# Package 'RJalaliDate'

January 20, 2025

**Title** Handling Jalali Date (Persian / Solar Hijri)

**Version** 0.1.0

**Description** Jalali calendar, or solar Hijri, is calendar of Iran and Afghanistan (<https://en.wikipedia.org/wiki/Solar_Hijri_calendar>). This package is designed to working with Jalali date. For this purpose, It defines JalaliDate class that is similar to Date class.

**License** GPL (>= 3)

**Depends** R (>= 2.10)

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Imports** lubridate (>= 1.9.3), settings (>= 0.2.7), stringi (>= 1.8.3), stringr (>= 1.5.1)

**Suggests** testthat (>= 3.0.0)

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Hosein Rabiee [aut, cre, cph] (<https://orcid.org/0000-0002-9379-5476>)

**Maintainer** Hosein Rabiee <hosein.rabiee@hotmail.com>

**Repository** CRAN

**Date/Publication** 2024-09-11 16:10:02 UTC

## Contents

---

`*.JalaliDate` *Operators*

---

## Description

Perform arithmetic operations

## Usage

```
## S3 method for class 'JalaliDate'
x * y
```

## Arguments

| | |
|---|---|
| x | JalaliDate or numeric |
| y | JalaliDate or numeric |

## Details

Only +, - operators work with JalaliDate objects in some cases. If each of the two arguments are JalaliDate, the '-' operator calculates the distance of two dates (see diffdate).

## Value

JalaliDate

## Examples

```
JalaliDate("1395/10/11") + 1
# [1] "1395/10/12"
JalaliDate("1403/08/10") - 367
# [1] "1402/08/08"
JalaliDate("1403/09/10") - JalaliDate("1403/08/10")
# [1] 30
```

---

`+.JalaliDate` *Operators*

---

## Description

Perform arithmetic operations

## Usage

```
## S3 method for class 'JalaliDate'
x + y
```

## Arguments

| | |
|---|---|
| x | JalaliDate or numeric |
| y | JalaliDate or numeric |

## Details

Only +, - operators work with JalaliDate objects in some cases. If each of the two arguments are JalaliDate, the '-' operator calculates the distance of two dates (see diffdate).

## Value

JalaliDate

## Examples

```
JalaliDate("1395/10/11") + 1
# [1] "1395/10/12"
JalaliDate("1403/08/10") - 367
# [1] "1402/08/08"
JalaliDate("1403/09/10") - JalaliDate("1403/08/10")
# [1] 30
```

## -.JalaliDate *Operators*

### Description

Perform arithmetic operations

### Usage

```
## S3 method for class 'JalaliDate'
x - y
```

### Arguments

| | |
|---|---|
| x | JalaliDate or numeric |
| y | JalaliDate or numeric |

### Details

Only +, - operators work with JalaliDate objects in some cases. If each of the two arguments are JalaliDate, the '-' operator calculates the distance of two dates (see diffdate).

### Value

JalaliDate

### Examples

```
JalaliDate("1395/10/11") + 1
# [1] "1395/10/12"
JalaliDate("1403/08/10") - 367
# [1] "1402/08/08"
JalaliDate("1403/09/10") - JalaliDate("1403/08/10")
# [1] 30
```

## /.JalaliDate *Operators*

### Description

Perform arithmetic operations

### Usage

```
## S3 method for class 'JalaliDate'
x / y
```

### Arguments

| | |
|---|---|
| x | JalaliDate or numeric |
| y | JalaliDate or numeric |

### Details

Only +, - operators work with JalaliDate objects in some cases. If each of the two arguments are JalaliDate, the '-' operator calculates the distance of two dates (see diffdate).

### Value

JalaliDate

### Examples

```
JalaliDate("1395/10/11") + 1
# [1] "1395/10/12"
JalaliDate("1403/08/10") - 367
# [1] "1402/08/08"
JalaliDate("1403/09/10") - JalaliDate("1403/08/10")
# [1] 30
```

---

## as.character.JalaliDate

*data type conversion*

---

### Description

convert JalaliDate to character, Date, or list

### Usage

```
## S3 method for class 'JalaliDate'
as.character(
  x,
  format = "A",
  separator = jdate_options("DEFAULT_SEPARATOR"),
  ...
)

## S3 method for class 'JalaliDate'
as.Date(x, ...)

## S3 method for class 'JalaliDate'
as.list(x, ...)
```

### Arguments

| | |
|---|---|
| x | JalaliDate object |
| format | character. One of c("A", "B", "C") elements:<br>• "A": simple, combining Jalali date with DEFAULT_SEPARATOR<br>• "B": (year) (month_name) (day)<br>• "C": (year) (month_name) (day) (day_of_week) |
| separator | character. One of VALID_SEPARATORS (see jdopt_get_options) that converts Jalali date elements to character |
| ... | • "as.character": The ... argument is used to pass options that overrides current options (see examples).<br>• "other": future usage |

### Value

character, Date, or list

### Examples

```
as.character(JalaliDate(1), separator= "+", VALID_SEPARATORS = c("+"))
```

---

## change_date_separator *change separator*

---

### Description

change the Jalali date character separators. For invalid inputs, returns input without change

### Usage

```
change_date_separator(date_char, new_separator, ...)
```

### Arguments

| | |
|---|---|
| date_char | Jalali date (character) |
| new_separator | character (valid separator) |
| ... | passing options ("MIN_YEAR" and/or "MAX_YEAR") to override package options (jdopt_get_options) |

### Value

Jalali date character

### Examples

```
change_date_separator(c("1350/01/02", "14021220", "1402/1/40"), "+", VALID_SEPARATORS=c("+"))
# [1] "1350+01+02" "1402+12+20" "1402/1/40"
```

## diffdate *distance of two JalaliDate*

### Description

calculate distance of two JalaliDate, that is, subtracts values of two JalaliDate and return

### Usage

```
diffdate(x, y)
```

### Arguments

| | |
|---|---|
| x | JalaliDate object |
| y | JalaliDate object |

### Value

double

### Examples

```
x <- JalaliDate(c(1, 2))
y <- JalaliDate(c(10, 12, 13, 20, 50))
diffdate(x,y)
# [1]  -9 -10  NA  NA  NA
# Warning message:
#  In diff.JalaliDate(x, y) : The length of two vectors aren't equal!
```

## is_gregorian_leap_year *check Gregorian leap year*

### Description

check if a Gregorian year is leap year

### Usage

```
is_gregorian_leap_year(year)
```

### Arguments

| | |
|---|---|
| year | double |

## Value

logical

## Examples

```
is_gregorian_leap_year(c(2000, 2001, 2002))
# [1] TRUE FALSE FALSE
```

---

# is_jalali_leap_year *check Jalali leap year*

---

## Description

check if a Jalali year is leap year

## Usage

```
is_jalali_leap_year(year)
```

## Arguments

| | |
|---|---|
| year | double |

## Details

for details of calculation see https://learn.microsoft.com/en-us/dotnet/fundamentals/runtime-libraries/system-globalization-persiancalendar

## Value

logical

## Examples

```
is_jalali_leap_year(c(1402, 1403, 1404))
# [1] FALSE  TRUE FALSE
```

`is_valid_date_elements` *validation JalaliDate elements*

## Description

check validation of Jalali date elements and returns results

## Usage

```
is_valid_date_elements(year, month, day, ...)
```

## Arguments

| | |
|---|---|
| `year` | double |
| `month` | double |
| `day` | double |
| `...` | passing options ("MIN_YEAR" and/or "MAX_YEAR") to override package options (jdopt_get_options) |

## Details

type of message are:

- "y": year is not valid
- "m": month is not valid
- "d1": day must be between 0 to 31
- "d2": in the last 6 months of the year, the day should not be more than 30
- "d3": in leap years, the day should not be 30

## Value

list of validation result and related message

## Examples

```
is_valid_date_elements(c(1402, 1000), c(12, 13), c(10, 11), MIN_YEAR=100)
# $result
# [1]  TRUE FALSE
#
# $message
# [1] ""  "m"
```

## is_valid_jalali_date_char *validation JalaliDate character*

### Description

check validation of Jalali date in form of character and returns results

### Usage

```
is_valid_jalali_date_char(date_char, return_all_assessment_data = TRUE, ...)
```

### Arguments

| | |
|---|---|
| date_char | character |
| return_all_assessment_data | logical , if it is FALSE only return validation result (logical vector) |
| ... | passing options ("MIN_YEAR" and/or "MAX_YEAR") to override package options (jdopt_get_options) |

### Value

list or logical, based on second argument

### Examples

```
is_valid_jalali_date_char("1402/10/15", FALSE)
# [1] TRUE
```

## is_valid_separator *check separator*

### Description

Checks whether a separator can be among the set of valid separators

### Usage

```
is_valid_separator(separator)
```

### Arguments

| | |
|---|---|
| separator | character |

## Value

list of validation result and related message

## Examples

```
is_valid_separator("+")
# $result
# [1] TRUE
#
# $message
# [1] ""

is_valid_separator("+/")
# $result
# [1] FALSE
#
# $message
# [1] "The number of character of the separator must be 0 or 1!"
```

---

`JalaliDate` *JalaliDate object constructor*

---

## Description

Creates an instance of JalaliDate object by S3 system.

## Usage

```
JalaliDate(x, ...)
```

## Arguments

| | |
|---|---|
| `x` | object (double, integer, Date, character, list) list' argument could be named like JalaliDate(list(y=1375, m=1, d=2)) |
| `...` | ... argument is used to pass options that overrides current options (see examples). |

## Details

JalaliDate object is designed as 'base::Date' to handle Jalali (solar Hijri) date that is calendar of Iran and Afghanistan. Like Date, the JalaliDate information is stored in the form of a 'double' and is converted to another data type when necessary using the corresponding algorithm. The base day (value = 0) is "1375/01/01". Calculation of leap year is like Microsoft .Net method (33 years cycles). If the argument value is not valid at the time of conversion, it will be replaced with NA and a message will be sent in this regard (see examples).

## Value

JalaliDate object

## Examples

```
JalaliDate(c(1, NA_real_, 2))
# [1] "1375/01/02" NA "1375/01/03"

JalaliDate(as.Date("2024-01-01"))
# [1] "1402/10/11"

JalaliDate(1.5)
# [1] "1375/01/02"

JalaliDate(c("1375/01/01", NA))
# [1] "1375/01/01" NA

# with warning
JalaliDate(c("1375/01/03", "1375/0201", ""))
# [1] "1375/01/03" NA NA
# Warning message:
#   NAs introduced by validation.

# year is out of default options range (1200-1500)
JalaliDate(list(9998,1,1))
# [1] NA
# Warning message:
#   NAs introduced by validation.

JalaliDate(c("1380/01/01", "9998/10/15"), MAX_YEAR=9999)
# [1] "1380/01/01" "9998/10/15"

tmp<- c("1375+01+01", "1390/02/02", "2000 02 02", "0100_02_02")
JalaliDate(tmp, VALID_SEPARATORS=c("+", "_", " ", "/"), MAX_YEAR=9999, MIN_YEAR = 0)
# [1] "1375/01/01" "1390/02/02" "2000/02/02" "0100/02/02"
```

---

`jalali_year_weeks` *list of a year weeks*

---

## Description

create a data.frame consists start and end days of a year weeks

## Usage

```
jalali_year_weeks(year, ...)
```

## Arguments

| | |
|---|---|
| `year` | double |
| `...` | passing options ("MIN_YEAR" and/or "MAX_YEAR") to override package options (jdopt_get_options) |

### Value

data.frame

### Examples

```
jalali_year_weeks(1402)
 #    week     f      l
 #1     1 1402/01/01 1402/01/04
 #2     2 1402/01/05 1402/01/11
 #.............................
 #52   52 1402/12/19 1402/12/25
 #53   53 1402/12/26 1402/12/29
```

---

## jdopt_get_options *return package options*

---

### Description

get a list of the package options

### Usage

```
jdopt_get_options()
```

### Details

Package Options have four parts: DEFAULT_SEPARATOR, VALID_SEPARATORS, MIN_YEAR, and MAX_YEAR. Options are used in validation and type conversion. For example, if 'VALID_SEPARATORS' part of options include c("/", " "), validation of "1390-01-01" return FALSE, because separator of this Jalali date is "-" that does not belong to valid separators set. By default, the conversion of "1000/10/11" to JalaliDate would be failed, because year of Jalali date should be between 1200 and 1500. By setting 'DEFAULT_SEPARATOR' to "_", the result of conversion of JalaliDate(1) to character will be "1375_01_02".

### Value

options list that includes DEFAULT_SEPARATOR, VALID_SEPARATORS, MIN_YEAR, MAX_YEAR

### Examples

```
jdopt_get_options()
# $DEFAULT_SEPARATOR
# [1] "/"
#
# $VALID_SEPARATORS
# [1] ""  "-" "/"
#
# $MIN_YEAR
# [1] 1200
```

```
#
# $MAX_YEAR
#
# [1] 1500
```

---

`jdopt_reset` *reset options to initial values*

---

## Description

return options value to factory settings

## Usage

```
jdopt_reset()
```

## Details

The initial values, or factory settings, are: DEFAULT_SEPARATOR = "/" , VALID_SEPARATORS = c("", "-", "/"), MIN_YEAR = 1200L, and MAX_YEAR = 1500L

## Value

options list that includes DEFAULT_SEPARATOR, VALID_SEPARATORS, MIN_YEAR, MAX_YEAR

## Examples

```
res <- jdopt_reset()
res

#$DEFAULT_SEPARATOR
#[1] "/"
#
#$VALID_SEPARATORS
#[1] ""  "-" "/"
#
#$MIN_YEAR
#[1] 1200
#
#$MAX_YEAR
#[1] 1500
```

## jdopt_set_default_separator

*Specify the default separator*

### Description

specifying one of valid separators as default separator

### Usage

```
jdopt_set_default_separator(separator)
```

### Arguments

| | |
|---|---|
| separator | character |

### Details

The default separator (where initially is "/") has several uses. For example, to print JalaliDate: JalaliDate(list(1375, 1, 2)) that display "1375/01/02". Selected separator must belong to 'VALID_SEPARATORS' set, otherwise an error would be raised.

### Value

options list that includes DEFAULT_SEPARATOR, VALID_SEPARATORS, MIN_YEAR, MAX_YEAR

### Examples

```
JalaliDate(Sys.Date())
# [1] "1403/04/29"
jdopt_set_default_separator("-")
JalaliDate(Sys.Date())
# [1] "1403-04-29"
```

## jdopt_set_min_max_year

*setting range of valid year*

### Description

Determining the minimum and maximum valid value of Jalali date year

### Usage

```
jdopt_set_min_max_year(min_year, max_year)
```

### Arguments

| | |
|---|---|
| min_year | integer |
| max_year | integer |

### Details

'min_year' must be equal or lower than 'max_year' and both must be integer. Minimum value of 'min_year' is 0 and maximum value of 'max_year' is 9999.

### Value

options list that includes DEFAULT_SEPARATOR, VALID_SEPARATORS, MIN_YEAR, MAX_YEAR

### Examples

```
JalaliDate(list(1000,1,1))
# [1] NA
jdopt_set_min_max_year(100L, 2000L)
JalaliDate(list(1000,1,1))
# [1] "1000/01/01"
```

---

## jdopt_set_valid_separators

*setting new valid separators*

---

### Description

changing existing set of valid separators and defining a new set

### Usage

```
jdopt_set_valid_separators(valid_separators)
```

### Arguments

| | |
|---|---|
| valid_separators | character |

### Details

Argument of the function is a character vector that each of elements has length of 0 or 1. After changing 'VALID_SEPARATORS', if current 'DEFAULT_SEPARATOR' doesn't belong to new 'VALID_SEPARATORS', first element of new 'VALID_SEPARATORS' (after sorting) will be set as 'DEFAULT_SEPARATOR' and a message will be displayed.

### Value

options list or warning

## Examples

```
jdopt_reset()
res <- jdopt_set_valid_separators(c("+", "$"))
 #After setting new valid separators, the default separator was changed automatically!

res
# $DEFAULT_SEPARATOR
# [1] "$"
#
# $VALID_SEPARATORS
# [1] "$" "+"
#
# $MIN_YEAR
# [1] 1200
#
# $MAX_YEAR
# [1] 1500
```

---

`Operators` *Operators*

---

## Description

Perform arithmetic operations

## Usage

```
Operators(x, y)
```

## Arguments

| | |
|---|---|
| x | JalaliDate or numeric |
| y | JalaliDate or numeric |

## Details

Only +, - operators work with JalaliDate objects in some cases. If each of the two arguments are JalaliDate, the '-' operator calculates the distance of two dates (see diffdate).

## Value

JalaliDate

## Examples

```
JalaliDate("1395/10/11") + 1
# [1] "1395/10/12"
JalaliDate("1403/08/10") - 367
# [1] "1402/08/08"
JalaliDate("1403/09/10") - JalaliDate("1403/08/10")
# [1] 30
```

---

# print.JalaliDate *print*

---

## Description

print

## Usage

```
## S3 method for class 'JalaliDate'
print(x, ...)
```

## Arguments

| | |
|---|---|
| x | JalaliDate |
| ... | for future usages |

## Value

display

## Examples

```
print(JalaliDate(1))
# [1] "1375/01/02"
```

---

# today.JalaliDate *today as Jalali*

---

## Description

return JalaliDate object of today

## Usage

```
today.JalaliDate()
```

### Value

JalaliDate object

### Examples

```
today.JalaliDate()
# [1] "1403/04/31"
```

---

## weekdays.JalaliDate *day of week*

---

### Description

return the day of week in Persian #'

### Usage

```
## S3 method for class 'JalaliDate'
weekdays(x, abbreviate = NULL)
```

### Arguments

| | |
|---|---|
| x | JalaliDate object |
| abbreviate | not applicable in Persian language |

### Value

character

### Examples

```
weekdays(JalaliDate(1))
#[1] `r stringi::stri_unescape_unicode("\u067E\u0646\u062C\u0020\u0634\u0646\u0628\u0647")`
```

---

## yearweek *number of week*

---

### Description

It shows which week of the year the desired date is.

### Usage

```
yearweek(x, ...)
```

### Arguments

| | |
|---|---|
| x | JalaliDate object |
| ... | for future usage |

### Value

list of current and last week number and label

### Examples

```
yearweek(JalaliDate(321))
# $week_number
# [1] "47"
#
# $week_label
# [1] "75W47"
#
# $last_week_number
# [1] "46"
#
# $last_week_label
# [1] "75W46"
```

---

## %%.JalaliDate *Operators*

---

### Description

Perform arithmetic operations

### Usage

```
## S3 method for class 'JalaliDate'
x %% y
```

### Arguments

| | |
|---|---|
| x | JalaliDate or numeric |
| y | JalaliDate or numeric |

### Details

Only +, - operators work with JalaliDate objects in some cases. If each of the two arguments are JalaliDate, the '-' operator calculates the distance of two dates (see diffdate).

### Value

JalaliDate

### Examples

```
JalaliDate("1395/10/11") + 1
# [1] "1395/10/12"
JalaliDate("1403/08/10") - 367
# [1] "1402/08/08"
JalaliDate("1403/09/10") - JalaliDate("1403/08/10")
# [1] 30
```

---

## %/%.JalaliDate *Operators*

---

### Description

Perform arithmetic operations

### Usage

```
## S3 method for class 'JalaliDate'
x %/% y
```

### Arguments

| | |
|---|---|
| x | JalaliDate or numeric |
| y | JalaliDate or numeric |

### Details

Only +, - operators work with JalaliDate objects in some cases. If each of the two arguments are JalaliDate, the '-' operator calculates the distance of two dates (see diffdate).

### Value

JalaliDate

### Examples

```
JalaliDate("1395/10/11") + 1
# [1] "1395/10/12"
JalaliDate("1403/08/10") - 367
# [1] "1402/08/08"
JalaliDate("1403/09/10") - JalaliDate("1403/08/10")
# [1] 30
```

## %*%.JalaliDate *Operators*

### Description

Perform arithmetic operations

### Usage

```
## S3 method for class 'JalaliDate'
x %*% y
```

### Arguments

| | |
|---|---|
| x | JalaliDate or numeric |
| y | JalaliDate or numeric |

### Details

Only +, - operators work with JalaliDate objects in some cases. If each of the two arguments are JalaliDate, the '-' operator calculates the distance of two dates (see diffdate).

### Value

JalaliDate

### Examples

```
JalaliDate("1395/10/11") + 1
# [1] "1395/10/12"
JalaliDate("1403/08/10") - 367
# [1] "1402/08/08"
JalaliDate("1403/09/10") - JalaliDate("1403/08/10")
# [1] 30
```

## ^.JalaliDate *Operators*

### Description

Perform arithmetic operations

### Usage

```
## S3 method for class 'JalaliDate'
x ^ y
```

## Arguments

| | |
|---|---|
| x | JalaliDate or numeric |
| y | JalaliDate or numeric |

## Details

Only +, - operators work with JalaliDate objects in some cases. If each of the two arguments are JalaliDate, the '-' operator calculates the distance of two dates (see diffdate).

## Value

JalaliDate

## Examples

```
JalaliDate("1395/10/11") + 1
# [1] "1395/10/12"
JalaliDate("1403/08/10") - 367
# [1] "1402/08/08"
JalaliDate("1403/09/10") - JalaliDate("1403/08/10")
# [1] 30
```

# Index