

Package: tutorizeR (via r-universe)

October 5, 2026

Type Package

Title Convert R Markdown or Quarto Content into Interactive Tutorials

Version 0.4.3

Author Aurélien Nicosia [aut, cre]

Maintainer Aurélien Nicosia <aurelien.nicosia@mat.ulaval.ca>

Description Helps teachers convert existing '.Rmd' and '.qmd' teaching material into interactive tutorials for 'learnr' or 'quarto-live'. Conversion preserves narrative text, setup chunks, and major chunk options, supports teacher tags, and provides explicit validation and conversion reports.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.0)

Imports cli, jsonlite, lifecycle, rlang, rmarkdown, rstudioapi, yaml

Suggests commonmark, covr, knitr, learnr, lintr, miniUI, shiny, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

URL <https://github.com/AurelienNicosiaULaval/tutorizeR>

BugReports <https://github.com/AurelienNicosiaULaval/tutorizeR/issues>

Config/pak/sysreqs cmake make libuv1-dev

Repository <https://aureliennicosiaulaval.r-universe.dev>

Date/Publication 2026-07-07 22:13:03 UTC

RemoteUrl <https://github.com/aureliennicosiaulaval/tutorizer>

RemoteRef HEAD

RemoteSha 1dcf0224e15006d9cc08b05ce06a44c926675d08

Contents

check_tutorial	2
convert_folder	3
convert_to_tutorial	4
export_lms_manifest	5
export_tutorial_package	6
launch_tutorizeR_addin	6
launch_tutorizeR_folder_addin	7
launch_tutorizeR_preview_addin	7
lint_source	8
load_question_bank	8
print.tutorize_folder_report	9
print.tutorize_lint_report	10
print.tutorize_lms_manifest	10
print.tutorize_report	11
tr	11
tutorize	12
validate_input	13
validate_output	13
validate_question_bank	14
write_tutorize_report	15
Index	16

check_tutorial	<i>Render a generated tutorial and return render status</i>
----------------	---

Description

Render a generated tutorial and return render status

Usage

```
check_tutorial(file, language = c("en", "fr"))
```

Arguments

- file Path to tutorial ‘.Rmd’ to render.
- language Message language (“en” or “fr”).

Value

Invisibly, the render output path or the error object.

Examples

```
## Not run:
check_tutorial("analysis-tutorial.Rmd")

## End(Not run)
```

convert_folder	<i>Convert all ‘.Rmd’/‘.qmd’ files in a folder</i>
----------------	--

Description

Converts every matching source file in ‘dir’, with resilient per-file error handling and a summary report.

Usage

```
convert_folder(
  dir = ".",
  pattern = "\\.(Rmd|qmd)$",
  recursive = FALSE,
  output_dir = NULL,
  format = c("learnr", "quarto-live"),
  assessment = c("code", "mcq", "both"),
  overwrite = FALSE,
  language = c("en", "fr"),
  seed = NULL,
  verbose = TRUE,
  question_bank = NULL,
  mcq_source = c("inline", "bank", "mixed"),
  lint_strict = FALSE
)
```

Arguments

dir	Folder containing source files.
pattern	Regex pattern used to select files.
recursive	Logical; recurse into subdirectories?
output_dir	Optional output directory.
format	Output format (“learnr” or “quarto-live”).
assessment	Assessment mode (“code”, “mcq”, “both”).
overwrite	Logical; overwrite existing outputs?
language	Message/template language (“en” or “fr”).
seed	Optional integer seed injected in setup chunk.
verbose	Logical; print progress and summary?

question_bank	Optional path(s) or 'tutorize_question_bank' object.
mcq_source	MCQ generation source ("inline", "bank", "mixed").
lint_strict	Logical; fail conversion when lint reports errors.

Value

A 'tutorize_folder_report' object (invisibly).

Examples

```
## Not run:
report <- convert_folder("labs", recursive = TRUE)
print(report)

## End(Not run)
```

convert_to_tutorial	<i>Backward-compatible wrapper for tutorial conversion</i>
---------------------	--

Description

'convert_to_tutorial()' is kept for compatibility with earlier versions. New code should use [tutorize()].

Usage

```
convert_to_tutorial(
  input_file,
  output_file = NULL,
  assessment = c("code", "mcq", "both"),
  format = c("learnr", "quarto-live"),
  add_mcq = NULL,
  question_bank = NULL,
  mcq_source = c("inline", "bank", "mixed"),
  lint_strict = FALSE
)
```

Arguments

input_file	Path to source '.Rmd' or '.qmd' file.
output_file	Optional output path. If 'NULL', a default path is used.
assessment	Assessment mode: "code", "mcq", or "both".
format	Output format: "learnr" or "quarto-live".
add_mcq	Deprecated. Use 'assessment' instead.
question_bank	Optional path(s) or 'tutorize_question_bank' object.
mcq_source	MCQ generation source: "inline", "bank", "mixed".
lint_strict	Logical; fail conversion when lint reports errors.

Value

Invisibly, the generated output file path.

Examples

```
## Not run:
convert_to_tutorial("analysis.Rmd")

## End(Not run)
```

export_lms_manifest	<i>Export a profile-aware LMS manifest</i>
---------------------	--

Description

Export a profile-aware LMS manifest

Usage

```
export_lms_manifest(
  input,
  output_file = NULL,
  profile = c("generic", "canvas", "moodle"),
  include_solutions = TRUE,
  language = c("en", "fr")
)
```

Arguments

input	Source ‘.Rmd’/‘.qmd’ file.
output_file	Optional output path. Default uses input stem.
profile	Manifest profile (“generic”, “canvas”, “moodle”).
include_solutions	Logical; mark exercise solutions as available.
language	Manifest language (“en” or “fr”).

Value

A ‘tutorize_lms_manifest’ object.

```
export_tutorial_package
```

Export a tutorial-ready package scaffold (opt-in)

Description

Export a tutorial-ready package scaffold (opt-in)

Usage

```
export_tutorial_package(
  input,
  path = ".",
  package_name = NULL,
  overwrite = FALSE
)
```

Arguments

input	Source ‘.Rmd’/‘.qmd’ file.
path	Parent directory where the package directory will be created.
package_name	Optional package name. Derived from input by default.
overwrite	Logical; replace existing package directory if present.

Value

Invisibly returns the created package path.

```
launch_tutorizeR_addin
```

RStudio Addin: Convert active document to tutorial

Description

Opens a lightweight prompt flow (format, assessment, output directory, overwrite) and converts the currently active file.

Usage

```
launch_tutorizeR_addin()
```

Value

Invisibly returns the ‘tutorize_report’ object, or ‘NULL’ when cancelled.

`launch_tutorizeR_folder_addin`*RStudio Addin: Convert a folder to tutorials*

Description

Prompts for folder and conversion options, then converts all matching files.

Usage

```
launch_tutorizeR_folder_addin()
```

Value

Invisibly returns a ‘tutorize_folder_report’ object, or ‘NULL’ when cancelled.

`launch_tutorizeR_preview_addin`*RStudio Addin: Preview conversion with lint and diff*

Description

Opens a gadget with Source, Output, Diff, Lint and Logs tabs.

Usage

```
launch_tutorizeR_preview_addin()
```

Value

Invisibly returns a ‘tutorize_report’ object or ‘NULL’.

lint_source	<i>Lint source pedagogical constraints before conversion</i>
-------------	--

Description

Lint source pedagogical constraints before conversion

Usage

```
lint_source(
  input,
  question_bank = NULL,
  language = c("en", "fr"),
  strict = FALSE
)
```

Arguments

input	Source ‘.Rmd’ or ‘.qmd’ file.
question_bank	Optional ‘tutorize_question_bank’ object.
language	Message language (“en” or “fr”).
strict	Logical; if ‘TRUE’, stop when lint errors are present.

Value

A ‘tutorize_lint_report’ object.

load_question_bank	<i>Load a reusable MCQ question bank from YAML/JSON files</i>
--------------------	---

Description

Load a reusable MCQ question bank from YAML/JSON files

Usage

```
load_question_bank(
  path,
  recursive = TRUE,
  format = c("auto", "yaml", "json"),
  strict = TRUE,
  language = c("en", "fr")
)
```

Arguments

path	File or directory path(s) containing question bank files.
recursive	Logical; recurse into subdirectories when 'path' is a directory.
format	Input format: "auto", "yaml", or "json".
strict	Logical; if 'TRUE', validation errors stop execution.
language	Language for validation messages ("en" or "fr").

Value

A 'tutorize_question_bank' object.

```
print.tutorize_folder_report
```

Print method for folder conversion reports

Description

Print method for folder conversion reports

Usage

```
## S3 method for class 'tutorize_folder_report'  
print(x, ...)
```

Arguments

x	A 'tutorize_folder_report' object.
...	Unused.

Value

'x', invisibly.

```
print.tutorize_lint_report
```

Print method for lint reports

Description

Print method for lint reports

Usage

```
## S3 method for class 'tutorize_lint_report'  
print(x, ...)
```

Arguments

x	A ‘tutorize_lint_report’ object.
...	Unused.

Value

‘x’, invisibly.

```
print.tutorize_lms_manifest
```

Print method for LMS manifest exports

Description

Print method for LMS manifest exports

Usage

```
## S3 method for class 'tutorize_lms_manifest'  
print(x, ...)
```

Arguments

x	A ‘tutorize_lms_manifest’ object.
...	Unused.

Value

‘x’, invisibly.

print.tutorize_report *Print method for tutorize conversion reports*

Description

Print method for tutorize conversion reports

Usage

```
## S3 method for class 'tutorize_report'
print(x, ...)
```

Arguments

x	A 'tutorize_report' object.
...	Unused.

Value

'x', invisibly.

tr	<i>Translate a tutorizeR message key</i>
----	--

Description

Looks up a message key in language dictionaries and interpolates named placeholders (for example, "{name}").

Usage

```
tr(key, ..., language = NULL)
```

Arguments

key	Dot-separated translation key (e.g. "messages.render_ok").
...	Named interpolation values.
language	Optional language ("en" or "fr").

Value

A translated character string.

tutorize

Convert a source document into an interactive tutorial

Description

`tutorize()` is the canonical high-level API. It converts an existing `‘.Rmd’` or `‘.qmd’` source document into a tutorial-ready document for `‘learnr’` or `‘quarto-live’`.

Usage

```
tutorize(
  input,
  output_dir = NULL,
  format = c("learnr", "quarto-live"),
  assessment = c("code", "mcq", "both"),
  overwrite = FALSE,
  language = c("en", "fr"),
  seed = NULL,
  verbose = TRUE,
  output_file = NULL,
  question_bank = NULL,
  mcq_source = c("inline", "bank", "mixed"),
  lint_strict = FALSE
)
```

Arguments

<code>input</code>	Path to source <code>‘.Rmd’</code> or <code>‘.qmd’</code> file.
<code>output_dir</code>	Optional output directory. Defaults to source directory.
<code>format</code>	Output format: <code>“learnr”</code> (default) or <code>“quarto-live”</code> .
<code>assessment</code>	Assessment mode: <code>“code”</code> , <code>“mcq”</code> , or <code>“both”</code> .
<code>overwrite</code>	Logical; overwrite existing output file?
<code>language</code>	Message/template language: <code>“en”</code> (default) or <code>“fr”</code> .
<code>seed</code>	Optional integer seed injected into setup chunk.
<code>verbose</code>	Logical; show progress and diagnostic messages?
<code>output_file</code>	Optional explicit output file path. Useful for wrappers.
<code>question_bank</code>	Optional path(s) or <code>‘tutorize_question_bank’</code> object.
<code>mcq_source</code>	MCQ generation source: <code>“inline”</code> , <code>“bank”</code> , <code>“mixed”</code> .
<code>lint_strict</code>	Logical; fail conversion when lint reports errors.

Value

A `‘tutorize_report’` object (invisibly).

Examples

```
## Not run:
rep <- tutorize("analysis.qmd", format = "learnr", assessment = "both")
print(rep)

## End(Not run)
```

validate_input	<i>Validate tutorizeR input arguments</i>
----------------	---

Description

Performs defensive checks for file path, format, assessment, and extension.

Usage

```
validate_input(
  input,
  format = c("learnr", "quarto-live"),
  assessment = c("code", "mcq", "both"),
  language = c("en", "fr")
)
```

Arguments

input	Path to source ‘.Rmd’ or ‘.qmd’ document.
format	Target output format (‘"learnr"’ or ‘"quarto-live"’).
assessment	Assessment mode (‘"code"’, ‘"mcq"’, or ‘"both"’).
language	Language for validation messages (‘"en"’ or ‘"fr"’).

Value

Invisibly returns ‘TRUE’ if validation succeeds.

validate_output	<i>Validate output path and overwrite strategy</i>
-----------------	--

Description

Validate output path and overwrite strategy

Usage

```
validate_output(
  output_file,
  input_file,
  overwrite = FALSE,
  language = c("en", "fr")
)
```

Arguments

output_file	Destination output file path.
input_file	Source file path.
overwrite	Logical; allow replacing an existing output.
language	Language for validation messages ("en" or "fr").

Value

Invisibly returns 'TRUE' if validation succeeds.

validate_question_bank

Validate a 'tutorize_question_bank' object

Description

Validate a 'tutorize_question_bank' object

Usage

```
validate_question_bank(bank, strict = TRUE, language = c("en", "fr"))
```

Arguments

bank	A question bank object from [load_question_bank()].
strict	Logical; if 'TRUE', stop on validation errors.
language	Language for validation messages ("en" or "fr").

Value

A 'tutorize_lint_report' object.

`write_tutorize_report` *Write a conversion report to JSON or YAML*

Description

Write a conversion report to JSON or YAML

Usage

```
write_tutorize_report(report, file, format = c("json", "yaml"), pretty = TRUE)
```

Arguments

<code>report</code>	A 'tutorize_report' object.
<code>file</code>	Output file path.
<code>format</code>	Output serialization format ("json" or "yaml").
<code>pretty</code>	Logical; pretty-print JSON output.

Value

Invisibly returns 'file'.

Index

check_tutorial, [2](#)
convert_folder, [3](#)
convert_to_tutorial, [4](#)

export_lms_manifest, [5](#)
export_tutorial_package, [6](#)

launch_tutorizeR_addin, [6](#)
launch_tutorizeR_folder_addin, [7](#)
launch_tutorizeR_preview_addin, [7](#)
lint_source, [8](#)
load_question_bank, [8](#)

print.tutorize_folder_report, [9](#)
print.tutorize_lint_report, [10](#)
print.tutorize_lms_manifest, [10](#)
print.tutorize_report, [11](#)

tr, [11](#)
tutorize, [12](#)

validate_input, [13](#)
validate_output, [13](#)
validate_question_bank, [14](#)

write_tutorize_report, [15](#)