



1. ■■■■■ ■■■■■ ■■■ ■■■■■
2. ■■■■ ■■■■■■■ ■■■■■■■■ ■■■■■
3. ■■■■■ ■■■■ ■■■■■■
4. ■■■■■ ■■■■■■ ■■■■■■■■
5. ■■■■■ ■■■■■■■ ■■■■■■■
6. ■■■■■ ■■■■■■
7. ■■■■■ ■■■■■■
8. ■■■■ ■■■■■■■■ ■■■■■

1. Polars string methods

Polars has a rich set of string methods that can be used to filter, transform, and manipulate string data. This section covers the most common string methods in Polars.

Method	Description	Example
<code>starts_with()</code>	Filter rows where the string starts with a specific character or substring.	<code>df.filter(pl.col('name').str.starts_with('A'))</code>
<code>ends_with()</code>	Filter rows where the string ends with a specific character or substring.	<code>df.filter(pl.col('email').str.ends_with('.com'))</code>
<code>contains()</code>	Filter rows where the string contains a specific substring.	<code>df.filter(pl.col('text').str.contains('word'))</code>
<code>contains_any()</code>	Filter rows where the string contains any of the specified substrings.	<code>df.filter(pl.col('text').str.contains_any(['a', 'b']))</code>
<code>len_chars()</code>	Filter rows based on the length of the string in characters.	<code>df.filter(pl.col('text').str.len_chars() > 10)</code>

Example:

```
df.filter(pl.col('content').str.starts_with('Very'))
.select('content')
.head()
```

2. ■■■■ ■■■■■■■■ ■■■■■■■■ ■■■■■■

Polars **nie** jest przeznaczony do przetwarzania danych w czasie rzeczywistym.

<div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div>	<div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div>	<div> <div></div> <div></div> <div></div> <div></div> <div></div> </div>
to_date()	Date	pl.col('date_str').str.to_date(format='%Y-%m-%d')
to_time()	Time	pl.col('time_str').str.to_time(format='%H:%M:%S')
to_datetime()	Datetime	pl.col('dt').str.to_datetime(format='%Y-%m-%d %H:%M')
strptime()	<div> <div></div> <div></div> <div></div> <div></div> </div>	pl.col('str').str.strptime(pl.Date, '%Y-%m-%d')

[REDACTED] [REDACTED] [REDACTED]:

■ ■ ■ ■	■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■
%Y	■■■■ (4 ■■■■■■)	2024
%m	■■■■■ (2 ■■■■■■)	01-12
%d	■■■■ (2 ■■■■■■)	01-31
%H	■■■■ (24 ■■■■)	00-23
%M	■■■■■	00-59
%S	■■■■■■	00-59

3. ■■■■■■ ■■■■■■ ■■■■■■ ■■■■■■

■■■■■ ■■■■■■ ■■■■■■■■■■ ■■■■ ■■■■ ■■■■■■ ■■■■■■■■■■ ■■■■■■. Polars ■■■■■■ ■■■■■■ ■■■■■■■■■■ - ■■■■■■■■ ■■■■■■ ■■■■ ■■■■■■■■■■ ■■■■■■■■■■ ■■■■■■■■■■.

■■■■■■■■■	■■■■■■■	■■■■■■■
slice()	■■■■■■■ ■■■■ ■■■■■■■■	.str.slice(3, 5)
extract()	■■■■■■■ ■■ Regex (■■■■■■■)	.str.extract(r'(\d+)')
extract_all()	■■■■■■■ ■■ ■■■■■■■■	.str.extract_all(r'(\w+)')
extract_groups()	■■■■■■■ ■■■■■■■■ ■■■■■■■■	.str.extract_groups(r'(\d+)-(\d+)')

■■■■■■■ Regex ■■■■■■■■■■:

■■■■■	■■■■■■■
\d+	■■■■■ ■■■■ ■■ ■■■■
[A-Za-z]+	■■■■■■■
\w+	■■■■■ (■■■■■■■, ■■■■■■, _)
\s+	■■■■■ ■■■■
^start	■■■■■■■ ■■■■■■■■
end\$	■■■■■ ■■■■■■■■

4. ■■■■■ ■■■■■ ■■■■■■■■■■

■■■■■ ■■■■■■■ ■■■ ■■■ ■■■■■ ■■■■■ ■■■■■■■. Polars ■■■■ ■■■■ ■■■ ■■■■■ ■■■■■■■, ■■■■■ ■■■■■■■ ■■■■■■■.

■■■■■	■■■■■	■■■■■
■■■■■■	strip_chars()	■■■■■ ■■■■■■■ ■■■■■■■/■■■■
■■■■■■	replace_all()	■■■■■■ ■■■■
■■■■■■	to_lowercase()	■■■■■ ■■■■■■■ ■■■■■
■■■■■■	to_uppercase()	■■■■■ ■■■■■■■ ■■■■■
■■■■■■	to_titlecase()	Title Case
■■■■■■	pad_start()	■■■■■■ ■■■■■ ■■■■■
■■■■■■	pad_end()	■■■■■■ ■■■■■ ■■■■■

7. ■■■■■■ ■■■■■■ ■■■■

■ ■■■■■■: ■■■■■ ■-literal=True ■■■■ ■■ ■■■■ Regex - ■■ ■■■■ ■■■■!

■ NULL values: ■■■■■ ■-fill_null() ■■■■ ■■■■ ■■■■■■ ■■■■■■ ■■■■■■

■ Case Sensitivity: ■■■ ■■■■■■■■■■ ■■■■■■ ■■■■■■■■. ■■■■■ ■-(?i) ■-Regex
■■■■■■ ■■ ■■■■

■ ■■■■■ ■■■■■■■■: ■■■■ ■■■■■ ■■■■ ■■■■■ ■■■■■■■■:
.str.strip().str.lower().str.replace()

■ ■■■■■: ■■■■■ ■-.str.len_chars().describe() ■■■■■■ ■■■■■■ ■■■■■■■■

■■■■■ ■■■■■ ■■■■■■■■:

■ ■■■■ ■■■ ■■■■■■■■ ■■■■ ■■■■■■ ■■■■

■ ■■■■■ ■■■■■■■■■■ ■■■■■■■■ (starts_with) ■■■■■■ Regex ■■■■■■

■ ■■■■ ■■■■■■■■ ■■ ■■■■■■■■ ■■■■ ■■■■■■■■ ■■■■

■ ■■■■ NULL values ■■■■ ■■■■■■ ■■■■■■

■ ■■ ■■■■ ■■ ■■■■■■ literal=True ■■■■■■ ■■■■■■ ■■■■■■

■ ■■ ■■■■■ ■-Regex ■■■■■■ ■■■■ ■■■■ - ■■ ■■■■



Polars is a fast, easy-to-use data manipulation library for Python and Rust. It is designed to be a drop-in replacement for pandas, but with a focus on performance and scalability. Polars is built on top of Apache Arrow, which allows it to leverage the power of SIMD instructions and columnar storage for fast data processing. It also supports a wide range of data sources, including CSV, JSON, and databases, making it a versatile tool for data analysis.

Getting started:

- ✓ [Polars documentation](#)
- ✓ [Polars installation](#)
- ✓ [Polars quickstart](#)
- ✓ [Polars Regex](#)
- ✓ [Polars performance](#)
- ✓ [Polars benchmarks](#)
- ✓ [Polars examples](#)

Resources:

- Polars website: <https://pola-rs.github.io/polars/>
- Regex Tester: <https://regex101.com/>
- Python strptime: <https://strptime.org/>
- Polars Discord Community
- Polars PDF