

[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](#)
add
 - [Chat elements](#)
add
 - [Status elements](#)
add
 - [Third-party components](#)*open in new*
 - APPLICATION LOGIC

 - [Navigation and pages](#)
remove
 - [st.navigation](#)
 - [st.Page](#)
 - [st.page_link](#)*link*
 - [st.switch_page](#)
 - [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/](#)
- [Navigation and pages/](#)
- [st.navigation](#)

st.navigation



Streamlit Version

Configure the available pages in a multipage app.

Call `st.navigation` in your entrypoint file with one or more pages defined by `st.Page`. `st.navigation` returns the current page, which can be executed using `.run()` method.

When using `st.navigation`, your entrypoint file (the file passed to `streamlit run`) acts like a router or frame of common elements around each of your pages. Streamlit executes the entrypoint file with every app rerun. To execute the current page, you must call the `.run()` method on the `StreamlitPage` object returned by `st.navigation`.

The set of available pages can be updated with each rerun for dynamic navigation. By default, `st.navigation` draws the available pages in the side navigation if there is more than one page. This behavior can be changed using the `position` keyword argument.

As soon as any session of your app executes the `st.navigation` command, your app will ignore the `pages/` directory (across all sessions).

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

`st.navigation(pages, *, position="sidebar", expanded=False)`

Parameters

	The available pages for the app.
<code>pages</code> (list[StreamlitPage] or dict[str, list[StreamlitPage]])	<p>To create labeled sections or page groupings within the navigation menu, <code>pages</code> must be a dictionary. Each key is the label of a section and each value is the list of <code>StreamlitPage</code> objects for that section.</p> <p>To create a navigation menu with no sections or page groupings, <code>pages</code> must be a list of <code>StreamlitPage</code> objects.</p> <p>Use <code>st.Page</code> to create <code>StreamlitPage</code> objects.</p>
<code>position</code> ("sidebar" or "hidden")	<p>The position of the navigation menu. If <code>position</code> is "sidebar" (default), the navigation widget appears at the top of the sidebar. If <code>position</code> is "hidden", the navigation widget is not displayed.</p> <p>If there is only one page in <code>pages</code>, the navigation will be hidden for any value of <code>position</code>.</p>
<code>expanded</code> (bool)	<p>Whether the navigation menu should be expanded. If this is <code>False</code> (default), the navigation menu will be collapsed and will include a button to view more options when there are too many pages to display. If this is <code>True</code>, the navigation menu will always be expanded; no button to collapse the menu will be displayed.</p> <p>If <code>st.navigation</code> changes from <code>expanded=True</code> to <code>expanded=False</code> on a rerun, the menu will stay expanded and a collapse button will be displayed.</p>

Returns

(StreamlitPage)	The current page selected by the user.
-----------------	--

Examples

The following examples show possible entrypoint files, which is the file you pass to `streamlit run`. Your entrypoint file manages your app's navigation and serves as a router between pages.

Example 1: Use a callable or Python file as a page

You can declare pages from callables or file paths.

`page_1.py` (in the same directory as your entrypoint file):

```
import streamlit as st

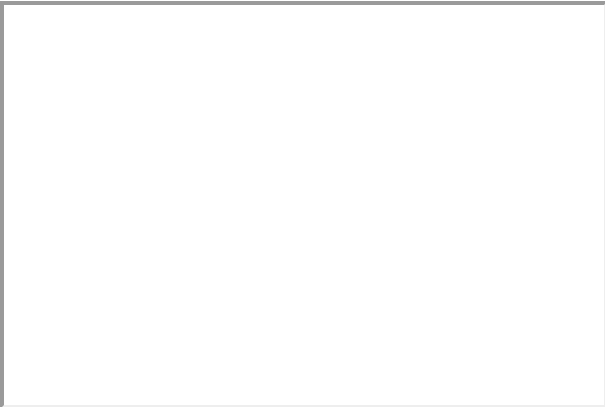
st.title("Page 1")
```

Your entrypoint file:

```
import streamlit as st

def page_2():
    st.title("Page 2")

pg = st.navigation([st.Page("page_1.py"), st.Page(page_2)])
pg.run()
```



[Built with Streamlit !\[\]\(dfbd6b3763a6d1d9afaa974f64e2e4b5_img.jpg\)](#)
[Fullscreen open in new](#)

Example 2: Group pages into sections

You can use a dictionary to create sections within your navigation menu. In the following example, each page is similar to Page 1 in Example 1, and all pages are in the same directory. However, you can use Python files from anywhere in your repository. For more information, see [st.Page](#).

Directory structure:

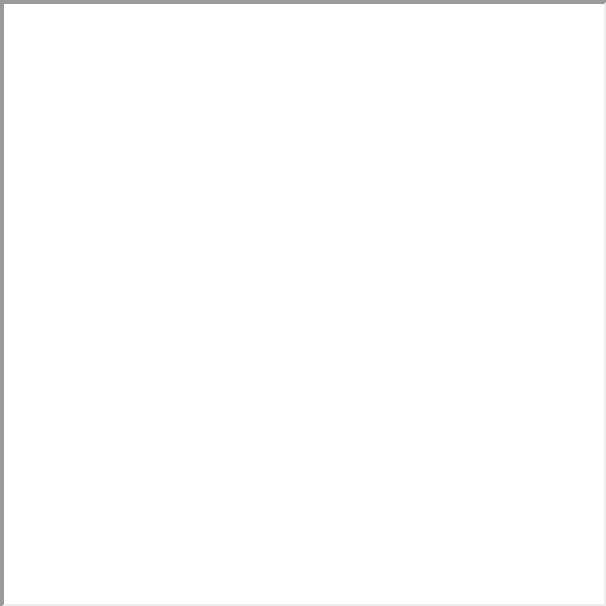
```
your_repository/
├── create_account.py
├── learn.py
├── manage_account.py
├── streamlit_app.py
└── trial.py
```


streamlit_app.py:

```
import streamlit as st

pages = {
    "Your account": [
        st.Page("create_account.py", title="Create your account"),
        st.Page("manage_account.py", title="Manage your account"),
    ],
    "Resources": [
        st.Page("learn.py", title="Learn about us"),
        st.Page("trial.py", title="Try it out"),
    ],
}

pg = st.navigation(pages)
pg.run()
```



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

Example 3: Stateful widgets across multiple pages

Call widget functions in your entrypoint file when you want a widget to be stateful across pages. Assign keys to your common widgets and access their values through Session State within your pages.

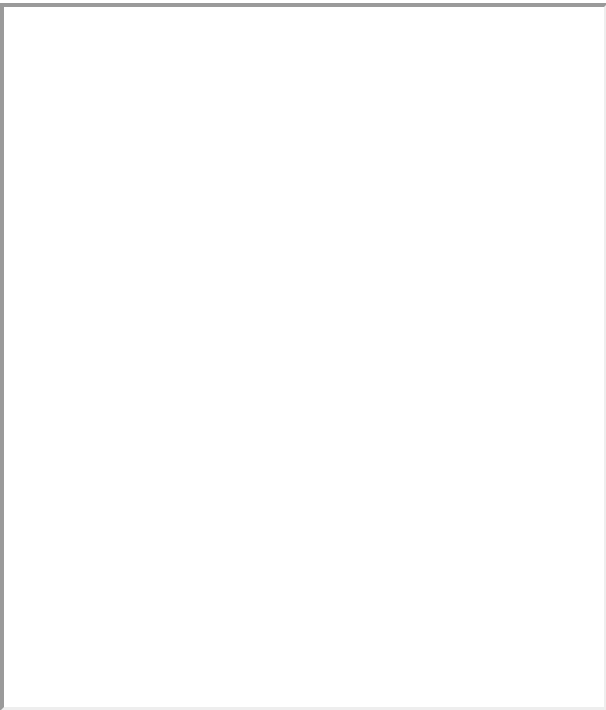
```
import streamlit as st

def page1():
    st.write(st.session_state.foo)

def page2():
    st.write(st.session_state.bar)

# Widgets shared by all the pages
st.sidebar.selectbox("Foo", ["A", "B", "C"], key="foo")
st.sidebar.checkbox("Bar", key="bar")

pg = st.navigation([st.Page(page1), st.Page(page2)])
pg.run()
```



Still have questions?

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

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

