

# Julian

INHERITS FROM

Object

WRITTEN BY

Charles G. Bennett

Version 1.2, , This class is in the Public Domain. No guaranties are made to its usefulness or correctness.

## CLASS DESCRIPTION

The Julian Class is an Interface Builder Module to implement julian day functions.

## FEATURES

- Rich set of class methods allows you to treat Julian as a function library.
- Fractional Days supported for easy time tracking and comparison.
- Instance variable and methods allow you to create lists of dates.

## INSTANCE VARIABLES

*Declared in Julian*

double

julianDayVal

## METHOD TYPES

Initialization

- (BOOL) initDay:::
- (BOOL) initDay::::::

Archiving

- read:
- write:

Converting to and from julian dates

- + (double) getCurrentDate
- + (double) julianDay:::
- + (double) julianDay::::::
- + (void) calendarDay:::
- (void) calendarDay::::::
- getCalendarDay:::
- getCalendarDay::::::
- (double) getJulianDay:
- setJulianDay:
- setJulianDay:::

Testing for valid dates	- setJulianDay::: + (BOOL) validDay:: + (BOOL) validDay:::
misc	+ (void) getEasterDay:: + (int) dow:: + (int) doy:: + (double) wkd::
Internal Methods	None.

## CLASS METHODS

### **getCurrentDate**

+ (double) **getCurrentDate**

Returns the julian day for the current month, day, year, hour, min, and second.  
This routine calls the unix localTime function.

### **getCalendarDay::**

- **getCalendarDay**  
   :(int) day  
                                   :(int) month  
                                   :(int) year

Returns the calendar date for the instance variable .

### **getCalendarDay::::**

- **getCalendarDay**  
   :(int) day  
                                   :(int) month  
                                   :(int) year  
                                   :(int) hour  
                                   :(int) min  
                                   :(int) sec

Returns the calendar date and time for the instance variable .

### **getEasterDay**

+ (void) **getEasterDay**:(int ) year  
                                   :(int \*) day  
                                   :(int \*) month

Returns the day and month of Easter. Valid for 1900-2099  
Submitted and written by kjell@oops.se (Kjell\_Nilsson).  
Thanks.

### **julianDay:::**

```

+ (double) julianDay
  :(int) day
                :(int) month
                :(int) year

```

Returns the julian day for the given month day and year;

#### **julianDay::::::**

```

+ (double) julianDay
  :(int) day
                :(int) month
                :(int) year
                :(int) hour
                :(int) min
                :(int) sec

```

Returns the julian day and fractional day for the given month, day, year, hour, min, and second.

#### **calendarDay::::**

```

+ (void) calendarDay
  :(double) julian
                :(int*) day
                :(int*) month
                :(int*) year

```

Returns the month, day, and year for the given julian day.

#### **calendarDay::::::**

```

+ (void) calendarDay
  :(double) julian
                :(int*) day
                :(int*) month
                :(int*) year
                :(int*) hour
                :(int*) min
                :(int*) sec

```

Returns the month, day, year, hour, minute, and second for the given julian day and fractional day.

#### **validDay:::**

```

+ (BOOL) validDay
                :(int) day
                :(int) month
                :(int) year

```

Returns YES if the day month and year are valid, NO otherwise.

#### **validDay::::::**

```

+ (BOOL) validDay
                :(int) day
                :(int) month
                :(int) year
                :(int) hour
                :(int) min
                :(int) sec

```

This routine extends the testing to include hour, min and seconds.

### **dow:**

+ (int) **dow**:(long) julian

This method **returns** the Day Of Week value. The Day Of Week is defined  
as 0 = Sunday, 1=Monday .... 6=Saturday

### **doy:::**

+ (int) **doy**  
:(int) day  
:(int) month  
:(int) year

This method **returns** the Day Of Year value. The Day Of Year is defined  
as 1= Jan 1

### **wkd:::**

+ (double) **wkd**  
:(int) day  
:(int) month  
:(int) year

This method **returns** the number of weekdays since some time in the past.  
Use this method to find the number of "workdays" between dates.

NOTE: a day is defined as 12:00 NOON to 12:00 NOON so there is .5 days  
difference between Friday and Saturday of the same week.

## INSTANCE METHODS

### - **initDay:::**

- (BOOL) **initDay**  
:(int) month  
:(int) day  
:(int) year

This method will initialize the instance variable to the given date.  
It returns **YES** if the date is valid or **NO** if not. If the date is invalid  
the instance variable is **NOT** changed.

### - **initDay::::::**

- (BOOL) **initDay**  
:(int) month  
:(int) day  
:(int) year  
:(int) hour  
:(int) min

:(int) sec

This method will initialize the instance variable to the given date.  
It returns **YES** if the date is valid or **NO** if not. If the date is invalid the instance variable is **NOT** changed.

- **read:**(NXTypedStream \*)*stream*

- **read**

Reads the Julian instance variable from *stream*. A **read:** message is sent during unarchiving. You never invoke this method directly.

- **write:**

- **write:**(NXTypedStream \*)*stream*

Writes the Julian instance variable to *stream*. A **write:** message is sent during archiving. You never invoke this method directly.

- **getJulianDay**

- (double) **getJulianDay**

This method **returns** the value of the julian day instance variable.

- **setJulianDay:**

- (BOOL) **setJulianDay:**(double) day

This method **sets** the value of the julian day instance variable.  
**Caution!** Use this with care since this directly sets the instance variable. Always returns **YES**

- **setJulianDay:::**

- (BOOL) **setJulianDay**  
:(int) month  
:(int) day  
:(int) year

This method **sets** the value of the julian day instance variable.  
Using the month, day and year parameters. Returns **YES** if the date was valid, **NO** if not., and the instance variable is NOT changed

- **setJulianDay:::::**

- (BOOL) **setJulianDay**  
:(int) month  
:(int) day  
:(int) year  
:(int) hour

:(int) min  
:(int) sec

This method **sets** the value of the julian day instance variable.

Using the month, day, year, hour, min, and sec parameters.

Returns **YES** if the date was valid, **NO** if not., and the instance variable is  
NOT  
changed

## CONSTANTS AND DEFINED TYPES

None.