Package ‘cowsay’

March 8, 2023

Title  Messages, Warnings, Strings with Ascii Animals

Description  Allows printing of character strings as messages/warnings/etc.
              with ASCII animals, including cats, cows, frogs, chickens, ghosts,
              and more.

Version  0.8.2

License  MIT + file LICENSE

URL  https://github.com/sckott/cowsay

BugReports  https://github.com/sckott/cowsay/issues

Encoding  UTF-8

Language  en-US

VignetteBuilder  knitr

Imports  crayon, fortunes, rmsfact

Suggests  curl, jsonlite, knitr, multicolor, rmarkdown, testthat

RoxygenNote  7.2.3

NeedsCompilation  no

Author  Scott Chamberlain [aut, cre],
        Amanda Dobbyn [aut],
        Tyler Rinker [ctb],
        Thomas Leeper [ctb],
        Noam Ross [ctb],
        Rich FitzJohn [ctb],
        Carson Sievert [ctb],
        Kiyoko Gotanda [ctb],
        Andy Teucher [ctb],
        Karl Broman [ctb],
        Franz-Sebastian Krah [ctb],
        Lucy D'Agostino McGowan [ctb],
        Guangchuang Yu [ctb],
        Philipp Boersch-Supan [ctb],
        Andreas Brandmaier [ctb],
        Marion Louveaux [ctb]
Description

Named vector of animals

Usage

animals

Format

An object of class character of length 49.

Details

animals is a named character vector of animals, with each element a character string of variable length specifying an ASCII animal. Note that some have unicode characters that won’t play well on some operating systems.

Examples

animals['cow']
animals['clippy']
animals[['clippy']]
Description

If you are familiar with cowsay on the cli, then you know what this is, but for R. If not, read below. Why? Why not?

Details

Contributors:

• Amanda Dobbyn
• Andreas Brandmaier
• Andy Teucher
• Carson Sievert
• Franz-Sebastian Krah
• Guangchuang Yu
• Karl Broman
• Kiyoko Gotanda
• Lucy D’Agostino McGowan
• Noam Ross
• Paolo Sonego
• Philipp Boersch-Supan
• Rich FitzJohn
• Scott Chamberlain
• Thomas Leeper
• Tyler Rinker
• Marion Louveaux

There is only one function in the cowsay package. Simplicity for the win.

Say something:
say(‘hello world!’)
**endless_horse**  
*Endless horse*

**Description**

Each time you press enter, the horse keeps going...and going...

**Usage**

```r
endless_horse(
  what = "Hello world!",
  endless = TRUE,
  wait = 0.5,
  what_color = NULL,
  horse_color = NULL
)
```

**Arguments**

- **what** *(character)* What do you want to say? See details.
- **endless** *(logical)* Should horse be enless, you better say yes. Default: TRUE
- **wait** How long to wait between leg segments (time grows geometrically after the first iteration in order to keep the horse on screen for a while, but it will keep going forever. Or until you hit escape/Ctrl-C depending on your platform).
- **what_color** *(character or crayon function)* A crayon-suported text color or crayon style function to color what. You might try colors() or ?rgb for ideas.
- **horse_color** *(character or crayon function)* A crayon-suported text color or crayon style function to color your steed.

**Examples**

```r
## Not run:
endless_horse()
endless_horse(endless = FALSE)
## End(Not run)
```

**say**  
*Sling messages and warnings with flair*

**Description**

Sling messages and warnings with flair
Usage
say(
  what = "Hello world!",
  by = "cat",
  type = NULL,
  what_color = NULL,
  by_color = NULL,
  length = 18,
  fortune = NULL,
  ...
)

Arguments
what (character) What do you want to say? See Details.
by (character) Type of thing, one of cow, chicken, chuck, clippy, poop, bigcat, ant, pumpkin, ghost, spider, rabbit, pig, snowman, frog, hypnotoad, shortcat, longcat, fish, signbunny, facecat, behindcat, stretchycat, anxiouscat, longtailcat, cat, trilobite, shark, grumpycat, smallcat, yoda, mushroom, endlesshorse, bat, bat2, turkey, monkey, daemon, duckling, duck, owl, squirrel, squirrel2, goldfish, alligator, stegosaurus, whale, wolf, or rms for Richard Stallman. Alternatively, use "random" to have your message spoken by a random character. We use match.arg() internally, so you can use unique parts of words that don’t conflict with others, like "g" for "ghost" because there’s no other animal that starts with "g".
type (character) One of message (default), warning, print (default in non-interactive mode), or string (returns string). If multiple colors are supplied to what_color or by_color, type cannot be warning. (This is a limitation of the multicolor package :/.) If run in non-interactive mode default type is print, so that output goes to stdout rather than stderr, where messages and warnings go.
what_color (character or crayon function) One or more crayon-supported text color(s) or crayon style function to color what. You might try colors() or ?rgb for ideas. Use "rainbow" for c("red", "orange", "yellow", "green", "blue", "purple").
by_color (character or crayon function) One or more crayon-supported text color(s) or crayon style function to color who. Use "rainbow" for c("red", "orange", "yellow", "green", "blue", "purple").
length (integer) Length of longcat. Ignored if other animals used.
fortune An integer specifying the row number of fortunes.data. Alternatively which can be a character and grep is used to try to find a suitable row.
...
Further args passed on to fortunes::fortune()

what
You can put in any phrase you like to the what parameter, OR you can type in one of a few special phrases that do particular things. They are:

• "catfact": A random cat fact from https://catfact.ninja
- "fortune": A random quote from an R coder, from fortunes library
- "time": Print the current time
- "rms": Prints a random 'fact' about Richard Stallman from the `rmsfact::rmsfact()` package. Best paired with `by = "rms"`.
- The fillerama API was an option ("arresteddevelopment", "doctorwho", "dexter", "futurama", "hollygrail", "simpsons", "starwars", "loremipsum"), a text generator with quotes from shows and movies. However, the API is currently down so you cannot use this option anymore.

**by**

Note that if you choose `by='hypnotoad'` the quote is forced to be, as you could imagine, 'All Glory to the HYPNO TOAD!' For reference see [http://knowyourmeme.com/memes/hypnotoad](http://knowyourmeme.com/memes/hypnotoad)

signbunny: It’s not for sure known who invented signbunny, but this article [http://www.vox.com/2014/9/18/6331753/signbunny-meme-explained](http://www.vox.com/2014/9/18/6331753/signbunny-meme-explained) thinks they found the first use in this tweet: [https://twitter.com/wei_bluebear/status/329101645780770817](https://twitter.com/wei_bluebear/status/329101645780770817)

trilobite: from [http://www.retrojunkie.com/asciiart/animals/dinos.htm](http://www.retrojunkie.com/asciiart/animals/dinos.htm) (site down though)

Note to Windows users: there are some animals (shortcat, longcat, fish, signbunny, stretchycat, anxiouscat, longtailcat, grumpycat, mushroom) that are not available because they use non-ASCII characters that don’t display properly in R on Windows.

### Examples

```r
say()
say("what")
say("time")

if (requireNamespace("multicolor", quietly=TRUE) && requireNamespace("jsonlite", quietly=TRUE)) {
  say("meow", "cat", what_color = "blue")
say("time", "poop", by_color = "cyan", what_color = "pink")

library(jsonlite)
say("hello world", 
   by = "hypnotoad",
   what_color = c("royalblue1", "tomato2"),
   by_color = c("rainbow", "rainbow"))
say("whats up", 
   what_color = crayon::cyan$bgMagenta,
   by_color = c("salmon1", "springgreen"))
}

say("who you callin chicken", "chicken")
say("ain't that some shit", "poop")
say("icanhazpdf?", "cat")
say("boo!", "pumpkin")
say("hot diggity", "frog")
say("fortune", "spider")
say("fortune", "facecat")
say("fortune", "behindcat")
say("fortune", "smallcat")
```
say("fortune", "monkey")
say("fortune", "egret")
say("rms", "rms")

# Vary type of output, default calls message()
say("hell no!")
say("hell no!", type="warning")
say("hell no!", type="string")

# Using fortunes
say(what="fortune")
## you don't have to pass anything to the 'what' parameter if 'fortune' is not null
say(fortune=10)
say(fortune=100)
say(fortune='whatever')
say(fortune=7)
say(fortune=45)

# Using catfacts
# say("catfact", "cat")

# The hypnotoad
say(by="hypnotoad")

# Trilobite
say(by='trilobite')

# Shark
say('Q: What do you call a solitary shark?\nA: A lone shark', by='shark')

# Buffalo
say('Q: What do you call a single buffalo?\nA: A buffalonely', by='buffalo')

# Clippy
say(fortune=59, by="clippy")
Index

* datasets
  animals, 2

animals, 2

cowsay, 3

endless_horse, 4

fortunes::fortune(), 5

match.arg(), 5

rmsfact::rmsfact(), 6

say, 4