

Untitled1

January 22, 2023

1 Day 3

`st.button`

`st.button` allows the display of a button widget.

1.1 What we're building?

A simple app that performs conditionally prints out alternative messages depending on whether the button was pressed or not.

Flow of the app:

- By default, the app prints Goodbye
- Upon clicking on the button, the app displays the alternative message Why hello there

1.2 Code

Here's the code to implement the above mentioned app:

```
[ ]: import streamlit as st

st.header('st.button')

if st.button('Say hello'):
    st.write('Why hello there')
else:
    st.write('Goodbye')
```

1.3 Line-by-line explanation

The very first thing to do when creating a Streamlit app is to start by importing the `streamlit` library as `st` like so:

```
[ ]: import streamlit as st
```

This is followed by creating a header text for the app:

```
[ ]: st.header('st.button')
```

Next, we will use conditional statements `if` and `else` for printing alternative messages.

```
[ ]: if st.button('Say hello'):
      st.write('Why hello there')
else:
      st.write('Goodbye')
```

As we can see from the above code box, the `st.button()` command accepts the `label` input argument of `Say hello`, which is the text that the button displays.

The `st.write` command is used to print text messages of either `Why hello there` or `Goodbye` depending on whether the button was clicked or not, which is implemented via:

```
[ ]: st.write('Why hello there')
```

and

```
[ ]: st.write('Goodbye')
```

It is important to note that the above `st.write` statements are placed under the `if` and `else` conditions in order to perform the above mentioned process of alternative displaying of messages