Rainbow Brackets
Documentation for the LaTeX package 'rainbowbrackets'

rainbowbrackets \v 1.2.1
Paul Eduard Koenig
Goethe University Frankfurt,
Institute of Linguistics
pauleduardkoenig@gmail.com

July 20, 2025

Abstract

This document presents a comprehensive overview of the LaTeX package rainbowbrackets. The primary function of this package is to replicate a common feature found in many integrated development environments (IDEs), wherein matching parentheses at the same nesting level are assigned corresponding colors. This visual aid facilitates improved readability and cognitive parsing of complex expressions.

Contents

1 Background

The rainbow brackets (or colored parentheses) feature originates from modern integrated development environments (IDEs), where it is commonly used to visually distinguish matching pairs of brackets, braces, and parentheses by applying distinct colors at each level of nesting. This technique is especially prevalent in programming and technical writing, where complex, deeply nested structures are common. By assigning consistent colors to matching delimiters, it enhances code readability and reduces the cognitive load required to trace nested scopes, thereby helping users quickly identify structural errors or imbalances. Its integration into LaTeX through the rainbowbrackets package brings these benefits to mathematical and technical documents, supporting clearer comprehension of intricate formulas and expressions.

The concept was first used in the `fragoli` package, where it was employed primarily for educational purposes (the code used there originated from an intial implementation by Matthew Towers, see Section 7). Specifically, it was used to support the teaching of complex semantic derivations in formal linguistics and logic, where deeply nested structures are common.

Due to the usefulness and growing demand for bracket colorization beyond its original pedagogical context in the `fragoli` package, the feature was separated into a standalone package, `'rainbowbrackets'`. This modularization allows the functionality to be reused across a broader range of LaTeX projects, independent of the specific goals of `fragoli`. In its new form, `'rainbowbrackets'` introduces additional features such as multiple coloring styles, improved customization options, and enhanced compatibility. Moreover,

isolating the feature into its own package facilitates better maintenance, easier debugging, and more focused development, ultimately making it a more robust and flexible tool for both educational and technical typesetting needs.

2 Examples

As an example see:
`\begin{rb}(level1(level2(level3)))\end{rb}` which results in: (level1(level2(level3))). To change the colors one needs to manually override the colors *rbbcpcolor0* to *rbbcpcolor9* like this:
`\definecolor{rbbcpcolor9}{RGB}{000, 000, 000}`.

Default color scheme:
(level0(level1((level3(level4(level5(level6(level7(level8(level9)))))))level2)))
Default color scheme max three colors:
(level0(level1((level3(level4(level5(level6(level7(level8(level9)))))))level2)))
Default color scheme bold:
(level0(level1((level3(level4(level5(level6(level7(level8(level9)))))))level2)))
Neon color scheme bold:
(level0(level1((level3(level4(level5(level6(level7(level8(level9)))))))level2)))
Pastel color scheme bold:
(level0(level1((level3(level4(level5(level6(level7(level8(level9)))))))level2)))
Replace all parentheses with curly brackets:
`\begin{rb}[\ensuremath{\{}][\ensuremath{\}}](level1(level2(level3)))\end{rb}`
{level1{level2{level3}}}.
Change brackets to curly brackets half way with custom local rbracket:
`\begin{rb}[\rbparenl][\ensuremath{>}](level1(level2\rbsetlbracket{\ensuremath{\{}}`
`\rbsetrbracket{\ensuremath{\}}}(level3)\rbresetbrackets))\end{rb}`
(level1(level2{level3}>>.
The final example illustrates a workaround necessitated by the package's method of parsing optional parameters. Specifically, it is not possible to directly use '()' as local custom bracket symbols, as the environment interprets them as commands, disrupting the argument parsing process. To circumvent this limitation, '()' are instead stored in the macros `\rbparenl` and `\rbparenr`. This approach enables the use of '()' as local bracket symbols within the *rb* environment(s).

3 Usage

To use the **'rainbowbrackets'** package place the 'rainbowbrackets.sty' file in the same folder as your document and include:

```
\usepackage{rainbowbrackets}
```

3.1 Package Options

To combine multiple package options use ',' like:

```
\usepackage[max=5, style=neon]{rainbowbrackets}
```

3.1.1 style

```
\usepackage[style=STYLE]{rainbowbrackets}
```

This option will effect the default color scheme. The default ist set to **'default'**, valid options are:

- default ((((((((((()))))))))))
- neon ((((((((((()))))))))))
- pastel ((((((((((()))))))))))
- spectral ((((((((((()))))))))))
- dualpolar ((((((((((()))))))))))
- warning ((((((((((()))))))))))

3.1.2 max

```latex
\usepackage[max=VALUE]{rainbowbrackets}
```

This option will define at which level the coloring starts repeating. The minimum is 2, the maximum is 10. The default ist set to '10'.

- 2 ((((((((((()))))))))))
- 4 ((((((((((()))))))))))
- 7 ((((((((((()))))))))))

3.1.3 lbracket/rbracket

```latex
\usepackage[lbracket={LEFTBRACKETSYMBOL}, rbracket={RIGHTBRACKETSYMBOL}]{rainbowbrackets}
```

This option(s) will define the global bracket symbols that will be used to replace the parenthesis in an *rb* environment. It is possible to change only one the left or the right bracket symbol individually. The defaults are set '(' and ')' respectively.

4 Commands

This section lists all commands.

(1)

```latex
\begin{rb}[lbracket][rbracket]
   content...
\end{rb}
```

Arguments:

lbracket(def='('): The left bracket symbol.
rbracket(def=')'): The left bracket symbol.

Description: Parentheses on the same nesting level will receive the same color. Setting *lbracket* and *rbracket* to custom values will replace every () with the passed values. To set the bracket arguments to '()' use `\rbparenl` and `\rbparenr` respectively (see Section 2 for details).

Example: (level0(level1((level3(level4(level5(level6(level7(level8(level9)))))))))level2)))

(2)

```latex
\begin{rbb}[lbracket][rbracket]
   content...
\end{rbb}
```

Arguments:

lbracket(def='('): The left bracket symbol.
rbracket(def=')'): The left bracket symbol.

Description: Same as *rb* environment but all parentheses are in bold mode. Setting *lbracket* and *rbracket* to custom values will replace every () with the passed values. To set the bracket arguments to '()' use `\rbparenl` and `\rbparenr` respectively (see Section 2 for details).

Example: **(**level0(level1((level3(level4(level5(level6(level7(level8(level9)))))))))level2)))

(3)

```latex
\begin{rbi}[lbracket][rbracket]
   content...
\end{rbi}
```

Arguments:

lbracket(def='('): The left bracket symbol.
rbracket(def=')'): The left bracket symbol.

Description: Same as *rb* environment but all parentheses are in italic mode. Setting *lbracket* and *rbracket* to custom values will replace every () with the passed values. To set the bracket arguments to '()' use **\rbparenl** and **\rbparenr** respectively (see Section 2 for details).

Example: *(*level0*(*level1*((*level3*(*level4*(*level5*(*level6*(*level7*(*level8*(*level9*)))))))*level2*)))*

(4)

```
1  \rbsetnormalmode
2  \rbsetboldmode
3  \rbsetitalicmode
4  \rbresetmode
```

Description: To change the text mode within a *rb* environment, use one of the commands. To reset the mode to the environment default use **\rbresetmode**. To change the text mode to a custom mode within an *rb* environment the transformer command has to be renewed **\renewcommand{\rbtransformer}[1]{#1}**. The transformer will be resettet each time a new *rb* environment is used.

Example: **\begin{rbi}(level0(level1((level3(level4\rbsetboldmode(level5(level6))\rbresetmode))level2)))\end{rbi}**
Will produce: *(*level0*(*level1*((*level3*(*level4**(**level5**(**level6**)))***)**level2*)))*

Example:

\begin{rbi}(level0(level1(\renewcommand{\rbtransformer}[1]{\underline{#1}}(level5(level6))\rbresetmode)))\end{rbi}

Will produce: *(*level0*(*level1*((*level5*(*level6*)))))*

(5)

```
1  \rbsetbracketmax{value}
```

Arguments:

   value: The maximal coloring level

Description: Changes the maximaal coloring level mid document. Allowed are integers from 2 to 10.

(6)

```
1  \rbresetbracketmax
```

Description: Resets the max coloring level to the package default.

(7)

```
1  \rbsetstyle{style}
```

Arguments:

   style: The color scheme to be used

Description: Changes the color scheme mid document to the specified style. For a list of styles see Section 3.1.1. If the specified style is unknown the default style is selected.

(8)

```
1  \rbresetstyle
```

Description: Resets the color scheme to the package default.

(9)

```
1  \definecolor{rbbcpcolor0}{HTML}{value}
2  \definecolor{rbbcpcolor1}{HTML}{value}
3  \definecolor{rbbcpcolor2}{HTML}{value}
4  \definecolor{rbbcpcolor3}{HTML}{value}
5  \definecolor{rbbcpcolor4}{HTML}{value}
6  \definecolor{rbbcpcolor5}{HTML}{value}
7  \definecolor{rbbcpcolor6}{HTML}{value}
8  \definecolor{rbbcpcolor7}{HTML}{value}
9  \definecolor{rbbcpcolor8}{HTML}{value}
10 \definecolor{rbbcpcolor9}{HTML}{value}
```

Arguments:

> `value`: The color code in hex.

Description: To change the color scheme to a custom scheme, override these colors. These commands are defined by the xcolor package.

(10)

```
1  \rbdisable
```

Description: Disables coloring mid document.

(11)

```
1  \rbenable
```

Description: Enables coloring again, sets color scheme to last selected style.

(12)

```
1  \rbsetlbracket{symbol}
2  \rbsetrbracket{symbol}
```

Arguments:

> `symbol`: The new bracket symbol. (Use `\ensuremath{text}` for symbols like {})

Description: Change the bracket symbol mid document. If used outside an rb environment it will change the global bracket symbol, if used within a rb environment it will only change the symbols within the current environment.

(13)

```
1  \rbresetbrackets
2  \rbresetlbracket
3  \rbresetrbracket
```

Description: Resets the bracket symbol(s). If used outside an rb environment it will reset the global bracket symbol(s) to the package default(s) (or the one(s) passed via the package arguments), if used within a rb environment it will reset them to the current environment default(s).

5 Known bugs and limitations

Since curly brackets {} are integral to the LaTeX syntax, automatic coloring of these brackets is likely infeasible. Similar constraints apply to square brackets [], due to their restriction to math mode. Solutions to these limitations are welcome.

Known bugs:

- No coloring when parentheses are generated via another macro within the *rb* environment (getting the correct order of expansion to work as intended seems to be complicated... again a solution is welcome). Example:
  \newcommand{\test}{(level2test)}
  \begin{rb}(level0(level1\test (level2)))\end{rb}
  Will render as: (level0(level1(level2test)(level2)))

6 Changelog

1.2.1   - Renamed some commands to get a cohesive command style (now all start with rb).

- Add optional package arguments to change bracket symbols.

- Add commands to change bracket symbols.

- Add new commands to temporarily disable coloring.

- Add new commands to temporarily change the text mode of the brackets.

- Add new italic environment.

1.1.1   - Add new styles

- Refactored style system.

- Fix typos.

- Added missing credits!

1.0.0   - Initial release.

7 Acknowledgments

This package builds upon foundational work by Ryan Reich and jub0bs, whose implementations for coloring nested parentheses in TeX (as shared on TeX StackExchange Contribution A and Contribution B) provided key conceptual and technical insights. Matthew Towers subsequently combined their approaches to produce an initial implementation of latex-rainbow, which this package further develops.