

Package ‘highr’

July 22, 2025

Type Package

Title Syntax Highlighting for R Source Code

Version 0.11

Description Provides syntax highlighting for R source code. Currently it supports LaTeX and HTML output. Source code of other languages is supported via Andre Simon's highlight package (<<https://gitlab.com/saalen/highlight>>).

Depends R (>= 3.3.0)

Imports xfun (>= 0.18)

Suggests knitr, markdown, testit

License GPL

URL <https://github.com/yihui/highr>

BugReports <https://github.com/yihui/highr/issues>

VignetteBuilder knitr

Encoding UTF-8

RoxygenNote 7.3.1

NeedsCompilation no

Author Yihui Xie [aut, cre] (ORCID: <<https://orcid.org/0000-0003-0645-5666>>),
Yixuan Qiu [aut],
Christopher Gandrud [ctb],
Qiang Li [ctb]

Maintainer Yihui Xie <xie@yihui.name>

Repository CRAN

Date/Publication 2024-05-26 20:00:03 UTC

Contents

highlight	2
hi_andre	3
Index	5

highlight	<i>Syntax highlight an R code fragment</i>
-----------	--

Description

This function [parses](#) the R code, fetches the tokens in it ([getParseData](#)), and attach syntax highlighting commands onto them. With proper style definitions for these commands (such as colors or font styles), the R code will be syntax highlighted in the LaTeX/HTML output. The two functions `hi_latex` and `hi_html` are wrappers of `highlight` for LaTeX and HTML output, respectively.

Usage

```
highlight(
  code,
  format = c("latex", "html"),
  markup,
  prompt = FALSE,
  fallback = FALSE
)

hi_latex(code, ...)

hi_html(code, ...)
```

Arguments

<code>code</code>	a character string (the R source code)
<code>format</code>	the output format
<code>markup</code>	a data frame of two columns containing the markup commands
<code>prompt</code>	whether to add prompts to the code
<code>fallback</code>	whether to use the fallback method, i.e. the regular expression based method; this method is not precise and only highlights a few types of symbols such as comments, strings and functions; fallback will be set to TRUE when the input code fails to be parsed
<code>...</code>	arguments to be passed to <code>highlight()</code>

Details

For the markup data frame, the first column is put before the R tokens, and the second column is behind; the row names of the data frame must be the R token names; a special row is named `DEFAULT`, which contains the markup for the standard tokens (i.e. those that do not need to be highlighted); if missing, the built-in data frames `highr:::cmd_latex` and `highr:::cmd_html` will be used.

This function only binds markups onto R tokens, and the real syntax highlighting must be done with style definitions, which is out of the scope of this package. It was designed to be used as the syntax highlighting infrastructure of other packages such as **knitr**, where the colors and font styles are properly defined in the LaTeX preamble and HTML header.

Value

A character vector for the syntax highlighted code.

Author(s)

Yihui Xie and Yixuan Qiu

See Also

See the package vignettes `browseVignettes('highr')` for how this function works internally.

Examples

```
library(highr)
highlight("x=1 # assignment")

txt = c("a <- 1 # something", "c(y=\"world\", z=\"hello\")", "b=function(x=5) {",
  "for(i in 1:10) {",
  "  if (i < x) print(i) else break}}", "z@child # S4 slot",
  "'special chars <>$$%&_{}'")
cat(hi_latex(txt), sep = "\n")
cat(hi_html(txt), sep = "\n")

# the markup data frames
highr:::cmd_latex
highr:::cmd_html
```

hi_andre

A wrapper to Andre Simon's Highlight

Description

This function calls `Highlight` to syntax highlight a code fragment.

Usage

```
hi_andre(code, language, format = "html")
```

Arguments

code	a character string of the source code
language	the input language (c, cpp, python, r, ...); see <code>system('highlight -p')</code>
format	the output format (html, latex, ...)

Value

A character string for the syntax highlighted code.

References

Andre Simon's Highlight package <https://gitlab.com/saalen/highlight>.

Examples

```
## Not run:  
hi_andre("1+1", language = "R")  
hi_andre("void main() {\nreturn(0)\n}", language = "c", format = "latex")  
  
## End(Not run)
```

Index

`getParseData`, [2](#)

`hi_andre`, [3](#)

`hi_html (highlight)`, [2](#)

`hi_latex (highlight)`, [2](#)

`highlight`, [2](#)

`parse`, [2](#)