Parsing

Read and validate dates, strictly or like PHP does, and validate the periods written as numbers

Anonymous · 05/11/2026

Parsing

```
use Derafu\Date\Support\DateParser;
```

Method	What it does
`DateParser::parse($text)`	Reads a date like PHP does.
`DateParser::parseStrict($text, $format = 'Y-m-d')`	Reads a date written exactly in a format.
`DateParser::ensure($dateOrText)`	A `DateTimeImmutable` from a date or a text.
`DateParser::isValid($text, $format = 'Y-m-d')`	Whether a text is a date written exactly in the format.
`DateParser::reformat($text, $format = 'd/m/Y')`	Writes a `Y-m-d` date in another format.
`DateParser::fromSpreadsheetSerial($number)`	The date of a spreadsheet serial number.

Lenient and strict

`parse()` is as lenient as `DateTimeImmutable`: it accepts relative texts (`tomorrow`), and a day that the month does not have is moved to the next month.

```
DateParser::parse('2025-02-31');   // 2025-03-03
```

`parseStrict()` and `isValid()` read a date in a format and refuse what is not a date: it must exist and have every field of the format. They do not accept relative texts, a day that the month does not have or extra text.

```
DateParser::parseStrict('15/01/2024', 'd/m/Y');   // 2024-01-15 00:00:00
DateParser::isValid('2025-02-31');                // false
DateParser::isValid('tomorrow');                  // false
```

They read the fields as PHP does, so the digits can be fewer than the format writes: `2025-1-5` is the date `2025-01-05`, and a year written `24` is the year 24. They are for reading; `->format('Y-m-d')` gives

the date in its canonical form, and `reformat()` refuses what is not written with all its digits.

`parseStrict()` gives the start of the day. `parse()` and `parseStrict()` throw an `InvalidDateException` when the text is not a date.

Reformat

`reformat()` is for a date that must be exactly `Y-m-d`. It gives `null` if it is not one, and that includes a date written without the zeros (`2024-1-5`).

```
DateParser::reformat('2023-03-12');            // 12/03/2023
DateParser::reformat('2023-03-12', 'Y.m.d');   // 2023.03.12
DateParser::reformat('12/03/2023');            // null
```

Spreadsheet serial numbers

Spreadsheets number the days since 1900: `1` is January 1, 1900. They also count a day that never existed, February 29, 1900, which is the number `60` and gives February 28.

```
DateParser::fromSpreadsheetSerial(45292);   // 2024-01-01
```

Periods written as numbers

A period written as a number is a year (`2025`) or a year and a month (`202511`).

Method	What it checks
`DateParser::isValidPeriod($period, $yearFrom = 2000, $yearTo = 2100, $length = null)`	A year or a month, within the years. With `$length` (4 or 6) it must have those digits.
`DateParser::isValidYearPeriod($period, ...)`	A year.
`DateParser::isValidMonthPeriod($period, ...)`	A year and a month.

To work with the months, use `Month::fromInt()`.

Last updated on 05/11/2026