

# Package: webexercises (via r-universe)

October 13, 2024

**Version** 1.1.0.9000

**Date** 2023-04-18

**Title** Create Interactive Web Exercises in 'R Markdown' (Formerly 'webex')

**Description** Functions for easily creating interactive web pages using 'R Markdown' that students can use in self-guided learning.

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

**Depends** R (>= 3.1.2)

**Imports** jsonlite, knitr, yaml, utils, grDevices, rstudioapi, rmarkdown (>= 2.2)

**License** CC BY-SA 4.0

**RoxygenNote** 7.2.1

**Suggests** testthat, bookdown, quarto, xfun

**Encoding** UTF-8

**VignetteBuilder** knitr

**Repository** <https://psyteachr.r-universe.dev>

**RemoteUrl** <https://github.com/psyteachr/webexercises>

**RemoteRef** HEAD

**RemoteSha** 01ea77227fe39b9b4c208517038521c227ff1eb3

## Contents

add_to_bookdown . . . . .	2
add_to_quarto . . . . .	3
create_quarto_doc . . . . .	3
escape_regex . . . . .	4
fitb . . . . .	4
hide . . . . .	5
longmcq . . . . .	6
mcq . . . . .	7

round2 . . . . .	8
strip_lzero . . . . .	8
style_widgets . . . . .	9
torf . . . . .	10
unhide . . . . .	10
webexercises_default . . . . .	11
<b>Index</b>	<b>13</b>

---

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

<code>bookdown_dir</code>	The base directory for your bookdown project
<code>include_dir</code>	The directory where you want to put the css and js files (defaults to "include")
<code>script_dir</code>	The directory where you want to put the .R script (defaults to "R")
<code>output_format</code>	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.
<code>render</code>	Whether to render the book after updating (defaults to FALSE).

## 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.

---

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 = calculated_width,  
  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.
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**

`opts`                    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"))
```

round2 *Round up from .5*

---

**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. This implementation was taken from <https://stackoverflow.com/a/12688836>.

**Value**

A vector of rounded numeric values.

**Examples**

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

---

strip\_lzero *Strip leading zero from numeric string*

---

**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	<i>Change webexercises widget style</i>
---------------	---

---

**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	<i>Create a true-or-false question</i>
------	--

---

**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	<i>End hidden HTML content</i>
--------	--------------------------------

---

**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()
```

---

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)
```

# Index

`add_to_bookdown`, 2  
`add_to_quarto`, 3  
  
`create_quarto_doc`, 3  
  
`escape_regex`, 4  
  
`fitb`, 4  
  
`hide`, 5  
`html_document`, 11  
  
`longmcq`, 6  
  
`mcq`, 7  
  
`render`, 11  
`round`, 8  
`round2`, 8  
  
`strip_lzero`, 8  
`style_widgets`, 9  
  
`torf`, 10  
  
`unhide`, 10  
  
`webexercises_default`, 11