# Package ‘iRcotofun’

December 16, 2015

**Type** Package

**Title** A Generator for Jeopardy-Like Quiz Games (HTML)

**Author** Meik Michalke [aut, cre]

**Maintainer** Meik Michalke <meik.michalke@hhu.de>

**Depends** R (>= 2.9.0),methods,base64enc,XiMpLe

**Description** Provides some functions and classes to build your own quiz with a few lines of code. The result is a single HTML file, i.e., the quiz is played using a web browser. The package was written to have fun at our famous DiagDiff Weihnachtsfeier ;-)

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyLoad** yes

**URL** http://reaktanz.de/?c=hacking&s=iRcotofun

**Version** 0.02-1

**Date** 2015-12-16

**RoxygenNote** 5.0.1

## R topics documented:

## `iRcotofun-package` *The iRcotofun Package*

### Description

A Generator for Jeopardy-Like Quiz Games (HTML).

### Details

| | |
|---|---|
| Package: | iRcotofun |
| Type: | Package |
| Version: | 0.02-1 |
| Date: | 2015-12-16 |
| Depends: | R (>= 2.9.0),methods,base64enc,XiMpLe |
| Encoding: | UTF-8 |
| License: | GPL (>= 3) |
| LazyLoad: | yes |
| URL: | http://reaktanz.de/?c=hacking&s=iRcotofun |

Provides some functions and classes to build your own quiz with a few lines of code. The result is a single HTML file, i.e., the quiz is played using a web browser. The package was written to have fun at our famous DiagDiff Weihnachtsfeier ;-)

### Author(s)

Meik Michalke

## `ask` *Ask a quiz question*

### Description

Creates an object of class `iRc_item`.

### Usage

```
ask(question, answer)
```

### Arguments

`question` Named character string to be used as the question. See details for valid names.

`answer` Character string to be used as the answer. See details for valid names.

## Details

Both parameters must be named character strings (or vectors). Currently these names are supported:

**text:** The value will be pasted as-is to the HTML document.

**img:** The value wil be treated as the relative path to an image.

**html:** The value must be a XiMpLe.node object.

## Value

An object of class `iRc_item`.

## Examples

```
q1 <- ask(
  question=list(text="What is Crichton's nick name for Chiana?"),
  answer=list(html=strong("Pip"))
)
q2 <- ask(
  question=list(text="What is Crichton's nick name for his gun?"),
  answer=list(html=strong("Winona"))
)
q3 <- ask(
  question=list(text="What is a famous quote from Rygel?"),
  answer=list(html=strong("Hail, prince of the obvious!"))
)
q4 <- ask(
  question=list(text="What is the Nebari Resistance fighting against?"),
  answer=list(html=strong("The Establishment"))
)
```

---

# `br` *HTML helper functions*

---

## Description

These functions can become handy to generate proper HTML code.

## Usage

```
br()

p(..., attrs = NULL)

span(..., attrs = NULL)

div(..., attrs = NULL)

th(..., attrs = NULL)
```

```
tr(..., attrs = NULL)

td(..., attrs = NULL)

table_(..., attrs = NULL)

a(..., attrs = NULL)

tbody(..., attrs = NULL)

strong(..., attrs = NULL)

em(..., attrs = NULL)

img(attrs = NULL, embed = TRUE)
```

## Arguments

`...` Child nodes of this node.

`attrs` A named list of attributes.

`embed` Logical, whether the specified `src` file should be ecoded using `base64encode` to ensure they are embedded into the HTML file. This keeps your presentation from breaking if you move it somewhere else. It will work with ordinary file paths as well, as long as all files remain in place.

## Details

If you want more control over the output of questions and answers, you can include HTML code, given it is in the form of `XiMpLe.node` objects.

You can of course use `XMLNode`, but it's abit quicker and more intuitive to rely on these helpers.

## Value

An object of class `XiMpLe.node`.

## Note

: Please note that `table_` ends with an underscore, to not confuse it with the `table` function.

---

# `category` *Create a quiz category*

---

## Description

Creates an object of class `iRc_category`.

### Usage

```
category(name, ...)
```

### Arguments

| | |
|---|---|
| `name` | Character string, name of the category. |
| `...` | Objects of class `iRc_item`. |

### Value

An object of class `iRc_category`.

### Examples

```
q1 <- ask(
  question=list(text="What is Crichton's nick name for Chiana?"),
  answer=list(html=strong("Pip"))
)
q2 <- ask(
  question=list(text="What is Crichton's nick name for his gun?"),
  answer=list(html=strong("Winona"))
)
q3 <- ask(
  question=list(text="What is a famous quote from Rygel?"),
  answer=list(html=strong("Hail, prince of the obvious!"))
)
q4 <- ask(
  question=list(text="What is the Nebari Resistance fighting against?"),
  answer=list(html=strong("The Establishment"))
)
# now make a category
farscape <- category(name="Farscape", q1, q2, q3, q4)
```

## `ircotofun` *Create a quiz*

### Description

Creates an object of class `XiMpLe.doc`, which in this case is an HTML document that can be written to a single file and opened in a web browser to get a quiz.

### Usage

```
ircotofun(..., points, file = NULL, title = "iRcotofun", sound = NULL,
  css = NULL, overwrite = FALSE, questions = FALSE)
```

## Arguments

`...` Objects of class `iRc_category`.

`points` A numeric vector defining the points for each item in all catgories.

`file` Character string, path to a file to write to.

`title` Character string, title of the quiz.

`sound` Character string, name of the sound file to use for background (if available).

`css` Character string, path to a custom CSS file if you don’t want to use the default.

`overwrite` Logical, whether existing files should be overwritten. Otherwise an error is thrown.

`questions` Logical, whether the questions should be shown (quiz style) or the answers (jeopardy style, default).

## Details

All categories must have the same number of items, and this number must also be identical to the values given as `points`. Also, if you use images, make sure that paths are valid.

## Value

An object of class `XiMpLe.doc`, or (if `file` is specified) no visible return value.

## Examples

```
q1 <- ask(
  question=list(text="What is Crichton's nick name for Chiana?”),
  answer=list(html=strong(”Pip”))
)
q2 <- ask(
  question=list(text="What is Crichton's nick name for his gun?”),
  answer=list(html=strong(”Winona”))
)
q3 <- ask(
  question=list(text="What is a famous quote from Rygel?”),
  answer=list(html=strong(”Hail, prince of the obvious!”))
)
q4 <- ask(
  question=list(text="What is the Nebari Resistance fighting against?”),
  answer=list(html=strong(”The Establishment”))
)

# make a category
farscape <- category(name="Farscape”, q1, q2, q3, q4)

# for the sake of demonstration, we'll create a quiz that shows
# the same category four times...
## Not run:
(output <- tempfile(fileext=".html”))
ircotofun(
  farscape,
  farscape,
  farscape,
  farscape,
  points=c(100,200,300,400),
  file=output,
  title="Best Science Fiction Shows Ever"
)

## End(Not run)
```

## iRc_category,-class *Class iRc_category*

### Description

Objects of this class represent one category in a iRcotofun game. They are combined to a game by the function ircotofun. To create categories, use the function category.

### Slots

name Character string, name of the category.

items A list of objects of class iRc_item.

## iRc_item,-class *Class iRc_item*

### Description

Objects of this class represent one item in a iRcotofun game. They are combined to a category in objects of class iRc_Category. To freate items, use the function ask.

### Details

Both slots must be named character strings (or vectors). Currently these names are supported:

**text:** The value will be pasted as-is to the HTML document.

**img:** The value will be treated as the relative path to an image.

**html:** The value must be a XiMpLe.node object.

### Slots

question Named list to be used as the question. See details for valid names.

answer Named list to be used as the answer. See details for valid names.

## titlepage *Generate a title page*

### Description

In case you would like to show off with a huge show, use this function to generate a title page that plays some intro music, before you open the actual quiz.

### Usage

```
titlepage(..., quizfile, file = NULL, title = "iRcotofun", sound = NULL,
  css = NULL, overwrite = FALSE)
```

### Arguments

| | |
|---|---|
| `...` | Objects of class `XiMpLe.node`, the page content. |
| `quizfile` | Character string, full path to the actual quiz file. The resulting title page will use a relative link if both files are in the same directory, so it's easy to move both afterwards. |
| `file` | Character string, path to a file to write to. |
| `title` | Character string, used as the window title. |
| `sound` | Character string, name of the sound file to use for automatic background (if available). |
| `css` | Character string, path to a custom CSS file if you don't want to use the default. |
| `overwrite` | Logical, whether existing files should be overwritten. Otherwise an error is thrown. |

### Value

An object of class `XiMpLe.doc`, or (if `file` is specified) no visible return value.

### Examples

```
## Not run:
# for a logo you'll probably get best results with scalable SVG
logo <- div(
  img(
    attrs=list(
      src=file.path("path", "to", "myQuiz.svg"),
      class="image",
      style="margin-top: 10%;"
    )
  ),
  attrs=list(class="imagediv")
)

# finally, write the title page
```

```
titlepage(
  logo,
  quizfile=file.path("path", "to", "myQuiz.html"),
  file=file.path("path", "to", "index.html"),
  sound=file.path("path", "to", "myQuiz_main_title.ogg"),
  overwrite=TRUE
)

## End(Not run)
```

# Index