

[Home](#) | [Categories](#) | [Alphabetical](#) | [Classes](#) | [All Contents](#) | [\[< \]](#) | [\[> \]](#)

JULDAY

[Syntax](#) | [Return Value](#) | [Arguments](#) | [Examples](#) | [Version History](#) | [See Also](#)

The JULDAY function calculates the Julian Day Number (which begins at noon) for the specified date. This is the inverse of the CALDAT procedure.

Note

The Julian calendar, established by Julius Caesar in the year 45 BCE, was corrected by Pope Gregory XIII in 1582, excising ten days from the calendar. The CALDAT procedure reflects the adjustment for dates after October 4, 1582. See the example below for an illustration.

Note

A small offset is added to the returned Julian date to eliminate roundoff errors when calculating the day fraction from hours, minutes, seconds. This offset is given by the larger of EPS and EPS*Julian, where Julian is the integer portion of the Julian date, and EPS is the EPS field from MACHAR. For typical Julian dates, this offset is approximately 6×10^{-1} (which corresponds to 5×10^{-5} seconds). This offset ensures that if the Julian date is converted back to hour, minute, and second, then the hour, minute, and second will have the same integer values as were originally input.

Note

Calendar dates must be in the range 1 Jan 4716 B.C.E. to 31 Dec 5000000, which corresponds to Julian values -1095 and 1827933925, respectively.

This routine is written in the IDL language. Its source code can be found in the file `julday.pro` in the `lib` subdirectory of the IDL distribution.

Syntax

Result = JULDAY(*Month*, *Day*, *Year*, *Hour*, *Minute*, *Second*)

Return Value

Result is of type double-precision if Hour, Minute, or Second is specified, otherwise *Result* is of type long integer. If all arguments are scalar, the function returns a scalar. If all arguments are arrays, the function matches up the corresponding elements of the arrays, returning an array with the same dimensions as the smallest array. If the inputs contain both scalars and arrays, the function uses the scalar value with each element of the arrays, and returns an array with the same dimensions as the smallest input array.

Arguments

Month

Number of the desired month (1 = January, ..., 12 = December). *Month* can be either a scalar or an array.

Day

Number of the day of the month (1-31). *Day* can be either a scalar or an array.

Year

Number of the desired year (e.g., 1994). *Year* can be either a scalar or an array.

Hour

Number of the hour of the day (0-23). *Hour* can be either a scalar or an array.

Minute

Number of the minute of the hour (0-59). *Minute* can be either a scalar or an array.

Second

Number of the second of the minute (0-59). *Second* can be either a scalar or an array.

Examples

In 1582, Pope Gregory XIII adjusted the Julian calendar to correct for its inaccuracy of slightly more than 11 minutes per year. As a result, the day following October 4, 1582 was October 15, 1582. JULDAY follows this convention, as illustrated by the following commands:

```
PRINT, JULDAY(10,4,1582), JULDAY(10,5,1582), JULDAY(10,15,1582)
```

IDL prints:

```
2299160      2299161      2299161
```

Using arrays, this can also be calculated as follows:

```
PRINT, JULDAY(10, [4, 5, 15], 1582)
```

If you are using JULDAY to calculate an absolute number of days elapsed, be sure to account for the Gregorian adjustment.

Version History

Introduced: Original

See Also

[BIN_DATE](#), [CALDAT](#), [SYSTIME](#)

[Home](#) | [Categories](#) | [Alphabetical](#) | [Classes](#) | [All Contents](#) | [\[< \]](#) | [\[> \]](#)