

# Package ‘rintrojs’

August 10, 2022

**Type** Package

**Title** Wrapper for the 'Intro.js' Library

**Version** 0.3.2

**Description** A wrapper for the 'Intro.js' library (For more info: <<https://introjs.com/>>).  
This package makes it easy to include step-by-step introductions, and clickable hints in a 'Shiny' application. It supports both static introductions in the UI, and programmatic introductions from the server-side.

**License** AGPL-3

**Imports** shiny, jsonlite

**Depends** R (>= 3.0.0)

**RoxygenNote** 7.2.0

**URL** <https://github.com/carlganz/rintrojs>

**BugReports** <https://github.com/carlganz/rintrojs/issues>

**Suggests** testthat, covr

**NeedsCompilation** no

**Author** Carl Ganz [aut, cre],  
Afshin Mehrabani [ctb, cph] (intro.js in javascript/introjs)

**Maintainer** Carl Ganz <carlganz@gmail.com>

**Repository** CRAN

**Date/Publication** 2022-08-09 22:20:05 UTC

## R topics documented:

introBox . . . . .	2
introjs . . . . .	3
introjsUI . . . . .	4
readCallback . . . . .	5

<b>Index</b>	<b>7</b>
--------------	----------

---

`introBox`*Generate intro elements in UI*

---

### Description

Wrap `introBox` around elements you want to include in introduction. Use `data.step` to order the boxes and `data.intro` to specify the comment in the introduction

### Usage

```
introBox(  
  ...,  
  data.step,  
  data.intro,  
  data.hint,  
  data.position = c("bottom", "auto", "top", "left", "right", "bottom",  
    "bottom-left_aligned", "bottom-middle-aligned", "bottom-right-aligned", "auto")  
)
```

### Arguments

<code>...</code>	Elements in introduction element
<code>data.step</code>	a number indicating its spot in the order in the intro
<code>data.intro</code>	text for introduction
<code>data.hint</code>	text for clickable hints
<code>data.position</code>	position of intro

### See Also

[introjsUI\(\)](#) [introjs\(\)](#)

### Examples

```
## Not run:  
library(rintrojs)  
library(shiny)  
ui <- shinyUI(fluidPage(  
  introjsUI(), # must include in UI  
  mainPanel(  
    introBox(  
      tableOutput("mtcars"),  
      data.step = 1,  
      data.intro = "This is the table"  
    ),  
    introBox(  
      actionButton("btn", "Intro"),  
      data.step = 2,  
    )  
  )  
)
```

```
        data.intro = "This is the button"
      )
    )))
server <- shinyServer(function(input, output, session) {
  output$mtcars <- renderTable({
    head(mtcars)
  })
  observeEvent(input$btn,
    introjs(session))
})
# Run the application
shinyApp(ui = ui, server = server)

## End(Not run)
```

---

introjs

*Initiate intro.js*

---

## Description

Initiates an introduction via the intro.js library

## Usage

```
introjs(session, options = list(), events = list())
```

```
hintjs(session, options = list(), events = list())
```

## Arguments

session	the Shiny session object (from the server function of the Shiny app)
options	List of options to be passed to intro.js
events	List of text that are the body of a Javascript function. Must wrap in <code>I()</code>

## Note

For documentation on intro.js options and events, see <https://introjs.com/docs/>.

## See Also

[introjsUI\(\)](#) [introBox\(\)](#)

**Examples**

```
## Not run:
library(rintrosjs)
library(shiny)
ui <- shinyUI(fluidPage(
  introsjsUI(), # must include in UI
  mainPanel(
    introBox(
      tableOutput("mtcars"),
      data.step = 1,
      data.intro = "This is the table"
    ),
    introBox(
      actionButton("btn", "Intro"),
      data.step = 2,
      data.intro = "This is the button",
      data.hint = "Here is clue"
    )
  ))
server <- shinyServer(function(input, output, session) {

  hintjs(session, options = list("hintButtonLabel"="That was a hint"))

  output$mtcars <- renderTable({
    head(mtcars)
  })
  observeEvent(input$btn,
    introsjs(session, options = list("nextLabel"="Onwards and Upwards"),
      events = list("oncomplete"=I('alert("It is over"')'))))
})
# Run the application
shinyApp(ui = ui, server = server)

## End(Not run)
```

---

 introsjsUI

*Set up Shiny app to use intros.js*


---

**Description**

This function must be called from a Shiny app's UI in order to use the package.

**Usage**

```
introsjsUI(includeOnly = FALSE, cdn = FALSE, version = "3.2.1")
```

**Arguments**

`includeOnly`      Only include intros.js files. For users who will write their own javascript

cdn	Indicate whether to include intro.js files from CDN
version	Specify intro.js version to use from cdn

### Examples

```
## Not run:
library(rintrojs)
library(shiny)

shinyApp(
  ui = fluidPage(
    introjsUI(), # must include in UI
    actionButton("btn", "Click me")
  ),
  server = function(input, output, session) {
    observeEvent(input$btn, {
      intro <- data.frame(element="#btn",
                          intro="In Codd we trust")
      introjs(session, options = list(steps= intro))
    })
  }
)

## End(Not run)
```

---

readCallback	<i>Read a JS callback function into rintrojs</i>
--------------	--

---

### Description

Reads a JS callback function into rintrojs

### Usage

```
readCallback(funname = c("switchTabs"))
```

### Arguments

funname	The name of the function you want to use. Options include: <b>switchTabs</b> This function is intended to be passed to IntroJs's <b>onbeforechange method</b> . It will switch the currently active tab in your Shiny app to be the one containing the next element in your tour (this function is called by IntroJs right before it moves to the next element). Try running <code>shiny::runApp(system.file("example", "intro.js", package = "rintrojs"))</code> to see an example.
---------	---

### Value

A string containing the body of a callback function

**Examples**

```
## Not run:  
introjs(session, events = list(onbeforechange = readCallback("switchTabs")))  
  
## End(Not run)
```

# Index

hintjs(introjs), 3

I(), 3

introBox, 2

introBox(), 3

introjs, 3

introjs(), 2

introjsUI, 4

introjsUI(), 2, 3

readCallback, 5