

# Package ‘keypress’

December 11, 2023

**Title** Wait for a Key Press in a Terminal

**Version** 1.3.1

**Author** Gábor Csárdi [aut, cre], Jon Griffiths [aut]

**Maintainer** Gábor Csárdi <csardi.gabor@gmail.com>

**Description** Wait for a single key press at the 'R' prompt.  
This works in terminals, but does not currently work in the 'Windows' 'GUI', the 'OS X' 'GUI' ('R.app'), in 'Emacs' 'ESS', in an 'Emacs' shell buffer or in 'R Studio'. In these cases 'keypress' stops with an error message.

**License** MIT + file LICENSE

**URL** <https://github.com/gaborcsardi/keypress#readme>

**BugReports** <https://github.com/gaborcsardi/keypress/issues>

**RoxygenNote** 7.1.2

**Encoding** UTF-8

**Suggests** covr

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2023-12-11 03:40:03 UTC

## R topics documented:

has_keypress_support . . . . .	2
keypress . . . . .	3
without_echo . . . . .	4
<b>Index</b>	<b>5</b>

has\_keypress\_support *Check if the current platform/terminal supports reading single keys.*

---

### Description

Check if the current platform/terminal supports reading single keys.

### Usage

```
has_keypress_support()
```

### Details

Supported platforms:

- Terminals in Windows and Unix.
- RStudio terminal.

Not supported:

- RStudio (if not in the RStudio terminal).
- R.app on macOS.
- Rgui on Windows.
- Emacs ESS.
- Others.

### Value

Whether there is support for waiting for individual keypresses.

### See Also

Other keypress function: [keypress\(\)](#)

### Examples

```
has_keypress_support()
```

---

`keypress`*Read a single keypress at the terminal*

---

## Description

It currently only works at Linux/Unix and OSX terminals, and at the Windows command line. see [has\\_keypress\\_support](#).

## Usage

```
keypress(block = TRUE)
```

## Arguments

`block` Whether to wait for a key press, if there is none available now.

## Details

The following special keys are supported:

- Arrow keys: 'up', 'down', 'right', 'left'.
- Function keys: from 'f1' to 'f12'.
- Others: 'home', 'end', 'insert', 'delete', 'pageup', 'pagedown', 'tab', 'enter', 'backspace' (same as 'delete' on OSX keyboards), 'escape'.
- Control with one of the following keys: 'a', 'b', 'c', 'd', 'e', 'f', 'h', 'k', 'l', 'n', 'p', 't', 'u', 'w'.

## Value

The key pressed, a character scalar. For non-blocking reads NA is returned if no keys are available.

## See Also

Other keypress function: [has\\_keypress\\_support\(\)](#)

## Examples

```
x <- keypress()
cat("You pressed key", x, "\n")
```

---

without_echo	<i>Call a function with echo suppressed</i>
--------------	---

---

**Description**

For Linux/Unix and OSX terminals, suppress key echoes from the terminal.

**Usage**

```
without_echo(expr)
```

**Arguments**

expr	Expression to evaluate without terminal echo.
------	---

**Details**

You will need to set `echo` to `FALSE` if you want the R script to handle all of the keypress-related behaviour, without any keys being echoed in the terminal by the operating system, when using non-blocking keypress. This is not necessary when running in a Windows command prompt, and will be safely ignored.

# Index

- \* **keypress function**

- has\_keypress\_support, [2](#)

- keypress, [3](#)

- \* **terminal functions**

- without\_echo, [4](#)

has\_keypress\_support, [2](#), [3](#)

keypress, [2](#), [3](#)

without\_echo, [4](#)