# Streamlit 코드 실행 방법

1. 코드 작성 및 저장

파일은 반드시 UTF-8 형식으로 저장해야 한다. 아니면 영문은 정상적이나 한글은 깨진다.

1. Streamlit 실행

(chromadb) dharma6872@DESKTOP-3LV1LJE:~/apps/support\_and\_resitence$ streamlit run app.py

Collecting usage statistics. To deactivate, set browser.gatherUsageStats to False.

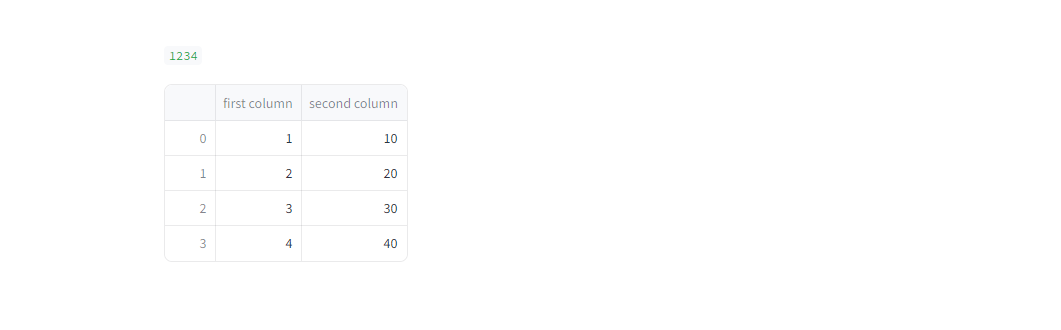
You can now view your Streamlit app in your browser.

Network URL: http://172.30.1.28:8501

External URL: http://112.169.61.42:8501

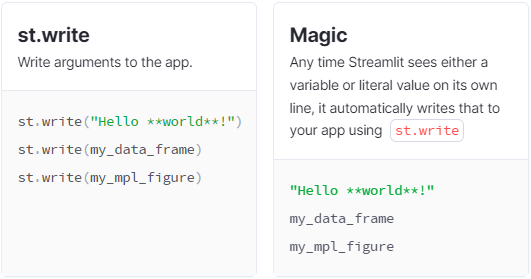
1. 결과 확인

웹 브라우저를 실행하고, 주소란에 <http://172.30.1.28:8501> 입력하고 결과 화면을 확인한다.



# Write and magic

Streamlit은 APP 화면에 정보를 표시하는 두 가지 쉬운 방법을 제공하는데, 가장 일반적인 방법은 st.write와 magic을 사용하는 것이다.



## st.write

* 여러 개의 파라미터를 전달할 수 있으며, 전달된 모든 파라미터는 화면에 출력된다.
* 이 함수의 동작은 파라미터의 유형에 따라 달라진다.
* 반환 값이 없다.

### 예제 1

import streamlit as st

st.write('Hello, \*World!\* :sunglasses:')



### 예제 2

import streamlit as st

import pandas as pd

st.write(1234)

st.write(pd.DataFrame({

'first column': [1, 2, 3, 4],

'second column': [10, 20, 30, 40],

}))



### 예제 3

import streamlit as st

import pandas as pd

data\_frame = pd.DataFrame({

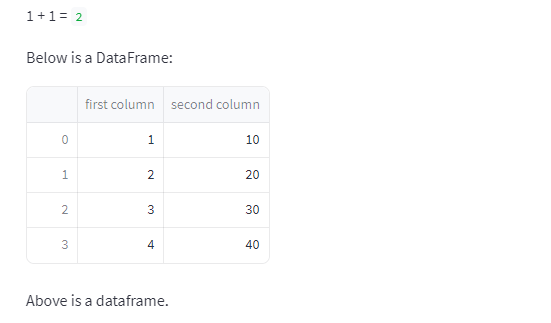
'first column': [1, 2, 3, 4],

'second column': [10, 20, 30, 40],

})

st.write('1 + 1 = ', 2)

st.write('Below is a DataFrame:', data\_frame, 'Above is a dataframe.')



### 예제 4

import streamlit as st

import pandas as pd

import numpy as np

import altair as alt

df = pd.DataFrame(

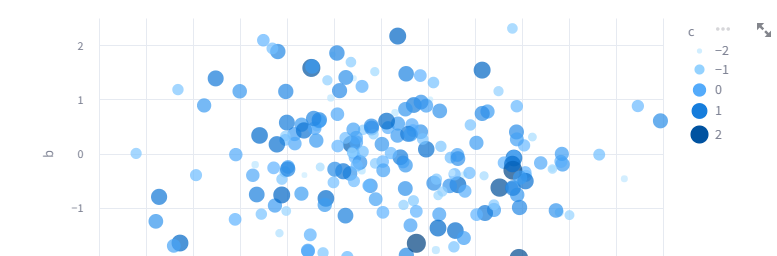
np.random.randn(200, 3),

columns=['a', 'b', 'c'])

c = alt.Chart(df).mark\_circle().encode(

x='a', y='b', size='c', color='c', tooltip=['a', 'b', 'c'])

st.write(c)



## Magic

Magic 명령어는 명시적 출력 함수를 사용하지 않아도 화면에 자료를 출력하는 기능이다. 내부적으로 st.write() 함수를 사용한다.

### 예제 1

# 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({'col1': [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

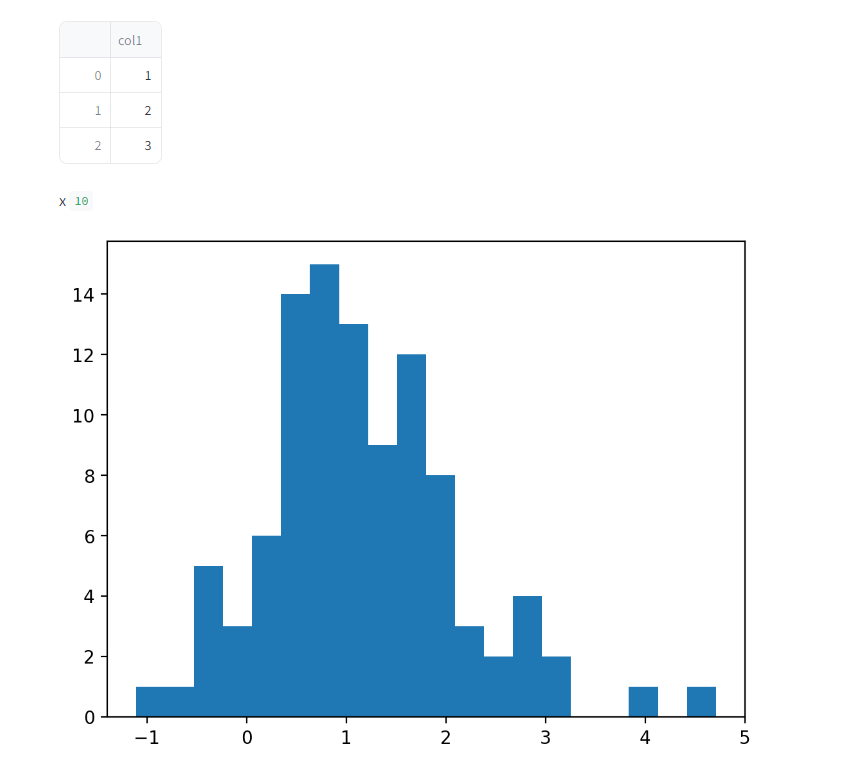
import numpy as np

arr = np.random.normal(1, 1, size=100)

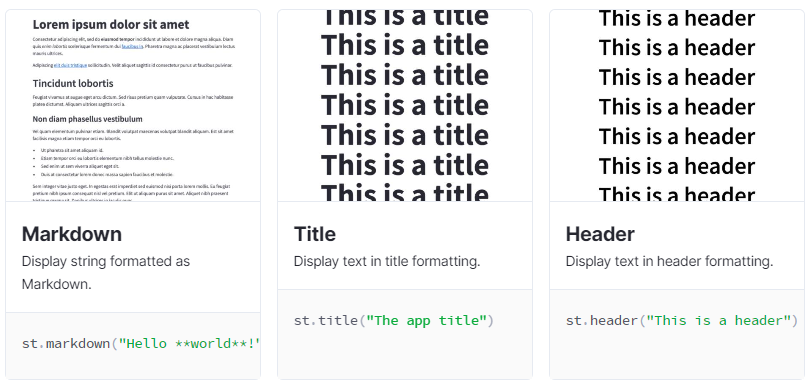
fig, ax = plt.subplots()

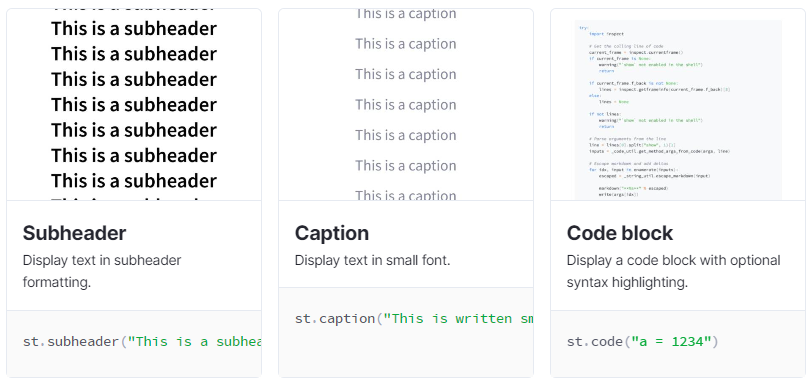
ax.hist(arr, bins=20)

fig # Draw a Matplotlib chart



# Text elements





|  |  |
| --- | --- |
| Title | st.title |
| 2 heading levels | st.header and st.subheader |
| Pure text | st.text |
| Markdown | st.markdown |
| 소스코드 출력 | st.code |
|  |  |

## st.title

import streamlit as st

st.title('This is a title')

st.title('\_Streamlit\_ is :blue[cool] :sunglasses:')

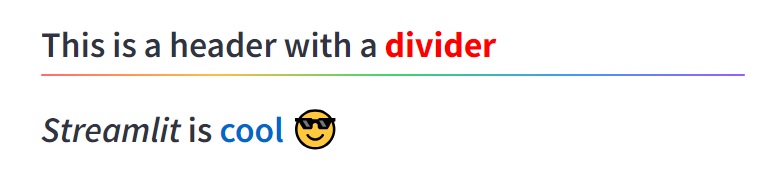


## st.header

import streamlit as st

st.header('This is a header with a divider', divider='rainbow')

st.header('\_Streamlit\_ is :blue[cool] :sunglasses:')

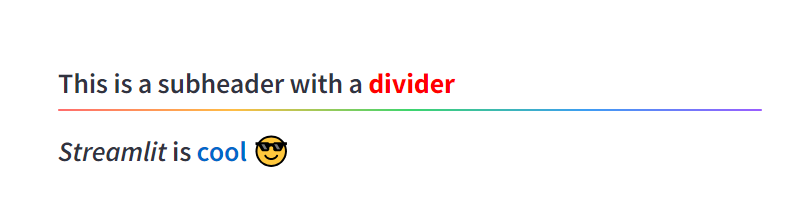


## st.subheader

import streamlit as st

st.subheader('This is a subheader with a divider', divider='rainbow')

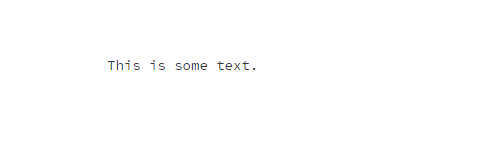
st.subheader('\_Streamlit\_ is :blue[cool] :sunglasses:')



## st.text

import streamlit as st

st.text('This is some text.')



## st.markdown

import streamlit as st

st.markdown("\*Streamlit\* is \*\*really\*\* \*\*\*cool\*\*\*.")

st.markdown('''

:red[Streamlit] :orange[can] :green[write] :blue[text] :violet[in]

:gray[pretty] :rainbow[colors].''')

st.markdown("Here's a bouquet &mdash;\

:tulip::cherry\_blossom::rose::hibiscus::sunflower::blossom:")

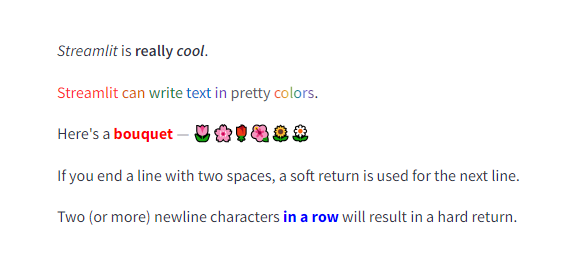
multi = '''If you end a line with two spaces,

a soft return is used for the next line.

Two (or more) newline characters in a row will result in a hard return.

'''

st.markdown(multi)



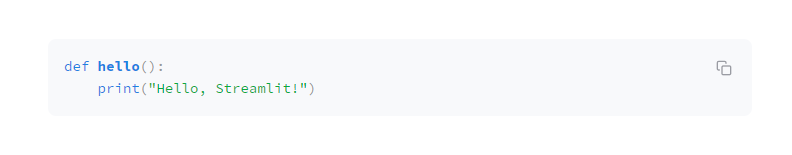
## st.code

import streamlit as st

code = '''def hello():

print("Hello, Streamlit!")'''

st.code(code, language='python')



# Data elements

## st.dataframe

동적인 테이블 데이터 출력

import streamlit as st

import pandas as pd

import numpy as np

df = pd.DataFrame(np.random.randn(50, 20), columns=("col %d" % i for i in range(20)))

st.dataframe(df) # Same as st.write(df)



## st.table

정적인 테이블 데이터 출력

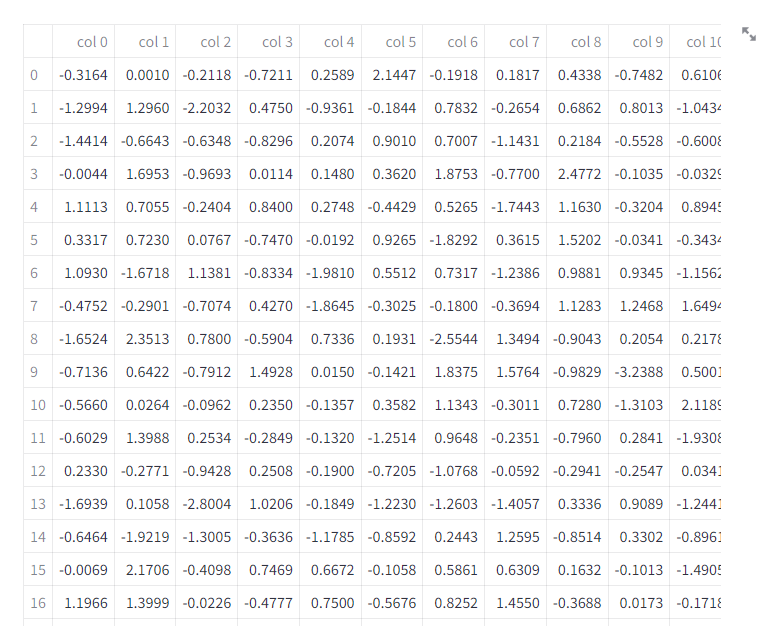
import streamlit as st

import pandas as pd

import numpy as np

df = pd.DataFrame(np.random.randn(50, 20), columns=("col %d" % i for i in range(20)))

st.table(df) # Same as st.write(df)



# Chart elements

## st.bar\_chart

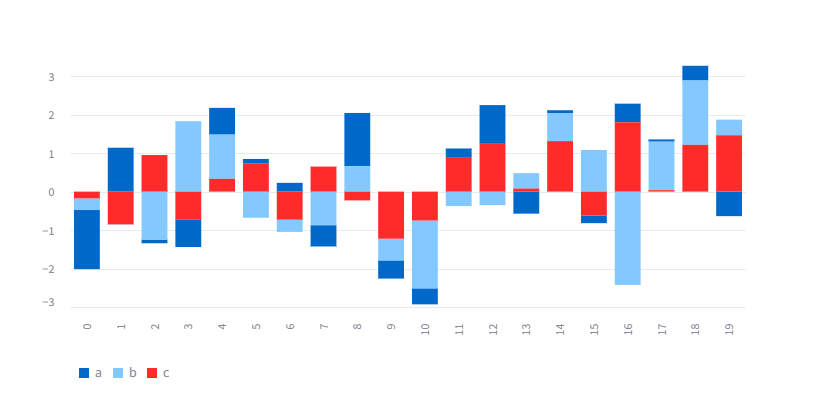
import streamlit as st

import pandas as pd

import numpy as np

chart\_data = pd.DataFrame(np.random.randn(20, 3), columns=["a", "b", "c"])

st.bar\_chart(chart\_data)



## st.line\_chart

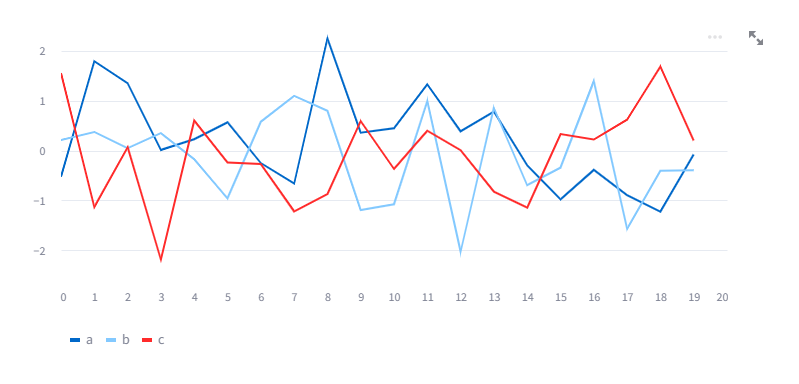
import streamlit as st

import pandas as pd

import numpy as np

chart\_data = pd.DataFrame(np.random.randn(20, 3), columns=["a", "b", "c"])

st.line\_chart(chart\_data)



## st.scatter\_chart

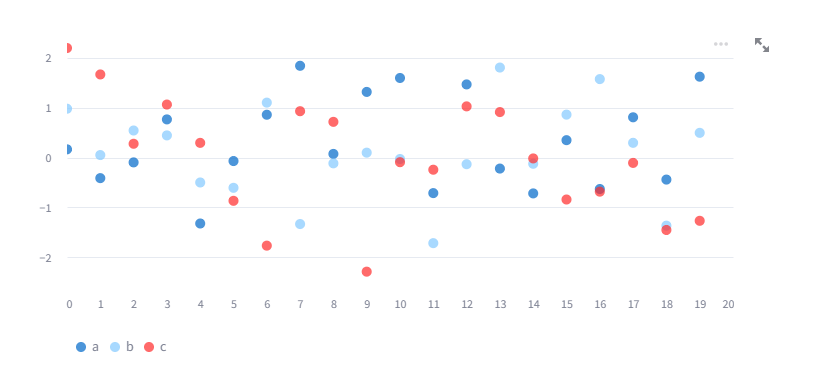
import streamlit as st

import pandas as pd

import numpy as np

chart\_data = pd.DataFrame(np.random.randn(20, 3), columns=["a", "b", "c"])

st.scatter\_chart(chart\_data)



## st.pyplot

import streamlit as st

import matplotlib.pyplot as plt

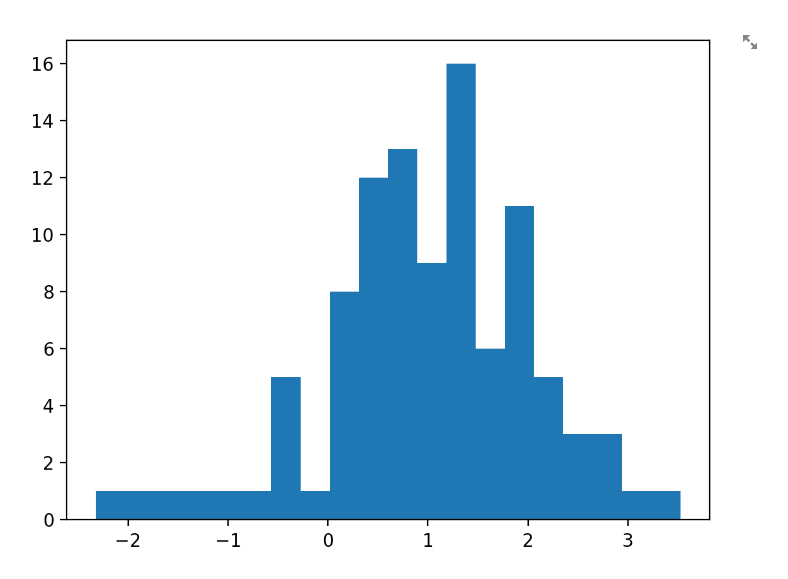
import numpy as np

arr = np.random.normal(1, 1, size=100)

fig, ax = plt.subplots()

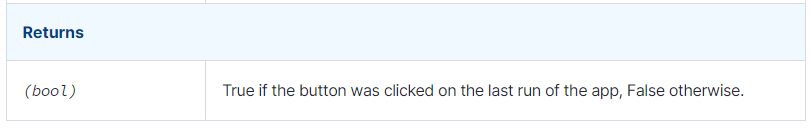
ax.hist(arr, bins=20)

st.pyplot(fig)



# Input widgets

## st.button



import streamlit as st

st.button("Reset", type="primary")

if st.button('Say hello'):

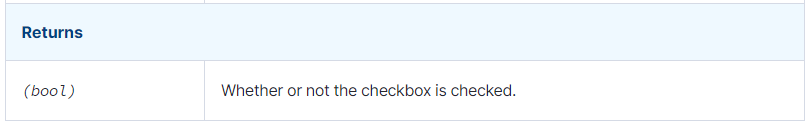
st.write('Why hello there')

else:

st.write('Goodbye')



## st.checkbox



import streamlit as st

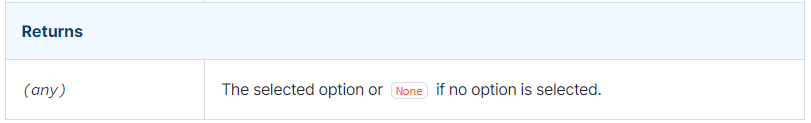
agree = st.checkbox('I agree')

if agree:

st.write('Great!')



## st.radio



import streamlit as st

genre = st.radio(

"What's your favorite movie genre",

[":rainbow[Comedy]", "\*\*\*Drama\*\*\*", "Documentary :movie\_camera:"],

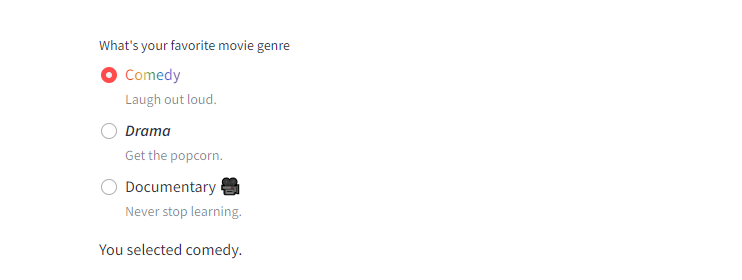
captions = ["Laugh out loud.", "Get the popcorn.", "Never stop learning."])

if genre == ':rainbow[Comedy]':

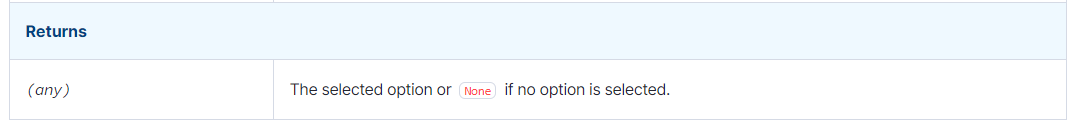
st.write('You selected comedy.')

else:

st.write("You didn\'t select comedy.")



## st.selectbox



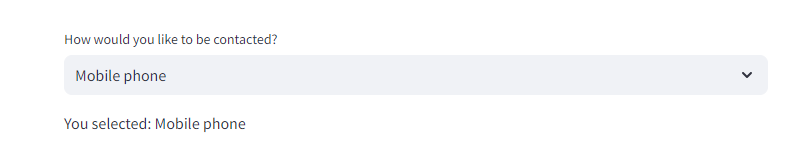
import streamlit as st

option = st.selectbox(

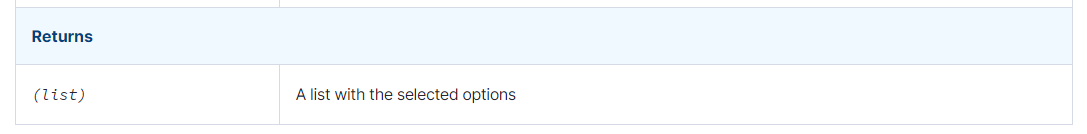
'How would you like to be contacted?',

('Email', 'Home phone', 'Mobile phone'))

st.write('You selected:', option)



## st.multiselect



import streamlit as st

options = st.multiselect(

'What are your favorite colors',

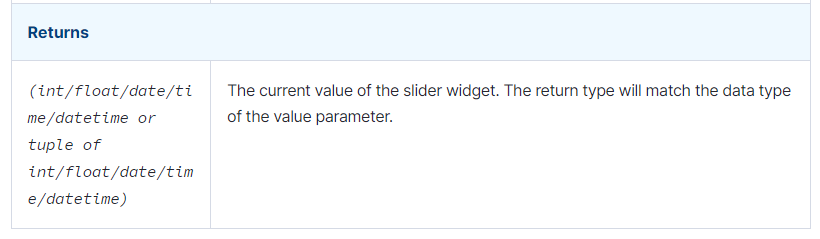
['Green', 'Yellow', 'Red', 'Blue'],

['Yellow', 'Red'])

st.write('You selected:', options)



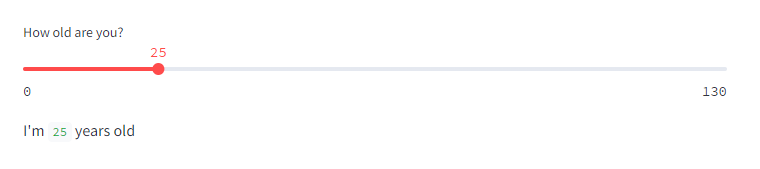
## st.slider



import streamlit as st

age = st.slider('How old are you?', 0, 130, 25)

st.write("I'm ", age, 'years old')



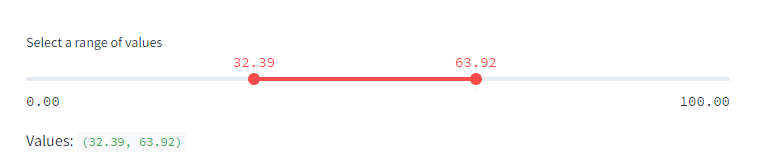
import streamlit as st

values = st.slider(

'Select a range of values',

0.0, 100.0, (25.0, 75.0))

st.write('Values:', values)



import streamlit as st

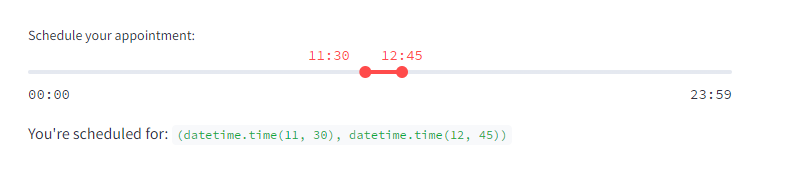
from datetime import time

appointment = st.slider(

"Schedule your appointment:",

value=(time(11, 30), time(12, 45)))

st.write("You're scheduled for:", appointment)



import streamlit as st

from datetime import datetime

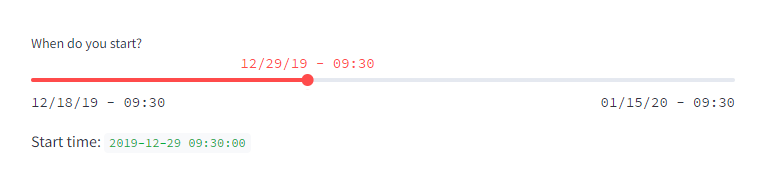
start\_time = st.slider(

"When do you start?",

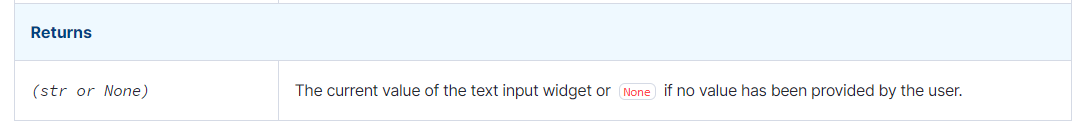
value=datetime(2020, 1, 1, 9, 30),

format="MM/DD/YY - hh:mm")

st.write("Start time:", start\_time)



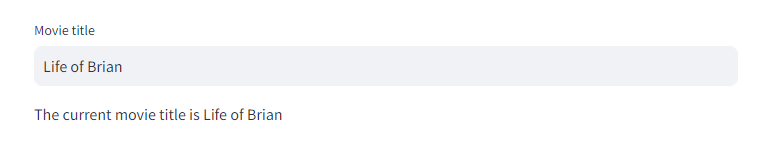
## st.text\_input



import streamlit as st

title = st.text\_input('Movie title', 'Life of Brian')

st.write('The current movie title is', title)



import streamlit as st

# Store the initial value of widgets in session state

if "visibility" not in st.session\_state:

st.session\_state.visibility = "visible"

st.session\_state.disabled = False

col1, col2 = st.columns(2)

with col1:

st.checkbox("Disable text input widget", key="disabled")

st.radio(

"Set text input label visibility 👉",

key="visibility",

options=["visible", "hidden", "collapsed"],

)

st.text\_input(

"Placeholder for the other text input widget",

"This is a placeholder",

key="placeholder",

)

with col2:

text\_input = st.text\_input(

"Enter some text 👇",

label\_visibility=st.session\_state.visibility,

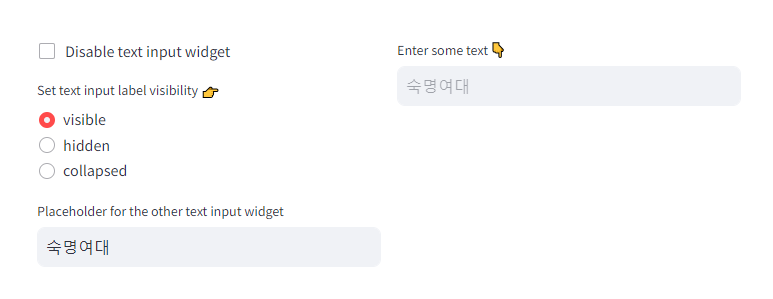
disabled=st.session\_state.disabled,

placeholder=st.session\_state.placeholder,

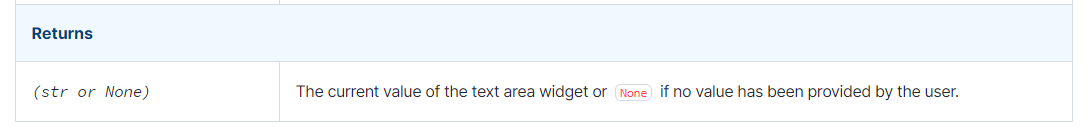
)

if text\_input:

st.write("You entered: ", text\_input)



## st.text\_area



import streamlit as st

txt = st.text\_area(

"Text to analyze",

"It was the best of times, it was the worst of times, it was the age of "

"wisdom, it was the age of foolishness, it was the epoch of belief, it "

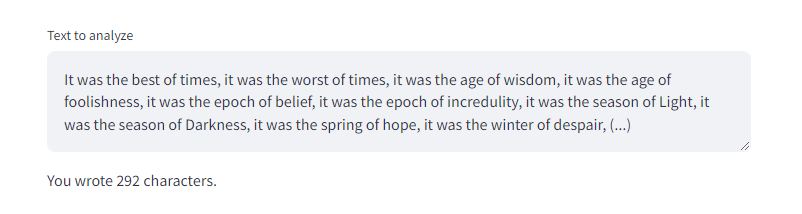
"was the epoch of incredulity, it was the season of Light, it was the "

"season of Darkness, it was the spring of hope, it was the winter of "

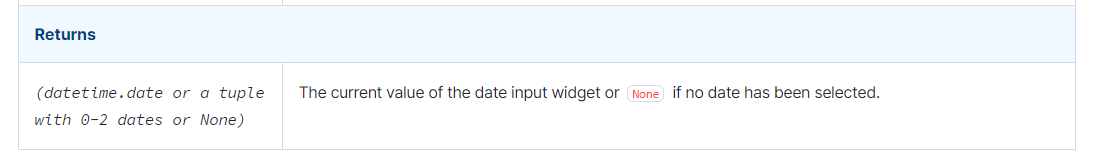
"despair, (...)",

)

st.write(f'You wrote {len(txt)} characters.')



## st.date\_input

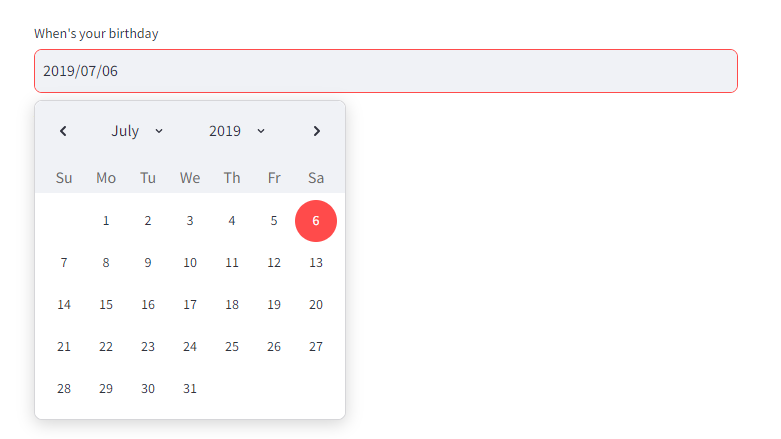


import datetime

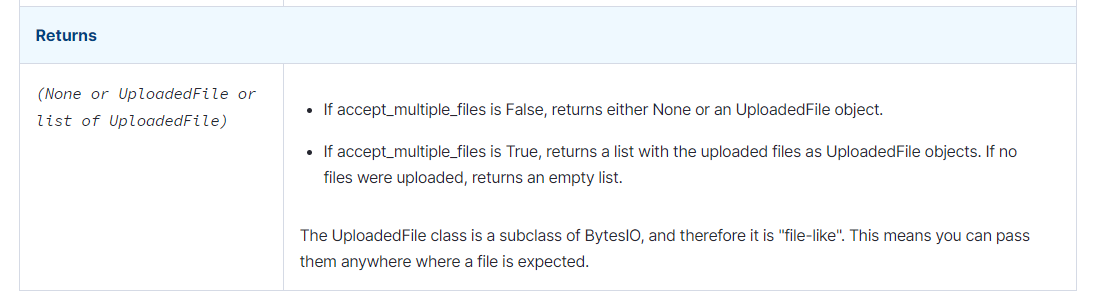
import streamlit as st

d = st.date\_input("When's your birthday", datetime.date(2019, 7, 6))

st.write('Your birthday is:', d)



## st.file\_uploader



import streamlit as st

import pandas as pd

from io import StringIO

uploaded\_file = st.file\_uploader("Choose a file")

if uploaded\_file is not None:

# To read file as bytes:

bytes\_data = uploaded\_file.getvalue()

st.write(bytes\_data)

# To convert to a string based IO:

stringio = StringIO(uploaded\_file.getvalue().decode("utf-8"))

st.write(stringio)

# To read file as string:

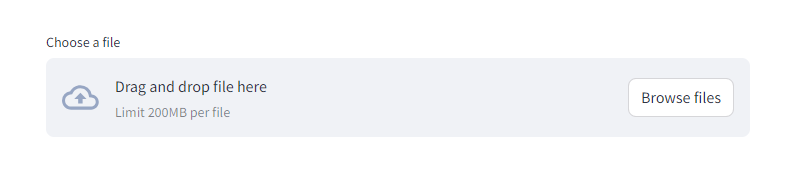
string\_data = stringio.read()

st.write(string\_data)

# Can be used wherever a "file-like" object is accepted:

dataframe = pd.read\_csv(uploaded\_file)

st.write(dataframe)



import streamlit as st

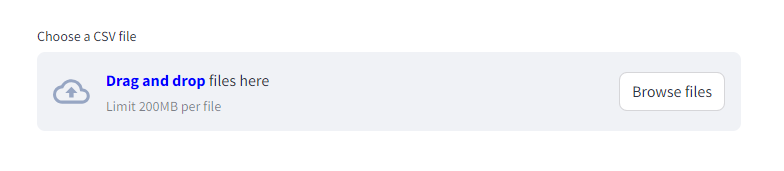
uploaded\_files = st.file\_uploader("Choose a CSV file", accept\_multiple\_files=True)

for uploaded\_file in uploaded\_files:

bytes\_data = uploaded\_file.read()

st.write("filename:", uploaded\_file.name)

st.write(bytes\_data)



# Media elements

# Layouts and containers

## st.sidebar

## st.columns

## st.tabs

# Status elements

# Control flow

# Utilities

# Mutate charts

# State management

# Performance

# Personalization

# Connections and databases