

[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](#)
add
 - [Text elements](#)
add
 - [Data elements](#)
add
 - [Chart elements](#)
add
 - [Input widgets](#)
add
 - [Media elements](#)
add
 - [Layouts and containers](#)
remove
 - [st.columns](#)
 - [st.container](#)
 - [st.dialoglink](#)
 - [st.empty](#)
 - [st.expander](#)
 - [st.formlink](#)
 - [st.popover](#)
 - [st.sidebar](#)
 - [st.tabs](#)
 - [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/](#)
- [Layouts and containers/](#)
- [st.columns](#)

st.columns



Streamlit Version ▼

Insert containers laid out as side-by-side columns.

Inserts a number of multi-element containers laid out side-by-side and returns a list of container objects.

To add elements to the returned containers, you can use the `with` notation (preferred) or just call methods directly on the returned object. See examples below.

Columns can only be placed inside other columns up to one level of nesting.

Warning

Columns cannot be placed inside other columns in the sidebar. This is only possible in the main area of the app.

Function signature[\[source\]](#)

`st.columns(spec, *, gap="small", vertical_alignment="top", border=False)`

Parameters

Controls the number and width of columns to insert. Can be one of:

- spec (int or Iterable of numbers)
- An integer that specifies the number of columns. All columns have equal width in this case.
 - An Iterable of numbers (int or float) that specify the relative width of each column. E.g. `[0.7, 0.3]` creates two columns where the first one takes up 70% of the available width and the second one takes up 30%. Or `[1, 2, 3]` creates three columns where the second one is two times the width of the first one, and the third one is three times that width.

gap ("small", "medium", or "large")

The size of the gap between the columns. The default is "small".

vertical_alignment ("top", "center", or "bottom")

The vertical alignment of the content inside the columns. The default is "top".

border (bool)

Whether to show a border around the column containers. If this is `False` (default), no border is shown. If this is `True`, a border is shown around each column.

Returns

(list of containers)

A list of container objects.

Examples

Example 1: Use context management

You can use the `with` statement to insert any element into a column:

```
import streamlit as st

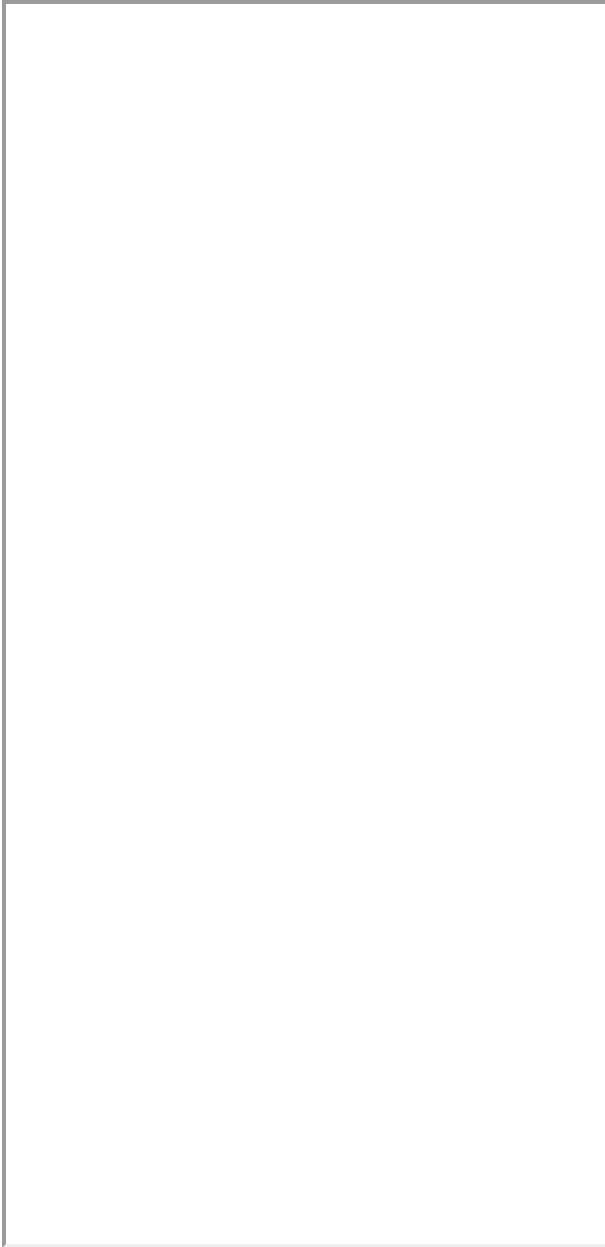
col1, col2, col3 = st.columns(3)

with col1:
    st.header("A cat")
    st.image("https://static.streamlit.io/examples/cat.jpg")

with col2:
    st.header("A dog")
    st.image("https://static.streamlit.io/examples/dog.jpg")

with col3:
```

```
st.header("An owl")
st.image("https://static.streamlit.io/examples/owl.jpg")
```



[Built with Streamlit](#) 📍
[Fullscreen open in new](#)

Example 2: Use commands as container methods

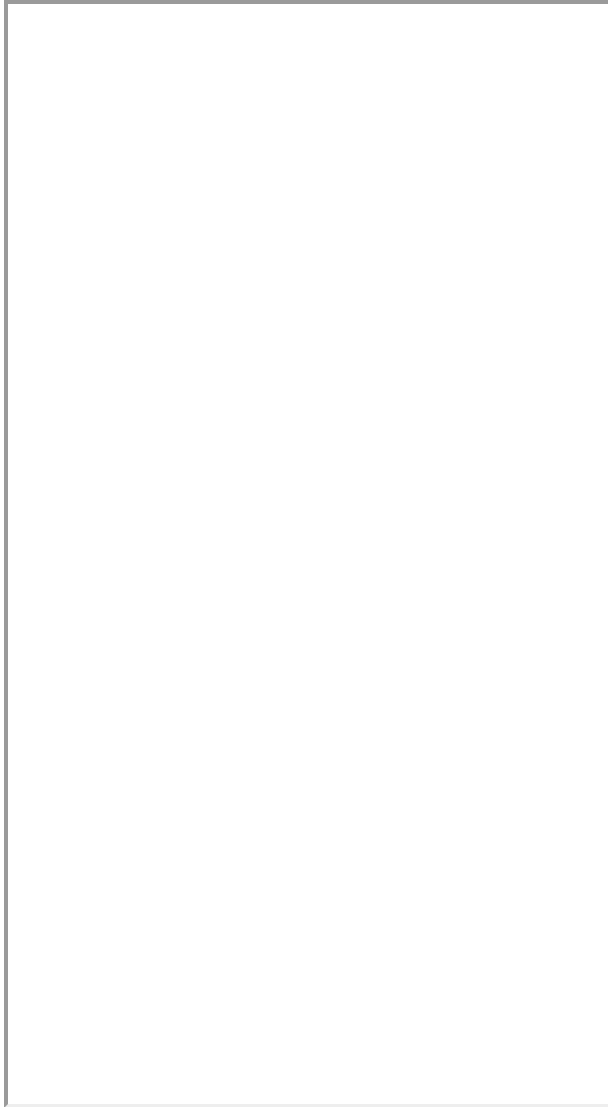
You can just call methods directly on the returned objects:

```
import streamlit as st
import numpy as np

col1, col2 = st.columns([3, 1])
data = np.random.randn(10, 1)

col1.subheader("A wide column with a chart")
col1.line_chart(data)

col2.subheader("A narrow column with the data")
col2.write(data)
```



[Built with Streamlit !\[\]\(8af806fb1314382d09bc5ec5b767526c_img.jpg\)](#)
[Fullscreen open in new](#)

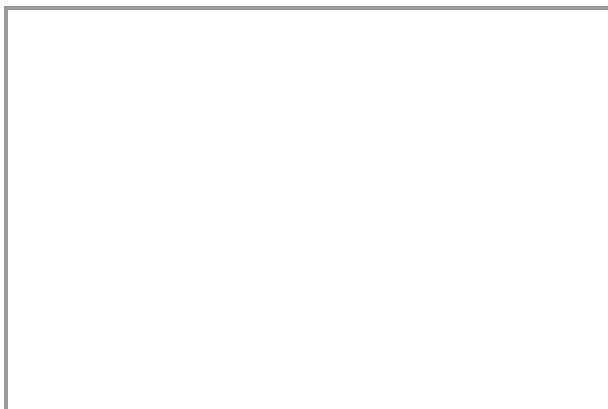
Example 3: Align widgets

Use `vertical_alignment="bottom"` to align widgets.

```
import streamlit as st

left, middle, right = st.columns(3, vertical_alignment="bottom")

left.text_input("Write something")
middle.button("Click me", use_container_width=True)
right.checkbox("Check me")
```



[Built with Streamlit !\[\]\(0aff635c4179ba9e710b00f4b01d3b20_img.jpg\)](#)
[Fullscreen open in new](#)

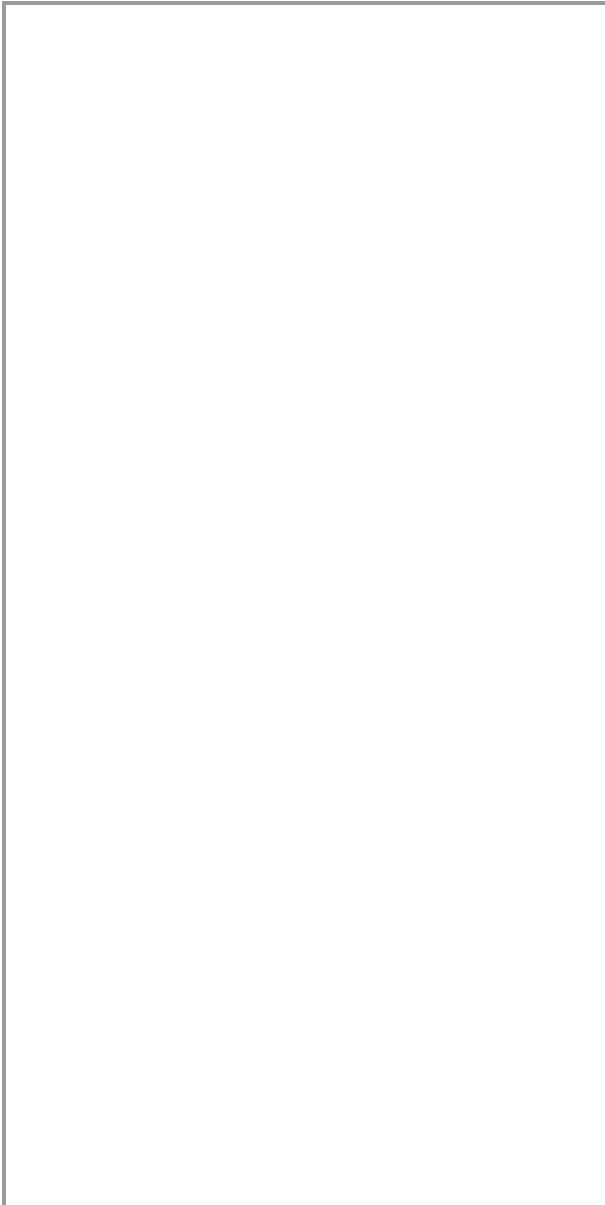
Example 4: Use vertical alignment to create grids


Adjust vertical alignment to customize your grid layouts.

```
import streamlit as st
import numpy as np

vertical_alignment = st.selectbox(
    "Vertical alignment", ["top", "center", "bottom"], index=2
)

left, middle, right = st.columns(3, vertical_alignment=vertical_alignment)
left.image("https://static.streamlit.io/examples/cat.jpg")
middle.image("https://static.streamlit.io/examples/dog.jpg")
right.image("https://static.streamlit.io/examples/owl.jpg")
```



[Built with Streamlit](#) 
[Fullscreen open in new](#)

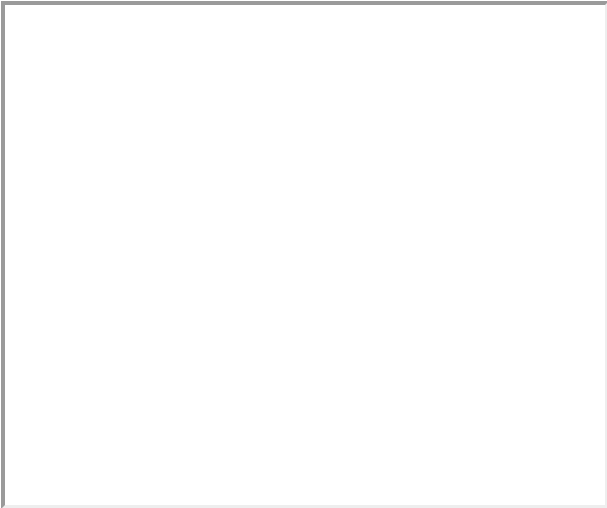
Example 5: Add borders


Add borders to your columns instead of nested containers for consistent heights.

```
import streamlit as st

left, middle, right = st.columns(3, border=True)
```

```
left.markdown("Lorem ipsum " * 10)
middle.markdown("Lorem ipsum " * 5)
right.markdown("Lorem ipsum ")
```



[Built with Streamlit](#) 
[Fullscreen open in new](#)

[←Previous: Layouts and containers](#)[Next: st.container→](#)
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](#)