

Package ‘webexercises’

September 9, 2026

Version 1.2.0

Title Create Interactive Web Exercises

Description Functions for easily creating interactive web exercises in 'R Markdown', 'Quarto', and package vignettes that students can use in self-guided learning.

URL <https://github.com/psyteachr/webexercises>

Depends R (>= 3.1.2)

Imports stats, utils, jsonlite, knitr, yaml, utils, grDevices,
rstudioapi, htmltools, cli, usethis, rmarkdown (>= 2.2)

License CC BY-SA 4.0

Suggests testthat, bookdown, quarto, xfun, pkgdown, pdftools

Encoding UTF-8

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Dale Barr [aut],
Lisa DeBruine [aut],
Caspar J. Van Lissa [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-0808-5024>>)

Maintainer Caspar J. Van Lissa <c.j.vanlissa@tilburguniversity.edu>

Repository CRAN

Date/Publication 2026-09-09 09:30:02 UTC

Contents

add_to_bookdown	2
add_to_pkgdown	3
add_to_quarto	3
add_to_quarto_file	4
create_quarto_doc	4

escape_regex	5
fitb	5
hide	6
longmcq	7
mcq	8
quiz	9
rmd_webex_support	10
round2	11
strip_lzero	11
style_widgets	12
torf	13
unhide	13
use_webex_vignette	14
webexercises_default	15
webex_vignette	16
Index	17

add_to_bookdown	<i>Add webexercises helper files to bookdown</i>
-----------------	--

Description

Adds the necessary helper files to an existing bookdown project and edits the `_output.yml` and `_bookdown.yml` files accordingly. If the directory does not have a bookdown project in it, a template project will be set up.

Usage

```
add_to_bookdown(  
  bookdown_dir = ".",  
  include_dir = "include",  
  script_dir = "R",  
  output_format = c("bs4_book", "gitbook", "html_book", "tufte_html_book"),  
  render = FALSE  
)
```

Arguments

bookdown_dir	The base directory for your bookdown project
include_dir	The directory where you want to put the css and js files (defaults to "include")
script_dir	The directory where you want to put the .R script (defaults to "R")
output_format	The bookdown format you want to add webexercises to (defaults to "bs4_book") This is typically your default HTML format in the <code>_output.yml</code> file.
render	Whether to render the book after updating (defaults to FALSE).

Value

No return value, called for side effects.

add_to_pkgdown	<i>Add webexercises helper files to pkgdown</i>
----------------	---

Description

Adds the necessary helper files to an existing pkgdown project.

Usage

```
add_to_pkgdown(pkgdown_dir = ".")
```

Arguments

pkgdown_dir The base directory for your pkgdown project

Value

No return value, called for side effects.

add_to_quarto	<i>Add webexercises helper files to quarto</i>
---------------	--

Description

Adds the necessary helper files to an existing quarto project and edits the _quarto.yml file accordingly. A demo file for webexercises will be added and optionally rendered.

Usage

```
add_to_quarto(
  quarto_dir = ".",
  include_dir = "include",
  output_format = c("html")
)
```

Arguments

quarto_dir The base directory for your quarto project
 include_dir The directory where you want to put the css and js files (defaults to "include")
 output_format The format you want to add webexercises to (only html for now)

Value

No return value, called for side effects.

add_to_quarto_file	<i>Add webexercises 'YAML' metadata</i>
--------------------	---

Description

Adds the necessary 'YAML' metadata to an existing 'Quarto' or 'Rmarkdown' file.

Usage

```
add_to_quarto_file(filename)
```

```
add_to_rmarkdown_file(filename)
```

Arguments

filename	The name of the file to edit.
----------	-------------------------------

Value

No return value, called for side effects.

create_quarto_doc	<i>Create a quarto document with webexercise</i>
-------------------	--

Description

Creates a new directory with the file name and copies in a demo qmd file and the necessary helper files.

Usage

```
create_quarto_doc(name = "Untitled", open = interactive())
```

Arguments

name	Name of the new document
open	Whether to open the document in RStudio

Value

The file path to the document

escape_regex	<i>Escape a string for regex</i>
--------------	----------------------------------

Description

Escape a string for regex

Usage

```
escape_regex(string)
```

Arguments

`string` A string to escape.

Value

A string with escaped characters.

Examples

```
escape_regex("library(tidyverse)")
```

fitb	<i>Create a fill-in-the-blank question</i>
------	--

Description

Create a fill-in-the-blank question

Usage

```
fitb(  
  answer,  
  width = min(100, max(nchar(answer))),  
  num = NULL,  
  ignore_case = FALSE,  
  tol = NULL,  
  ignore_ws = TRUE,  
  regex = FALSE  
)
```

Arguments

answer	The correct answer (can be a vector if there is more than one correct answer).
width	Width of the input box in characters. Defaults to the length of the longest answer, with a minimum of 100 characters.
num	Whether the input is numeric, in which case allow for leading zeroes to be omitted. Determined from the answer data type if not specified.
ignore_case	Whether to ignore case (capitalization).
tol	The tolerance within which numeric answers will be accepted; i.e. if $\text{abs}(\text{response} - \text{true.answer}) < \text{tol}$, the answer is correct (implies <code>num=TRUE</code>).
ignore_ws	Whether to ignore whitespace.
regex	Whether to use regex to match answers (concatenates all answers with 'l' before matching).

Details

Writes html code that creates an input box widget. Call this function inline in an RMarkdown document. See the Web Exercises RMarkdown template for examples of its use in RMarkdown.

Value

A character string with HTML code to generate an input box.

Examples

```
# What is 2 + 2?
fitb(4, num = TRUE)

# What was the name of the Beatles drummer?
fitb(c("Ringo", "Ringo Starr"), ignore_case = TRUE)

# What is pi to three decimal places?
fitb(pi, num = TRUE, tol = .001)
```

hide

Create button revealing hidden content

Description

Create button revealing hidden content

Usage

```
hide(button_text = "Solution")
```

Arguments

button_text	Text to appear on the button that reveals the hidden content.
-------------	---

Details

Writes HTML to create a content that is revealed by a button press. Call this function inline in an RMarkdown document. Any content appearing after this call up to an inline call to `unhide()` will only be revealed when the user clicks the button. See the Web Exercises RMarkdown Template for examples.

Value

A character string containing HTML code to create a button that reveals hidden content.

See Also

`unhide`

Examples

```
# default behavior is to generate a button that says "Solution"
hide()

# or the button can display custom text
hide("Click here for a hint")
```

`longmcq`*Longer MCQs with Radio Buttons*

Description

Longer MCQs with Radio Buttons

Usage

```
longmcq(opts)
```

Arguments

<code>opts</code>	Vector of alternatives. The correct answer is the element(s) of this vector named 'answer'.
-------------------	---

Details

Writes html code that creates a radio button widget, with a single correct answer. This is more suitable for longer answers. Call this function inline in an RMarkdown document. See the Web Exercises RMarkdown template for further examples.

Value

A character string containing HTML code to create a set of radio buttons.

Examples

```
# What is a p-value?
opts <- c(
  "the probability that the null hypothesis is true",
  answer = paste("the probability of the observed, or more extreme, data",
    "under the assumption that the null-hypothesis is true"),
  "the probability of making an error in your conclusion"
)

longmcq(opts)
```

mcq

Create a multiple-choice question

Description

Create a multiple-choice question

Usage

```
mcq(opts)
```

Arguments

opts	Vector of alternatives. The correct answer is the element(s) of this vector named 'answer'.
------	---

Details

Writes html code that creates an option box widget, with one or more correct answers. Call this function inline in an RMarkdown document. See the Web Exercises RMarkdown template for further examples.

Value

A character string with HTML code to generate a pull-down menu.

Examples

```
# How many planets orbit closer to the sun than the Earth?
mcq(c(1, answer = 2, 3))

# Which actor played Luke Skywalker in the movie Star Wars?
mcq(c("Alec Guinness", answer = "Mark Hamill", "Harrison Ford"))
```

quiz

Create a Quiz

Description

Convenience function for creating a basic quiz in HTML format from the arguments captured by ‘...’, where the type of each question is determined automatically from the class of the arguments.

Usage

```
quiz(..., title = "Quiz", show_box = TRUE, show_check = TRUE)
```

Arguments

...	Each argument should be a named vector, see Details.
title	Atomic character, default: ‘Quiz’
show_box	Logical, whether or not to draw a box around the quiz. Default: ‘TRUE’
show_check	Logical, whether or not to show a button to check answers. Default: ‘TRUE’

Details

The function renders questions captured by the arguments in ‘...’. The name of each argument is the text of the question. The value of each argument determined the question type and its correct answer. The following types of questions are supported:

“**torf()**” The argument should be a single value of type ‘logical’, e.g.: “The answer to this question is true.” = TRUE‘

“**mcq()**” The argument should be a vector of type ‘character’. The first element is taken as the correct answer; the order of answers is randomized. E.g.: “This multiple choice question has three answers.” = c(“Correct”, “Incorrect”, “Not sure”)‘

“**fitb()**” The argument should be of type ‘numeric’. If the vector is atomic, the first element is taken as the correct answer, e.g.: “Provide an exact floating point answer of 0.81” = 0.81‘. If the vector has two elements, the second element is taken as the tolerance ‘tol’, e.g.: “Here, 0.8 will be correct.” = c(0.81, 0.01)‘. If the vector is of type ‘integer’, the tolerance is set to zero, e.g.: “The answer is 4.” = 4L‘

Alternatively, ‘...’ may contain a single atomic character referring to a text file that contains the questions, see examples.

Value

‘NULL’, this function is called for its side effect of printing HTML code using ‘cat()’.

Examples

```
# Quiz from arguments:
invisible(capture.output(
  quiz(
    "The answer to this question is true." = TRUE,
    "This multiple choice question has three answers." = c("Correct", "Incorrect", "Not sure"),
    "Provide an exact floating point answer of 0.81" = 0.81
  )
))
# From a file:
quiz_file <- tempfile()
writeLines(
  c("The answer is true. = TRUE",
    "The answer is correct = c(answer = \"Correct\", \"Incorrect\", \"Not sure\")",
    "The answer is exactly .81 = 0.81",
    "But here, .8 is also fine = c(0.81, .01)",
    "Here, answer exactly 4. = 4L")
  , quiz_file)
invisible(capture.output(quiz(quiz_file)))
```

rmd_webex_support	<i>Add webexercises support</i>
-------------------	---------------------------------

Description

This convenience function can be called within an 'Rmarkdown' document to add the 'css' and 'JavaScript' code required for webexercises' html output.

Usage

```
rmd_webex_support()
```

Value

Character string of class 'HTML'.

See Also

[HTML](#)

Examples

```
invisible(rmd_webex_support())
```

round2	<i>Round up from .5</i>
--------	-------------------------

Description

Round up from .5

Usage

```
round2(x, digits = 0)
```

Arguments

x	A vector of numeric values.
digits	Integer indicating the number of decimal places ('round') or significant digits ('signif') to be used.

Details

Implements rounding using the "round up from .5" rule, which is more conventional than the "round to even" rule implemented by R's built-in [round](#) function.

Value

A vector of rounded numeric values.

Examples

```
round2(c(2, 2.5))  
  
# compare to:  
round(c(2, 2.5))
```

strip_lzero	<i>Strip leading zero from numeric string</i>
-------------	---

Description

Strip leading zero from numeric string

Usage

```
strip_lzero(x)
```

Arguments

x	A numeric string (or number that can be converted to a string).
---	---

Value

A string with leading zero removed.

Examples

```
strip_lzero("0.05")
```

style_widgets

Change webexercises widget style

Description

Change webexercises widget style

Usage

```
style_widgets(
  incorrect = "#983E82",
  correct = "#59935B",
  highlight = "#467AAC"
)
```

Arguments

incorrect	The colour of the widgets when the answer is incorrect (defaults to pink #983E82).
correct	The colour of the widgets when the correct answer not filled in (defaults to green #59935B).
highlight	The colour of the borders around hidden blocks and checked sections (defaults to blue #467AAC).

Details

Call this function in an RMarkdown document to change the feedback colours using R colour names (see 'colours()') or any valid CSS colour specification (e.g., red, rgb(255,0,0), hsl(0, 100

If you want more control over the widget styles, please edit the webex.css file directly.

Value

A character string containing HTML code to change the CSS style values for widgets.

Examples

```
style_widgets("goldenrod", "purple")
```

torf	Create a true-or-false question
------	---------------------------------

Description

Create a true-or-false question

Usage

```
torf(answer)
```

Arguments

answer	Logical value TRUE or FALSE, corresponding to the correct answer.
--------	---

Details

Writes html code that creates an option box widget with TRUE or FALSE as alternatives. Call this function inline in an RMarkdown document. See the Web Exercises RMarkdown template for further examples.

Value

A character string with HTML code to generate a pull-down menu with elements TRUE and FALSE.

Examples

```
# True or False? 2 + 2 = 4
torf(TRUE)

# True or False? The month of April has 31 days.
torf(FALSE)
```

unhide	End hidden HTML content
--------	-------------------------

Description

End hidden HTML content

Usage

```
unhide()
```

Details

Call this function inline in an RMarkdown document to mark the end of hidden content (see the Web Exercises RMarkdown Template for examples).

Value

A character string containing HTML code marking the end of hidden content.

See Also

hide

Examples

```
# just produce the closing </div>
unhide()
```

use_webex_vignette	<i>Create a webexercises vignette</i>
--------------------	---------------------------------------

Description

Wraps [use_vignette](#) to add a vignette or article to ‘vignettes/’ with support for ‘webexercises’.

Usage

```
use_webex_vignette(name, title = NULL, type = c("vignette", "article"))
```

Arguments

name	Atomic character, vignette name. See use_vignette .
title	Atomic character, vignette title. See use_vignette .
type	Atomic character, one of ‘c("vignette", "article")’, defaults to “"vignette"”.

Value

Returns ‘NULL’ invisibly, called for its side effects.

Examples

```
## Not run:
use_webex_vignette("vignette_with_quiz.Rmd", "Quiz people with webexercises")

## End(Not run)
```

webexercises_default *Create default webexercises document*

Description

This function wraps `rmarkdown::html_document` to configure compilation to embed the default webexercises CSS and JavaScript files in the resulting HTML.

Usage

```
webexercises_default(...)
```

Arguments

... Additional function arguments to pass to [html_document](#).

Details

Call this function as the `output_format` argument for the [render](#) function when compiling HTML documents from RMarkdown source.

Value

R Markdown output format to pass to 'render'.

See Also

[render](#), [html_document](#)

Examples

```
# copy the webexercises 'R Markdown' template to a temporary file
## Not run:
my_rmd <- tempfile(fileext = ".Rmd")
rmarkdown::draft(my_rmd, "webexercises", "webexercises")

# compile it
rmarkdown::render(my_rmd, webexercises::webexercises_default())

# view the result
browseURL(sub("\\.Rmd$", ".html", my_rmd))

## End(Not run)
```

webex_vignette*Create Vignette with webexercises Support*

Description

This function wraps `rmarkdown::html_document` to configure compilation to embed the default webexercises CSS and JavaScript files in the resulting HTML.

Usage

```
webex_vignette(...)
```

Arguments

... Additional function arguments to pass to [html_document](#).

Details

Call this function as the `output_format` argument for the [render](#) function when compiling HTML documents from RMarkdown source.

Value

R Markdown output format to pass to 'render'.

See Also

[render](#), [html_document](#)

Examples

```
# copy the webexercises 'R Markdown' template to a temporary file
## Not run:
my_rmd <- tempfile(fileext = ".Rmd")
rmarkdown::draft(my_rmd, "webexercises", "webexercises")

# compile it
rmarkdown::render(my_rmd, webexercises::webex_vignette())

# view the result
browseURL(sub("\\.Rmd$", ".html", my_rmd))

## End(Not run)
```

Index

`add_to_bookdown`, [2](#)
`add_to_pkgdown`, [3](#)
`add_to_quarto`, [3](#)
`add_to_quarto_file`, [4](#)
`add_to_rmarkdown_file`
 (`add_to_quarto_file`), [4](#)

`create_quarto_doc`, [4](#)

`escape_regex`, [5](#)

`fitb`, [5](#)

`hide`, [6](#)
`HTML`, [10](#)
`html_document`, [15](#), [16](#)

`longmcq`, [7](#)

`mcq`, [8](#)

`quiz`, [9](#)

`render`, [15](#), [16](#)
`rmd_webex_support`, [10](#)
`round`, [11](#)
`round2`, [11](#)

`strip_lzero`, [11](#)
`style_widgets`, [12](#)

`torf`, [13](#)

`unhide`, [13](#)
`use_vignette`, [14](#)
`use_webex_vignette`, [14](#)

`webex_vignette`, [16](#)
`webexercises_default`, [15](#)