

# Package ‘accessrmd’

July 22, 2025

**Title** Improving the Accessibility of 'rmarkdown' Documents

**Version** 1.0.0

**Description** Provides a simple method to improve the accessibility of 'rmarkdown' documents. The package provides functions for creating or modifying 'rmarkdown' documents, resolving known errors and alerts that result in accessibility issues for screen reader users.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.1.2

**Depends** ggplot2, R (>= 2.10)

**Imports** htmltools, stringr, rlist, knitr, RCurl

**Suggests** testthat, rmarkdown, markdown, covr

**NeedsCompilation** no

**Author** Rich Leyshon [aut, cre],  
Crown Copyright 2021 [cph]

**Maintainer** Rich Leyshon <richard.leyshon@ons.gov.uk>

**Repository** CRAN

**Date/Publication** 2022-05-03 10:00:05 UTC

## Contents

access_head . . . . .	2
access_img . . . . .	3
access_rmd . . . . .	4
assemble_header . . . . .	5
check_compat . . . . .	6
check_urls . . . . .	6
detect_html_lang . . . . .	7
find_all_tags . . . . .	8
find_alt_lim . . . . .	8
find_theme . . . . .	9

handle_rmd_path . . . . .	9
insert_yaml . . . . .	10
retrieve_rmds . . . . .	10
return_heading . . . . .	11
sus_alt . . . . .	12

<b>Index</b>	<b>14</b>
--------------	-----------

---

access_head	<i>Convert YAML to an accessible header</i>
-------------	---

---

## Description

Reads an Rmd file, converting the YAML header to a format that is screen reader friendly.

## Usage

```
access_head(
  rmd_path = NULL,
  lan = detect_html_lang(lines),
  inplace = FALSE,
  encoding = "utf-8",
  theme = "default",
  highlight = "null"
)
```

## Arguments

rmd_path	Path to the Rmd that requires accessible header metadata. Rmd must be output type html.
lan	Identify the language of text content. Attempts to find a lang attribute value from the rmd document. Alternatively, use a character string such as "en".
inplace	When set to FALSE (the default) writes to new file. If TRUE, writes in place.
encoding	Defaults to utf-8.
theme	Set to "default", currently the only in-built theme that does not result in accessibility errors.
highlight	Set to "null", currently the only in-built highlight that does not result in accessibility errors.

## Value

Adjust the Rmd YAML provided to rmd\_path, improving its accessibility for screen readers. Only works with html output.

**Examples**

```

# create a testfile
rmd <- tempfile("testing", fileext = ".rmd")
# write basic markdown content
writeLines('---
title: "testfile"
author: "Some Author"
date: "`r format(Sys.Date(), "%d %b %Y")`"
output: html_document
---

```${r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

## R Markdown', con = rmd)

# Adjust the document header to improve screen reader accessibility
access_head(rmd, lan = "en")

```

---

access\_img

---

*Produce an accessible image or chart.*


---

**Description**

Reads in an image and produces the HTML structure expected by web accessibility checkers such as WAVE. Also works as a wrapper around ggplot2 charts.

**Usage**

```

access_img(
  img = last_plot(),
  alt = NULL,
  wid = NULL,
  ht = NULL,
  dpi = 300,
  css_class = NULL,
  css_id = NULL
)

```

**Arguments**

|     |   |
|-----|---|
| img | The image to include as accessible HTML. Defaults to 'ggplot2::last_plot()'. Can be replaced with an image written to disc. |
| alt | A character string describing the image for screen reader accessibility.  |
| wid | Width of the image in pixels.   |

|           |   |
|-----------|---|
| ht        | Height of the image in pixels.                                      |
| dpi       | Resolution. Please see <code>?ggplot2::ggsave()</code> for details. |
| css_class | Specify a css class selector for the image.                         |
| css_id    | Specify a css ID selector for the image.                            |

**Value**

Inline HTML with the necessary structure for screen reader accessibility.

**Examples**

```
# create a ggplot2 chart
ggplot(pressure, aes(temperature, pressure)) +
  geom_point()

# Use 'access_img()' to render the chart with alt text
access_img(alt = "Vapor Pressure of Mercury as a Function of Temperature")

# Create a png.
file.create(tempfile("some_img", fileext = ".png"))

# Read it from disk and include alt text
access_img("some_img.png", alt = "Some meaningful alt text")
```

---

access\_rmd

*Produce an accessible Rmarkdown file template.*

---

**Description**

Creates an Rmarkdown template with the HTML structure required by screen readers.

**Usage**

```
access_rmd(
  filenm = NULL,
  title = NULL,
  subtitle = NULL,
  lan = NULL,
  author = Sys.info()[8],
  date = format(Sys.Date(), "%d %b %Y"),
  toc = FALSE,
  encoding = "utf-8",
  force = FALSE,
  theme = "default",
  highlight = "null"
)
```

**Arguments**

|           |   |
|-----------|---|
| filenm    | Required - The name of the file.  |
| title     | Required - The document title.  |
| subtitle  | Optional - The document subtitle.   |
| lan       | Required - The HTML language to set as the value for the lang attribute.  |
| author    | Required - The document author. Defaults to the effective user identified by Sys.info().                            |
| date      | Required - The author date. Defaults to today's date.   |
| toc       | Optional, defaults to FALSE. Should a table of contents be included. Valid entries are FALSE (the default) or TRUE. |
| encoding  | Defaults to "utf-8".  |
| force     | Defaults to FALSE. If TRUE, overwrites a pre-existing file with the same filenm with warning.                       |
| theme     | Set to "default", currently the only in-built theme that does not result in accessibility errors.                   |
| highlight | Set to "null", currently the only in-built highlight that does not result in accessibility errors.                  |

**Value**

An Rmarkdown file with an HTML head, populated with metadata specified within the function parameters.

**Examples**

```
# create an accessible rmarkdown document from scratch
access_rmd(
  "some_filenm",
  title = "Title Goes Here", lan = "en", author = "Author here"
)
```

---

|                 |  |
|-----------------|--|
| assemble_header | <i>Assemble HTML header output from text provided.</i> |
|-----------------|--|

---

**Description**

Returns header structure for use with 'access\_head()' and 'access\_rmd()'.

**Usage**

```
assemble_header(title, subtitle = NULL, auth, doc_date, enc)
```

**Arguments**

|          |  |
|----------|--|
| title    | Text string to use as document title and h1 heading.     |
| subtitle | Text string to use as subtitle heading.                  |
| auth     | Text string or inline code to include as author heading. |
| doc_date | Text string or inline code to include as date heading.   |
| enc      | Text string to use as document encoding.                 |

**Value**

An assembled html output page containing the required toc code.

---

|              |  |
|--------------|--|
| check_compat | <i>Checks if the html doctype is compatible.</i> |
|--------------|--|

---

**Description**

Checks to ensure file is not a special flavour of html output, including xaringan, ioslides, flexdash-board or slidy. Output is an error if html doc is a special case.

**Usage**

```
check_compat(yaml_txt)
```

**Arguments**

|          |   |
|----------|---|
| yaml_txt | Output of 'readLines()' separated from body text. |
|----------|---|

**Value**

Error condition if non-standard html found.

---

|            |   |
|------------|---|
| check_urls | <i>Check an Rmarkdown for broken links.</i> |
|------------|---|

---

**Description**

Check links within an Rmarkdown document for any urls that responds with an error.

**Usage**

```
check_urls(rmd_path)
```

**Arguments**

rmd\_path            Path to the Rmd that requires links to be checked. Rmd must be output type html.

**Value**

Lines of any urls that respond with an error.

**Examples**

```
# create a testfile
links <- tempfile("mixed_links", fileext = ".rmd")
file.create(links)
writeLines("[a good link](https://datasciencecampus.ons.gov.uk/)
[a bad link](https://datasciencecampus.ons.gov.uk/broken)",
           con = links
)
# Test the file
check_urls(links)
```

---

detect\_html\_lang            *Searches a file for valid methods of setting HTML lang attribute.*

---

**Description**

Takes the output of 'readLines()' and looks for lang attribute value, stopping if no valid value is found. Typically used on the output of 'handle\_rmd\_path()'.

**Usage**

```
detect_html_lang(lines = NULL, lang_tags = langs)
```

**Arguments**

lines                The output of 'readLines()' or 'handle\_rmd\_path()'.  
lang\_tags            A vector of valid lang subtag values, taken from <https://www.iana.org/assignments/language-subtag-registry/language-subtag-registry>. Used to confirm whether the lang value is valid.

**Value**

The lang attribute value if found. Stops if no value found.

---

|               |   |
|---------------|---|
| find_all_tags | <i>Find any markdown or HTML syntax tags within read lines.</i> |
|---------------|---|

---

**Description**

Check any lines for images or links and return the line numbers and values.

**Usage**

```
find_all_tags(lines = NULL, tag = c("img", "link"))
```

**Arguments**

|       |  |
|-------|--|
| lines | Read lines, default is NULL & typically used with output of handle_rmd_path. |
| tag   | The type of tag to be searched. Valid values are "img" or "link".            |

**Value**

A named vector with line numbers of any found images and their values.

---

|              |  |
|--------------|--|
| find_alt_lim | <i>Find alt length limit for specific lang values.</i> |
|--------------|--|

---

**Description**

Return language specific alt text length limits. Limits are: eng 100, ger 115, kor 90.

**Usage**

```
find_alt_lim(lines = NULL, lan = detect_html_lang(lines))
```

**Arguments**

|       |   |
|-------|---|
| lines | Any Rmarkdown or HTML file lines. Typically the output of 'handle_rmd_path()'.  |
| lan   | Identify the language of text content. Attempts to find a lang attribute value from the rmd document. Alternatively, use a character string such as "en". |

**Value**

A line limit for alt text.



---

|            |  |
|------------|--|
| find_theme | <i>Searches YAML head for YAML theme parameters, returning the theme if found.</i> |
|------------|--|

---

**Description**

Use on YAML head text only. 'check\_compat()' should be used prior to the use of 'find\_theme()'. Looks for YAML theme parameter value. Returns the theme if found or 'default' if no theme specified.

**Usage**

```
find_theme(yaml = NULL)
```

**Arguments**

yaml            A YAML header text string.

**Value**

The theme value as a character string.

---

|                 |  |
|-----------------|--|
| handle_rmd_path | <i>Check rmd path &amp; file type.</i> |
|-----------------|--|

---

**Description**

Checks rmd\_path exists and that the file suffix is correct.

**Usage**

```
handle_rmd_path(rmd_path = NULL)
```

**Arguments**

rmd\_path        Path to the Rmd to check.

**Value**

Reads rmd lines on success. Error if file does not exist or is not an rmd file.

---

|             |   |
|-------------|---|
| insert_yaml | <i>Insert the required code for the specified toc type.</i> |
|-------------|---|

---

### Description

Inserts YAML header for floating toc if toc is TRUE.

### Usage

```
insert_yaml(toc, header, text, lan, theme = "default", highlight = "null")
```

### Arguments

|           |   |
|-----------|---|
| toc       | TOC status, FALSE (the default) or TRUE.  |
| header    | Metadata items wrapped with 'tags\$header()'  |
| text      | Raw text required for Rmarkdown body.   |
| lan       | lang attribute value.   |
| theme     | Text styling to apply. Current valid value is "default" only. All other themes present accessibility errors on testing. |
| highlight | Currently only "null" is a valid, due to accessibility errors found in all built-in highlight options.                  |

### Value

An assembled html output page containing the required toc code.

---

|               |   |
|---------------|---|
| retrieve_rmds | <i>Find Rmd files within a provided file structure.</i> |
|---------------|---|

---

### Description

Recursively searches file structure for Rmarkdown files. Returns the found Rmd relative paths.

### Usage

```
retrieve_rmds(search_dir = ".", recurse = TRUE, to_txt = FALSE)
```

### Arguments

|            |  |
|------------|--|
| search_dir | The directory to search. Defaults to ".".                                      |
| recurse    | Defaults to TRUE. Should child directories be searched or not.                 |
| to_txt     | Defaults to FALSE. If TRUE, writes found rmd paths in a txt to the search_dir. |

**Value**

Relative paths to all found Rmd files.

**Examples**

```
# Create a test directory
dir.create("parent")
dir.create("parent/child")

# Create some rmds to find
# An empty vector to collect file names
nm_vec <- character()
# create numbered file names
for (num in 1:6) {
  nm <- paste0(num, "-test.Rmd")
  nm_vec <- append(nm_vec, nm)
}
# Create some files in parent & child directories
file.create(paste0("parent/", nm_vec[1:3]))
file.create(paste0("parent/child/", nm_vec[4:6]))

# Return all Rmd files
retrieve_rmds("parent")

# tidy up environment
unlink(c("parent", "child"), recursive = TRUE)
```

---

|                |   |
|----------------|---|
| return_heading | <i>Return a HTML p tag formatted to resemble a heading tag.</i> |
|----------------|---|

---

**Description**

Returns a HTML heading p tag formatted with inline css to appear like of any of the available levels (h1, h2, h3 and so on). This approach is in response to review comments regarding headings that contained no text potentially confusing screen reader users. The class selectors applied will always include "toc-ignore", but an additional selector may be included within the class parameter. If an object of length zero is passed to the txt parameter, NULL is returned, allowing for vectorised usage.

**Usage**

```
return_heading(txt, lvl, class, font_weights = c(h1 = 38, h2 = 30, h3 = 24))
```

**Arguments**

|              |  |
|--------------|--|
| txt          | The text to be used as the 'heading' text.   |
| lvl          | A number indicating the heading level. 1 == tags\$h1() and so on.  |
| class        | A character string to use as the first class attribute. Do not include "toc-ignore", this will be added. |
| font_weights | A named character vector. Names are heading levels & numeric values to apply as font weight.             |

**Value**

null if txt is length 0 or p tag styled to appear as required heading level with class attr applied.

---

|         |  |
|---------|--|
| sus_alt | <i>Check an R markdown document for suspicious alt text.</i> |
|---------|--|

---

**Description**

Checks if an R markdown contains images alt text is equal to alt attribute placeholder values, including: 'nbsp', 'spacer' and src attribute value (filename).

**Usage**

```
sus_alt(rmd_path = NULL, lan = detect_html_lang(lines))
```

**Arguments**

|          |   |
|----------|---|
| rmd_path | Path to the Rmd that contains image tags to check.  |
| lan      | Identify the language of text content. Attempts to find a lang attribute value from the rmd document. Alternatively, use a character string such as "en". |

**Value**

Line numbers of images that has alt text equal to placeholder values.

**Examples**

```
# create a testfile
rmd <- tempfile("testing", fileext = ".rmd")
# write basic markdown content
writeLines('---
title: "Suspicious Alt Text"
author: "Some Author"
date: "`r format(Sys.Date(), "%d %b %Y")`"
output: html_document
---

```{r setup, include=FALSE}
```

```
knitr::opts_chunk$set(echo = TRUE)
````

## R Markdown



'
  con = rmd
)

# test the file for suspicious alt text
sus_alt(rmd, lan = "en")

# Adjust the document header to improve screen reader accessibility
access_head(rmd, lan = "en")
```

# Index

`access_head`, 2  
`access_img`, 3  
`access_rmd`, 4  
`assemble_header`, 5  
  
`check_compat`, 6  
`check_urls`, 6  
  
`detect_html_lang`, 7  
  
`find_all_tags`, 8  
`find_alt_lim`, 8  
`find_theme`, 9  
  
`handle_rmd_path`, 9  
  
`insert_yaml`, 10  
  
`retrieve_rmds`, 10  
`return_heading`, 11  
  
`sus_alt`, 12