

Package ‘learnr’

November 16, 2017

Type Package

Title Interactive Tutorials for R

Version 0.9.1

Description Create interactive tutorials using R Markdown. Use a combination of narrative, figures, videos, exercises, and quizzes to create self-paced tutorials for learning about R and R packages.

License Apache License 2.0 | file LICENSE

URL <https://rstudio.github.io/learnr/>

BugReports <https://github.com/rstudio/learnr/issues>

Imports utils, parallel, withr, rappdirs, rprojroot, jsonlite, htmltools (>= 0.3.5), htmlwidgets, evaluate, knitr (>= 1.14), markdown, shiny (>= 1.0), rmarkdown (>= 1.6)

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

Suggests testthat

NeedsCompilation no

Author Barbara Borges [aut, cre],
JJ Allaire [aut],
RStudio, Inc. [cph],
Ajax.org B.V. [ctb, cph] (Ace library),
Zeno Rocha [ctb, cph] (clipboard.js library),
Nick Payne [ctb, cph] (Bootbox library),
Julie Cameron [ctb] (SlickQuiz library),
Quicken Loans [cph] (SlickQuiz library),
Mozilla [ctb, cph] (localforage library)

Maintainer Barbara Borges <barbara@rstudio.com>

Repository CRAN

Date/Publication 2017-11-16 22:29:57 UTC

R topics documented:

filesystem_storage	2
initialize_tutorial	2
quiz	3
run_tutorial	4
tutorial	4
tutorial_html_dependency	6
tutorial_options	6

Index	8
--------------	----------

filesystem_storage	<i>Filesystem-based storage for tutor state data</i>
--------------------	--

Description

Tutorial state storage handler that uses the filesystem as a backing store. The directory will contain tutorial state data partitioned by `user_id`, `tutorial_id`, and `tutorial_version` (in that order)

Usage

```
filesystem_storage(dir, compress = TRUE)
```

Arguments

<code>dir</code>	Directory to store state data within
<code>compress</code>	Should <code>.rds</code> files be compressed?

Value

Storage handler suitable for `options(tutorial.storage = ...)`

initialize_tutorial	<i>Initialize Tutorial R Markdown Extensions</i>
---------------------	--

Description

One time initialization of R Markdown extensions required by the **tutor** package. This function is typically called automatically as a result of using exercises or questions.

Usage

```
initialize_tutorial()
```

Description

Add interactive multiple choice quiz questions to a tutorial.

Usage

```
quiz(..., caption = "Quiz")

question(text, ..., type = c("auto", "single", "multiple"),
  correct = "Correct!", incorrect = "Incorrect.", allow_retry = FALSE,
  random_answer_order = FALSE)

answer(text, correct = FALSE, message = NULL)
```

Arguments

...	One or more questions or answers
caption	Optional quiz caption (defaults to "Quiz")
text	Question or option text
type	Type of quiz question. Typically this can be automatically determined based on the provided answers. Pass "single" to indicate that even though multiple correct answers are specified that inputs which include only one correct answer are still correct. Pass "multiple" to force the use of checkboxes (as opposed to radio buttons) even though only once correct answer was provided.
correct	For question, text to print for a correct answer (defaults to "Correct!"). For answer, a boolean indicating whether this answer is correct.
incorrect	Text to print for an incorrect answer (defaults to "Incorrect.")
allow_retry	Allow retry for incorrect answers.
random_answer_order	Display answers in a random order.
message	Additional message to display along with correct/incorrect feedback.

Examples

```
## Not run:
question("What number is the letter A in the alphabet?",
  answer("8"),
  answer("14"),
  answer("1", correct = TRUE),
  answer("23")
)
```

```

question("Where are you right now? (select ALL that apply)",
  answer("Planet Earth", correct = TRUE),
  answer("Pluto"),
  answer("At a computing device", correct = TRUE),
  answer("In the Milky Way", correct = TRUE),
  incorrect = paste0("Incorrect. You're on Earth, ",
                    "in the Milky Way, at a computer.")
)

## End(Not run)

```

run_tutorial

Run a tutorial

Description

Run a tutorial which is contained within an R package.

Usage

```
run_tutorial(name, package, shiny_args = NULL)
```

Arguments

name	Tutorial name (subdirectory within tutorials/ directory of installed package).
package	Name of package
shiny_args	Additional arguments to forward to shiny::runApp .

Details

Note that when running a tutorial Rmd file with `run_tutorial` the tutorial Rmd should have already been rendered as part of the development of the package (i.e. the corresponding tutorial .html file for the .Rmd file must exist).

tutorial

Tutorial document format

Description

Long-form tutorial which includes narrative, figures, videos, exercises, and questions.

Usage

```
tutorial(fig_width = 6.5, fig_height = 4, fig_retina = 2,
  fig_caption = TRUE, progressive = FALSE, allow_skip = FALSE,
  dev = "png", df_print = "paged", smart = TRUE, theme = "rstudio",
  mathjax = "default", extra_dependencies = NULL, css = NULL,
  includes = NULL, md_extensions = NULL, pandoc_args = NULL, ...)
```

Arguments

<code>fig_width</code>	Default width (in inches) for figures
<code>fig_height</code>	Default width (in inches) for figures
<code>fig_retina</code>	Scaling to perform for retina displays (defaults to 2, which currently works for all widely used retina displays). Set to NULL to prevent retina scaling. Note that this will always be NULL when <code>keep_md</code> is specified (this is because <code>fig_retina</code> relies on outputting HTML directly into the markdown document).
<code>fig_caption</code>	TRUE to render figures with captions
<code>progressive</code>	Display sub-topics progressively (i.e. wait until previous topics are either completed or skipped before displaying subsequent topics).
<code>allow_skip</code>	Allow users to skip sub-topics (especially useful when <code>progressive</code> is TRUE).
<code>dev</code>	Graphics device to use for figure output (defaults to <code>png</code>)
<code>df_print</code>	Method to be used for printing data frames. Valid values include "default", "kable", "tibble", and "paged". The "default" method uses <code>print.data.frame</code> . The "kable" method uses the <code>knitr::kable</code> function. The "tibble" method uses the tibble package to print a summary of the data frame. The "paged" method creates a paginated HTML table (note that this method is only valid for formats that produce HTML). In addition to the named methods you can also pass an arbitrary function to be used for printing data frames. You can disable the <code>df_print</code> behavior entirely by setting the option <code>rmarkdown.df_print</code> to FALSE.
<code>smart</code>	Produce typographically correct output, converting straight quotes to curly quotes, — to em-dashes, – to en-dashes, and ... to ellipses.
<code>theme</code>	Visual theme ("rstudio", "default", "cerulean", "journal", "flatly", "readable", "spacelab", "united", "cosmo", "lumen", "paper", "sandstone", "simplex", or "yeti").
<code>mathjax</code>	Include mathjax. The "default" option uses an https URL from a MathJax CDN. The "local" option uses a local version of MathJax (which is copied into the output directory). You can pass an alternate URL or pass NULL to exclude MathJax entirely.
<code>extra_dependencies</code>	Additional function arguments to pass to the base R Markdown HTML output formatter <code>html_document_base</code>
<code>css</code>	One or more css files to include
<code>includes</code>	Named list of additional content to include within the document (typically created using the <code>includes</code> function).
<code>md_extensions</code>	Markdown extensions to be added or removed from the default definition or R Markdown. See the <code>rmarkdown_format</code> for additional details.

pandoc_args Additional command line options to pass to pandoc
 ... Forward parameters to html_document

tutorial_html_dependency
Tutorial HTML dependency

Description

Tutorial HTML dependency

Usage

```
tutorial_html_dependency()
```

Details

HTML dependency for core tutorial JS and CSS. This should be included as a dependency for custom tutorial formats that wish to ensure that that tutorial.js and tutorial.css are loaded prior their own scripts and stylesheets.

tutorial_options *Set tutorial options*

Description

Set various tutorial options that control the display and evaluation of exercises.

Usage

```
tutorial_options(exercise.cap = "Code", exercise.eval = FALSE,  

  exercise.timelimit = 30, exercise.lines = NULL, exercise.checker = NULL,  

  exercise.completion = TRUE, exercise.diagnostics = TRUE,  

  exercise.startover = TRUE)
```

Arguments

exercise.cap Caption for exercise chunk (defaults to "Code").
 exercise.eval Whether to pre-evaluate the exercise so the reader can see some default output (defaults to FALSE).
 exercise.timelimit Number of seconds to limit execution time to (defaults to 30).
 exercise.lines Lines of code for exercise editor (defaults to the number of lines in the code chunk).

- `exercise.checker`
Function used to check exercise answers.
- `exercise.completion`
Use code completion in exercise editors.
- `exercise.diagnostics`
Show diagnostics in exercise editors.
- `exercise.startover`
Show "Start Over" button on exercise.

Index

`answer (quiz)`, [3](#)

`filesystem_storage`, [2](#)

`html_document_base`, [5](#)

`includes`, [5](#)

`initialize_tutorial`, [2](#)

`knitr::kable`, [5](#)

`question (quiz)`, [3](#)

`quiz`, [3](#)

`rmarkdown_format`, [5](#)

`run_tutorial`, [4](#)

`shiny::runApp`, [4](#)

`tutorial`, [4](#)

`tutorial_html_dependency`, [6](#)

`tutorial_options`, [6](#)