

# Package ‘argparse’

November 26, 2021

**Type** Package

**Title** Command Line Optional and Positional Argument Parser

**Version** 2.1.3

**Description** A command line parser to  
be used with Rscript to write `#!" shebang scripts that gracefully  
accept positional and optional arguments and automatically generate usage.

**License** GPL (>= 2)

**Copyright** See file (inst/)COPYRIGHTS.

**URL** <https://github.com/trevorld/r-argparse>

**BugReports** <https://github.com/trevorld/r-argparse/issues>

**LazyLoad** yes

**Imports** R6, findpython, jsonlite

**SystemRequirements** python (>= 3.2)

**Suggests** knitr (>= 1.15.19), rmarkdown, testthat

**VignetteBuilder** knitr

**RoxygenNote** 7.1.1

**Encoding** UTF-8

**NeedsCompilation** no

**Author** Trevor L Davis [aut, cre],  
Allen Day [ctb] (Some documentation and examples ported from the getopt  
package.),  
Python Software Foundation [ctb] (Some documentation from the optparse  
Python module.),  
Paul Newell [ctb]

**Maintainer** Trevor L Davis <trevor.l.davis@gmail.com>

**Repository** CRAN

**Date/Publication** 2021-11-26 14:20:02 UTC

## R topics documented:

ArgumentParser . . . . .	2
<b>Index</b>	<b>4</b>

---

ArgumentParser	<i>Create a command line parser</i>
----------------	-------------------------------------

---

### Description

ArgumentParser creates a parser object that acts as a wrapper to Python's argparse module

### Usage

```
ArgumentParser(..., python_cmd = NULL)
```

### Arguments

...	Arguments cleaned and passed to Python's argparse.ArgumentParser()
python_cmd	Python executable for argparse to use. Must have argparse and json modules (automatically included Python 2.7 and 3.2+). If you need Unicode argument support then you must use Python 3.0+. Default will be to use findpython package to find suitable Python binary.

### Value

ArgumentParser returns a parser object which contains an add\_argument function to add arguments to the parser, a parse\_args function to parse command line arguments into a list, a print\_help and print\_usage function to print usage information. See code examples, package vignette, and corresponding python module for more information on how to use it.

### References

Python's argparse library, which this package is based on, is described here: <https://docs.python.org/3/library/argparse.html>

### Examples

```
if (argparse:::detects_python()) {
  parser <- ArgumentParser(description='Process some integers')
  parser$add_argument('integers', metavar='N', type = "integer", nargs='+',
    help='an integer for the accumulator')
  parser$add_argument('--sum', dest='accumulate', action='store_const',
    const='sum', default='max',
    help='sum the integers (default: find the max)')
  parser$print_help()
  # default args for ArgumentParser()$parse_args are commandArgs(TRUE)
```

```
# which is what you'd want for an Rscript but not for interactive use
args <- parser$parse_args(c("--sum", "1", "2", "3"))
accumulate_fn <- get(args$accumulate)
print(accumulate_fn(args$integers))
}
```

# Index

ArgumentParser, [2](#)