Amtiri

AMTIRI SCRIPT

Dates

Generated on November 25, 2026

Dates

Add days

`addDays` · `addDays(date: DateTime, days: double)`

Shifts a date by a number of days. Fractions count: addDays(d, 1.5) adds 36 hours.

Parameters

Name	Type	Description
`date`	`DateTime`	The date to shift.
`days`	`double`	How many days to add, fractions included. Negative moves back.

Returns `DateTime`: The shifted date.

Examples

```
addDays(START, 1) # 2026-09-30T08:00
START.addDays(-7) # 2026-09-22T08:00
addDays(START, 1.5) # 2026-09-30T20:00
```

Add hours

`addHours` · `addHours(date: DateTime, hours: double)`

Shifts a date and time by a number of hours. Fractions count: addHours(d, 1.5) adds 90 minutes.

Parameters

Name	Type	Description
`date`	`DateTime`	The date and time to shift.
`hours`	`double`	How many hours to add, fractions included. Negative moves back.

Returns `DateTime`: The shifted date and time.

Examples

```
addHours(START, 2) # 2026-09-29T10:00
addHours(START, -8) # 2026-09-29T00:00
```

Add minutes

`addMinutes` · `addMinutes(date: DateTime, minutes: double)`

Shifts a date and time by a number of minutes. Fractions count: addMinutes(d, 1.5) adds 90 seconds.

Parameters

Name	Type	Description
`date`	`DateTime`	The date and time to shift.
`minutes`	`double`	How many minutes to add, fractions included. Negative moves back.

Returns `DateTime` : The shifted date and time.

Examples

```
addMinutes(START, 90) # 2026-09-29T09:30
```

Add months

`addMonths` · `addMonths(date: DateTime, months: int)`

Shifts a date by a whole number of months.

Parameters

Name	Type	Description
`date`	`DateTime`	The date to shift.
`months`	`int`	How many months to add, as a whole number: a fraction is dropped. Negative moves back.

Returns `DateTime` : The shifted date.

Examples

```
addMonths(START, 1) # 2026-10-29T08:00
addMonths(date(2026, 1, 31), 1) # 2026-02-28T00:00
```

Add weeks

`addWeeks` · `addWeeks(date: DateTime, weeks: double)`

Shifts a date by a number of weeks. Fractions count: addWeeks(d, 0.5) adds three and a half days.

Parameters

Name	Type	Description
`date`	`DateTime`	The date to shift.
`weeks`	`double`	How many weeks to add, fractions included. Negative moves back.

Returns `DateTime`: The shifted date.

Examples

```
addWeeks(START, 2) # 2026-10-13T08:00
```

Add years

`addYears` · `addYears(date: DateTime, years: int)`

Shifts a date by a whole number of years.

Parameters

Name	Type	Description
`date`	`DateTime`	The date to shift.
`years`	`int`	How many years to add, as a whole number: a fraction is dropped. Negative moves back.

Returns `DateTime`: The shifted date.

Examples

```
addYears(START, 1) # 2027-09-29T08:00
```

Date

`date` · `date(year: int, month: int, day: int, [hour: int], [minute: int], [second: int])`

Builds a date and time from its parts. Give three parts for a plain date, five to add a time of day, or six to add seconds - hour and minute must be supplied together.

Parameters

Name	Type	Description
`year`	`int`	The four-digit year.
`month`	`int`	The month, 1 to 12.
`day`	`int`	The day of the month, starting at 1.
`hour`	`int` (optional)	The hour, 0 to 23. Defaults to 0, and must be given together with minute.
`minute`	`int` (optional)	The minute, 0 to 59. Defaults to 0, and must be given together with hour.
`second`	`int` (optional)	The second, 0 to 59. Defaults to 0.

Returns `DateTime`: The assembled date and time.

Examples

```
date(2026, 9, 29) # 2026-09-29T00:00
date(2026, 9, 29, 8, 30) # 2026-09-29T08:30
date(2026, 9, 29, 8, 30, 15) # 2026-09-29T08:30:15
```

Date part

`date_of` · `date_of(date: DateTime)`

Drops the time of day, giving the same date at midnight.

Parameters

Name	Type	Description
`date`	`DateTime`	The date and time to truncate.

Returns `DateTime`: The date, at the start of its day.

Examples

```
date_of(addHours(START, 3)) # 2026-09-29T00:00
```

Day of month

`dayOfMonth` · `dayOfMonth(date: DateTime)`

The day part of a date, starting at 1.

Parameters

Name	Type	Description
`date`	`DateTime`	The date to read.

Returns `int`: The day of the month.

Examples

```
dayOfMonth(START) # 29
```

Day of week

`dayOfWeek` · `dayOfWeek(date: DateTime)`

Which day of the week a date falls on, Monday being 1.

Parameters

Name	Type	Description
`date`	`DateTime`	The date to read.

Returns `int`: The day of the week, 1 to 7.

Examples

```
dayOfWeek(START) # 2
```

From day number

`toDate` · `toDate(days: double)`

Turns a day number counted from 1899-12-31 back into a date and time - the inverse of toDays. A spreadsheet's serial number for the same date is one more.

Parameters

Name	Type	Description
`days`	`double`	The day count since 1899-12-31, with the time of day as its fraction.

Returns `DateTime`: The date and time.

Examples

```
toDate(46294.5) # 2026-09-30T12:00
toDate(toDays(START)) # 2026-09-29T08:00
```

Month

`month` · `month(date: DateTime)`

The month part of a date, 1 to 12.

Parameters

Name	Type	Description
`date`	`DateTime`	The date to read.

Returns `int`: The month number.

Examples

```
month(START) # 9
```

Now

`now` · `now()`

The current date and time in UTC. Convert with toLocale only where a value is about to be displayed.

Returns `DateTime`: The current UTC date and time.

Examples

```
now() # *
type_of(now()) # "LocalDateTime"
```

To day number

`toDays` · `toDays(date: DateTime)`

A date as a decimal count of days since 1899-12-31, the time of day as its fraction, so a duration becomes a subtraction. It is one less than a spreadsheet's serial number for the same date.

Parameters

Name	Type	Description
`date`	`DateTime`	The date and time to convert.

Returns `double`: The day number, with the time of day as its fraction.

Examples

```
toDays(START) # 46293.333333333336
toDays(addDays(START, 1)) - toDays(START) # 1.0
```

To local time

`toLocale` · `toLocale(date: DateTime)`

Reads a UTC date and time as the configured time zone. Use it only at the point of display; keep stored values in UTC.

Parameters

Name	Type	Description
`date`	`DateTime`	The UTC date and time to convert.

Returns `DateTime`: The same instant, expressed in local time.

Examples

```
toLocale(START) # 2026-09-29T08:00
```

To UTC

`toUTC` · `toUTC(date: DateTime)`

Reads a date and time expressed in the configured time zone back as UTC - the inverse of toLocale, and what to apply to any date a person entered.

Parameters

Name	Type	Description
`date`	`DateTime`	The local date and time to convert.

Returns `DateTime`: The same instant, expressed in UTC.

Examples

```
toUTC(START) # 2026-09-29T08:00
```

Year

`year` · `year(date: DateTime)`

The year part of a date.

Parameters

Name	Type	Description
`date`	`DateTime`	The date to read.

Returns `int`: The four-digit year.

Examples

```
year(START) # 2026
START.year() # 2026
```

This page describes Amtiri Script 5.5.0 as shipped with Data Orchester engine 6.22.0 and Site 1.27.0.