

[Documentation](#)

search

Search

- [rocket launch](#)

[Get started](#)

- [Installation](#)
add
- [Fundamentals](#)
add
- [First steps](#)
add
- [code](#)

[Develop](#)

- [Concepts](#)
add
 - [API reference](#)
remove
 - PAGE ELEMENTS
-
- [Write and magic](#)
remove
 - [st.write](#)
 - [st.write stream](#)
 - [magic](#)
 - [Text elements](#)
add
 - [Data elements](#)
add
 - [Chart elements](#)
add
 - [Input widgets](#)
add
 - [Media elements](#)
add
 - [Layouts and containers](#)
add
 - [Chat elements](#)
add
 - [Status elements](#)
add
 - [Third-party components](#)*open in new*
 - APPLICATION LOGIC
-
- [Navigation and pages](#)
add
 - [Execution flow](#)
add
 - [Caching and state](#)
add

- [Connections and secrets](#)
add
 - [Custom components](#)
add
 - [Utilities](#)
add
 - [Configuration](#)
add
 - TOOLS
-
- [App testing](#)
add
 - [Command line](#)
add
 - [Tutorials](#)
add
 - [Quick reference](#)
add
 - [web_asset](#)

[Deploy](#)

- [Concepts](#)
add
- [Streamlit Community Cloud](#)
add
- [Snowflake](#)
- [Other platforms](#)
add
- [school](#)

[Knowledge base](#)

- [FAQ](#)
- [Installing dependencies](#)
- [Deployment issues](#)
- [Home/](#)
- [Develop/](#)
- [API reference/](#)
- [Write and magic/](#)
- [magic](#)

Magic



Magic commands are a feature in Streamlit that allows you to write almost anything (markdown, data, charts) without having to type an explicit command at all. Just put the thing you want to show on its own line of code, and it will appear in your app. Here's an example:

```
# Draw a title and some text to the app: ''' # This is the document title This is some _markdown_. '''
import pandas as pd df = pd.DataFrame({'coll': [1,2,3]}) df # 🖱 Draw the dataframe x = 10 'x', x # 🖱
Draw the string 'x' and then the value of x # Also works with most supported chart types import
matplotlib.pyplot as plt import numpy as np arr = np.random.normal(1, 1, size=100) fig, ax =
plt.subplots() ax.hist(arr, bins=20) fig # 🖱 Draw a Matplotlib chart
```

How Magic works



Any time Streamlit sees either a variable or literal value on its own line, it automatically writes that to your app using [st.write](#) (which you'll learn about later).

Also, magic is smart enough to ignore docstrings. That is, it ignores the strings at the top of files and functions.

If you prefer to call Streamlit commands more explicitly, you can always turn magic off in your `~/.streamlit/config.toml` with the following setting:

```
[runner] magicEnabled = false
priority_high
```

Important

Right now, Magic only works in the main Python app file, not in imported files. See GitHub issue #288 for a discussion of the issues.

Featured video



Learn what the [st.write](#) and [magic](#) commands are and how to use them.



[←Previous: st.write streamNext: Text elements→](#)
forum

Still have questions?

Our [forums](#) are full of helpful information and Streamlit experts.

[Home](#)[Contact Us](#)[Community](#)



© 2025 Snowflake Inc. [Cookie policy](#)

[forum](#) [Ask AI](#)