Package 'downlit'

June 10, 2024

Title Syntax Highlighting and Automatic Linking

Version 0.4.4

Description Syntax highlighting of R code, specifically designed for the needs of 'RMarkdown' packages like 'pkgdown', 'hugodown', and 'bookdown'. It includes linking of function calls to their documentation on the web, and automatic translation of ANSI escapes in output to the equivalent HTML.

License MIT + file LICENSE

URL https://downlit.r-lib.org/, https://github.com/r-lib/downlit

BugReports https://github.com/r-lib/downlit/issues

Depends R (>= 4.0.0)

Imports brio, desc, digest, evaluate, fansi, memoise, rlang, vctrs, withr, yaml

Suggests covr, htmltools, jsonlite, MASS, MassSpecWavelet, pkgload, rmarkdown, testthat (>= 3.0.0), xml2

Config/Needs/website tidyverse/tidytemplate

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.1

NeedsCompilation no

Author Hadley Wickham [aut, cre], Posit Software, PBC [cph, fnd]

Maintainer Hadley Wickham <hadley@posit.co>

Repository CRAN

Date/Publication 2024-06-10 09:10:02 UTC

Contents

`autolink` *Automatically link inline code*

Description

Automatically link inline code

Usage

```
autolink(text)

autolink_url(text)
```

Arguments

`text`	String of code to highlight and link.

Value

If `text` is linkable, an HTML link for `autolink()`, and or just the URL for `autolink_url()`. Both return `NA` if the text is not linkable.

Options

downlit provides a number of options to control the details of the linking. They are particularly important if you want to generate "local" links.

- `downlit.package`: name of the current package. Determines when `topic_index` and `article_index`
- `downlit.topic_index` and `downlit.article_index`: named character vector that maps from topic/article name to path.
- `downlit.rdname`: name of current Rd file being documented (if any); used to avoid self-links.
- `downlit.attached`: character vector of currently attached R packages.
- `downlit.local_packages`: named character vector providing relative paths (value) to packages (name) that can be reached with relative links from the target HTML document.
- `downlit.topic_path` and `downlit.article_path`: paths to reference topics and articles/vignettes relative to the "current" file.

Examples

```
autolink("stats::median()")
autolink("vignette('grid', package = 'grid')")

autolink_url("stats::median()")
```

`downlit_html_path` *Syntax highlight and link an HTML page*

Description

- Code blocks, identified by <pre> tags with class `sourceCode r` or any <pre> tag inside of `<div class='downlit'>`, are processed with `highlight()`.
- Inline code, identified by <code> tags that contain only text (and don't have a header tag (e.g. <h1>) or <a> as an ancestor) are processed processed with `autolink()`.

Use `downlit_html_path()` to process an `.html` file on disk; use `downlit_html_node()` to process an in-memory `xml_node` as part of a larger pipeline.

Usage

```
downlit_html_path(in_path, out_path, classes = classes_pandoc())

downlit_html_node(x, classes = classes_pandoc())
```

Arguments

`in_path`, `out_path`	Input and output paths for HTML file
`classes`	A mapping between token names and CSS class names. Bundled `classes_pandoc()` and `classes_chroma()` provide mappings that (roughly) match Pandoc and chroma (used by hugo) classes so you can use existing themes.
`x`	An `xml2::xml_node`

Value

`downlit_html_path()` invisibly returns `output_path`; `downlit_html_node()` modifies `x` in place and returns nothing.

Examples

```
node <- xml2::read_xml("<p><code>base::t()</code></p>")
node

# node is modified in place
downlit_html_node(node)
node
```

`downlit_md_path` *Syntax highlight and link a md document*

Description

`downlit_md_*` works by traversing the markdown AST generated by Pandoc. It applies `highlight()` to `CodeBlocks` and `autolink()` to inline `Code`.

Use `downlit_md_path()` to transform a file on disk; use `downlit_md_string()` to transform a string containing markdown as part of a larger pipeline.

Needs pandoc 1.19 or later.

Usage

```
downlit_md_path(in_path, out_path, format = NULL)

downlit_md_string(x, format = NULL)
```

Arguments

`in_path`, `out_path`	Input and output paths for markdown file.
`format`	Pandoc format; defaults to "gfm" if you have pandoc 2.0.0 or greater, otherwise "markdown_github".
`x`	A string containing markdown.

Value

`downlit_md_path()` invisibly returns `output_path`; `downlit_md_string()` returns a string containing markdown.

Examples

```
if (rmarkdown::pandoc_available("1.19")) {
downlit_md_string("`base::t()`")
downlit_md_string("`base::t`")
downlit_md_string("* `base::t`")

# But don't highlight in headings
downlit_md_string("## `base::t`")
}
```

evaluate_and_highlight *Evaluate code and syntax highlight the results*

Description

This function runs code and captures the output using `evaluate::evaluate()`. It syntax higlights code with `highlight()`, and intermingles it with output.

Usage

```
evaluate_and_highlight(
  code,
  fig_save,
  classes = downlit::classes_pandoc(),
  env = NULL,
  output_handler = evaluate::new_output_handler(),
  highlight = TRUE
)
```

Arguments

`code`	Code to evaluate (as a string).
`fig_save`	A function with arguments `plot` and `id` that is responsible for saving `plot` to a file (using `id` to disambiguate multiple plots in the same chunk). It should return a list with components `path`, `width`, and `height`.
`classes`	A mapping between token names and CSS class names. Bundled `classes_pandoc()` and `classes_chroma()` provide mappings that (roughly) match Pandoc and chroma (used by hugo) classes so you can use existing themes.
`env`	Environment in which to evaluate code; if not supplied, defaults to a child of the global environment.
`output_handler`	Custom output handler for `evaluate::evaluate()`.
`highlight`	Optionally suppress highlighting. This is useful for tests.

Value

An string containing HTML with a `dependencies` attribute giving an additional htmltools dependencies required to render the HTML.

Examples

```
cat(evaluate_and_highlight("1 + 2"))
cat(evaluate_and_highlight("x <- 1:10\nmean(x)"))

# ------------------------------------------------------------------
# evaluate_and_highlight() powers pkgdown's documentation formatting so
```

```
# here I include a few examples to make sure everything looks good
# ---------------------------------------------------------------

blue <- function(x) paste0("\033[34m", x, "\033[39m")
f <- function(x) {
  cat("This is some output. My favourite colour is ", blue("blue"), ".\n", sep = "")
  message("This is a message. My favourite fruit is ", blue("blueberries"))
  warning("Now at stage ", blue("blue"), "!")
}
f()

plot(1:10)
```

`highlight` *Highlight and link a code block*

Description

This function:

- syntax highlights code
- links function calls to their documentation (where possible)
- in comments, translates ANSI escapes in to HTML equivalents.

Usage

```
highlight(text, classes = classes_chroma(), pre_class = NULL, code = FALSE)

classes_pandoc()

classes_chroma()
```

Arguments

`text`	String of code to highlight and link.
`classes`	A mapping between token names and CSS class names. Bundled `classes_pandoc()` and `classes_chroma()` provide mappings that (roughly) match Pandoc and chroma (used by hugo) classes so you can use existing themes.
`pre_class`	Class(es) to give output `<pre>`.
`code`	If `TRUE`, wrap output in a `<code>` block

Value

If `text` is valid R code, an HTML `<pre>` tag. Otherwise, `NA`.

A string containing syntax highlighted HTML or `NA` (if `text` isn't parseable).

Options

downlit provides a number of options to control the details of the linking. They are particularly important if you want to generate "local" links.

- `downlit.package`: name of the current package. Determines when `topic_index` and `article_index`
- `downlit.topic_index` and `downlit.article_index`: named character vector that maps from topic/article name to path.
- `downlit.rdname`: name of current Rd file being documented (if any); used to avoid self-links.
- `downlit.attached`: character vector of currently attached R packages.
- `downlit.local_packages`: named character vector providing relative paths (value) to packages (name) that can be reached with relative links from the target HTML document.
- `downlit.topic_path` and `downlit.article_path`: paths to reference topics and articles/vignettes relative to the "current" file.

Examples

```
cat(highlight("1 + 1"))
cat(highlight("base::t(1:3)"))

# Unparseable R code returns NA
cat(highlight("base::t("))
```

`is_low_change` *Compare two recorded plots*

Description

Compare two recorded plots

Usage

```
is_low_change(p1, p2)
```

Arguments

`p1`, `p2`	Plot results

Value

Logical value indicating whether `p2` is a low-level update of `p1`.

Index