misspelling `frame = fasle` would draw a line along the left edge.

`rounded corners` *-, true, false* *false*
`roundedcorners`

If true, then the frame and background (if present) will be drawn with rounded corners. By default, these will have a radius of 0.5 em, but this can be changed using the `mdframed options` key.

4 Examples

For all of the following examples, the language Python will be used:

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

The LaTeX code is given in an orange border, while the resultant listing is shown on below.

4.1 Default

```
\begin{qolisting}
import example  # This will be a normal comment in all  QoListings styles

def exampleFunc(arg1):  # (*This will be rendered in \LaTeX{} in appropriate styles*)
    # Only some of this comment will be (*rendered in \LaTeX*), but the rest will be
   normal
    return arg1.upper()
    # 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.

print("This can cause (*problems*).")
print(r"The maths in this string will be rendered in LaTeX even though it's not a
   comment: $\int\limits_{0}^{\infty} \text{e}^{-x^2} \text{d}x$")
\end{qolisting}
```

```
import example  # This will be a normal comment in all  QoListings styles

def exampleFunc(arg1):  # (*This will be rendered in \LaTeX{} in appropriate styles*)
    # Only some of this comment will be (*rendered in \LaTeX*), but the rest will be
   normal
    return arg1.upper()
    # 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.

print("This can cause (*problems*).")
print(r"The maths in this string will be rendered in LaTeX even though it's not a comment:
    $\int\limits_{0}^{\infty} \text{e}^{-x^2} \text{d}x$")
```

4.2 lst options

```
\begin{qolisting}[lst options = {stringstyle = \color{Accent5}}]
import example  # This will be a normal comment in all  QoListings styles

def exampleFunc(arg1):  # (*This will be rendered in \LaTeX{} in appropriate styles*)
    # Only some of this comment will be (*rendered in \LaTeX*), but the rest will be normal
    return arg1.upper()
    # 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.

print("This can cause (*problems*).")
print(r"The maths in this string will be rendered in LaTeX even though it's not a comment: $\int\limits_{0}^{\infty} \text{e}^{-x^2} \text{d}x$")
\end{qolisting}
```

```
import example  # This will be a normal comment in all  QoListings styles

def exampleFunc(arg1):  # (*This will be rendered in \LaTeX{} in appropriate styles*)
    # Only some of this comment will be (*rendered in \LaTeX*), but the rest will be normal
    return arg1.upper()
    # 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.

print("This can cause (*problems*).")
print(r"The maths in this string will be rendered in LaTeX even though it's not a comment: $\int\limits_{0}^{\infty} \text{e}^{-x^2} \text{d}x$")
```