

parseInterval

Description

Returns an interval parsed from a string representation such as `2d` or `35m10s`, enabling calculations with durations.

A valid string must follow these rules:

- Format: `NdNhNmNsNmsNusNns` where `N` is a non-negative integer
- At least one time unit must be present
- No unit may appear more than once (`1d2d` is invalid)
- Units must appear in descending order (days → nanoseconds). `1d1s` is valid; `1s1d` is not
- A leading `-` is allowed to indicate a negative interval (the only valid position for `-`)
- If the format is invalid, the function returns `null`

Syntax

Like many functions in DataPrime, `parseInterval` supports [two notations](#), **function** and **method** notation. These interchangeable forms allow flexibility in how you structure expressions.

Function notation

```
parseInterval(string: string): interval
```

Method notation

```
(string: string).parseInterval(): interval
```

Arguments

Name	Type	Required	Description
string	string	true	The string to parse into an interval

Example

Use case: Parse an interval string and add it to a timestamp

A log includes a timestamp and a `completed_in` field with a duration string. Parse the string to an interval, then add it to the timestamp to calculate the job's completion time.

```
{
  "timestamp": 1728763337,
  "job_name": "BUILD_VIDEO",
  "completed_in": "35m10s"
}
```

Example query

Function notation

```
create completed_time from addTime(timestamp, parseInterval(completed_in))
```

Method notation

```
create completed_time from timestamp.addTime(completed_in.parseInterval())
```

Example output

```
{
  "timestamp": 1728763337,
  "job_name": "BUILD_VIDEO",
  "completed_in": "35m10s",
  "completed_time": 1728765447
}
```

🕒 Last updated: October 19, 2025

Was this helpful?

Leave your feedback here.

☐ Yes ☐ No

Send

© 2025 Coralogix. All rights reserved.

Generated on: November 17, 2025

Source: <https://coralogix.com/docs/dataprime/language-reference/functions-reference/time/parseinterval/>