

# streamlit 使用文档

使用 streamlit 中可能用到的库。

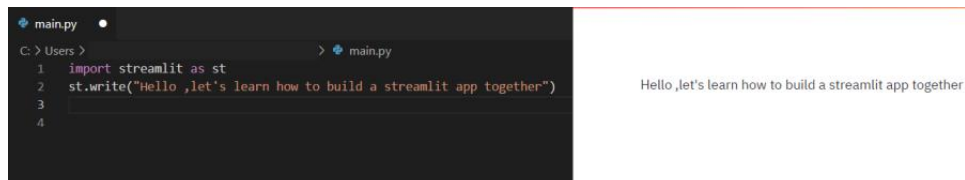
```
import streamlit as st
import pandas as pd
import numpy as np
import plotly.express as px
from plotly.subplots import make_subplots
import plotly.graph_objects as go
import matplotlib.pyplot as plt
```

## 显示文本

`st.write()`：此函数用于向 Web 应用程序添加任何内容，从格式化字符串到

matplotlib 图表、Altair 图表、绘图图表、数据框、Keras 模型等中的图表。

```
import streamlit as st
st.write("Hello ,let's learn how to build a streamlit app together")
```



`st.title()`：添加应用程序的标题

`st.header()`：设置节的标题

`st.subheader()`：设置节的子标题

`st.text()`：编写特定图形的描述

`st.markdown()`：将文本显示为 Markdown

`st.latex()`：在仪表板中显示数学表达式

`st.write()`：显示所有内容，例如绘图、数据框、函数、模型等

`st.sidebar()`: 在侧边栏上显示数据

`st.dataframe()`: 显示数据框

`st.map()`: 以单行代码等显示地图

`st.caption()`: 写字幕

`st.code()`: 设置密码

```
st.title("this is the app title")
st.header("this is the markdown")
st.markdown("this is the header")
st.subheader("this is the subheader")
st.caption("this is the caption")
st.code("x=2021")
st.latex(r''' a + a r^1 + a r^2 + a r^3 ''')
```



```
main.py X
C:\Users\> .\main.py
1 import streamlit as st
2
3 st.title("this is the app title")
4 st.markdown("this is the markdown")
5 st.header("this is the header")
6 st.subheader("this is the subheader")
7 st.caption("this is the caption")
8 st.code("x=2021")
9 st.latex(r''' a + a r^1 + a r^2 + a r^3 ''')
10
11
```

## this is the app title

this is the markdown

this is the header

this is the subheader

this is the caption

文本x=2021

$$a + ar^1 + ar^2 + ar^3$$

Made with Streamlit

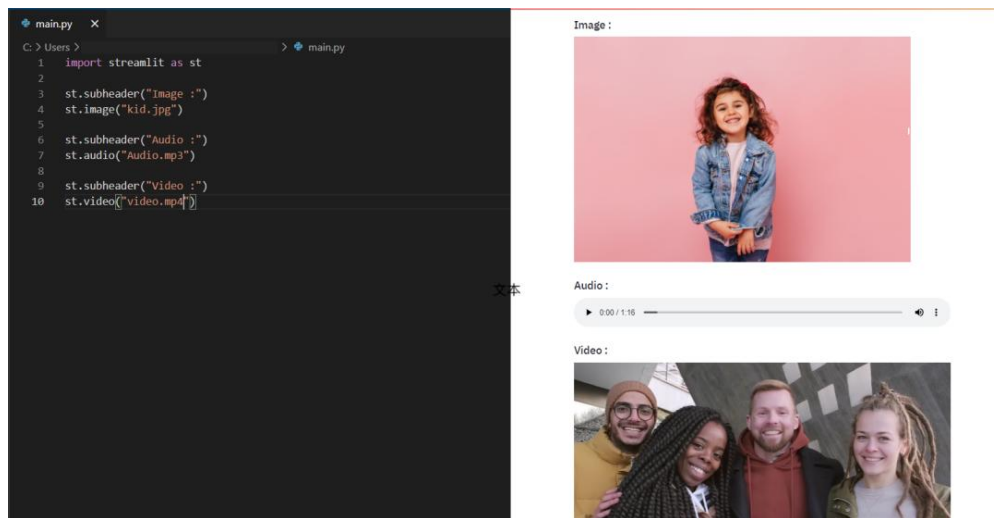
## 显示图像、视频或音频文件

`st.image()`: 显示图像

`st.audio()`: 显示音频

`st.video()`: 显示视频

```
st.image("kid.jpg")
st.audio("Audio.mp3")
st.video("video.mp4")
```



## 输入小部件

小部件是最重要的用户界面组件。Streamlit 有各种各样的小部件，允许通过按钮、滑动条、文本输入等直接将交互性嵌入到应用程序中。

`st.checkbox()`: 返回一个布尔值。选中该框时，返回 True 值，否则返回 False 值。

`st.button()`: 按钮小部件。

`st.radio()`: 单选按钮小部件。

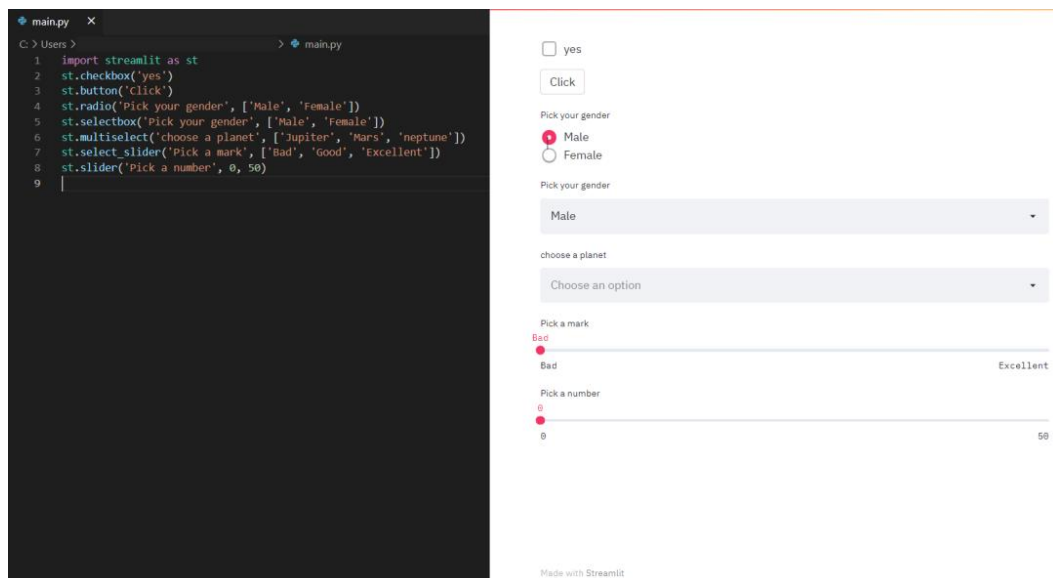
`st.selectbox()`: 选择小部件。

`st.multiselect()`: 多选小部件。

`st.select_slider()`: 选择滑块小部件。

`st.slider()`: 滑块小部件。

```
st.checkbox('yes')
st.button('Click')
st.radio('Pick your gender', ['Male', 'Female'])
st.selectbox('Pick your gender', ['Male', 'Female'])
st.multiselect('choose a planet', ['Jupiter', 'Mars', 'neptune'])
st.select_slider('Pick a mark', ['Bad', 'Good', 'Excellent'])
st.slider('Pick a number', 0, 50)
```



`st.number_input()`: 显示数字输入小部件

`st.text_input()`: 显示文本输入小部件

`st.date_input()`: 显示日期输入小部件以选择日期

`st.time_input()`: 显示时间输入小部件以选择时间

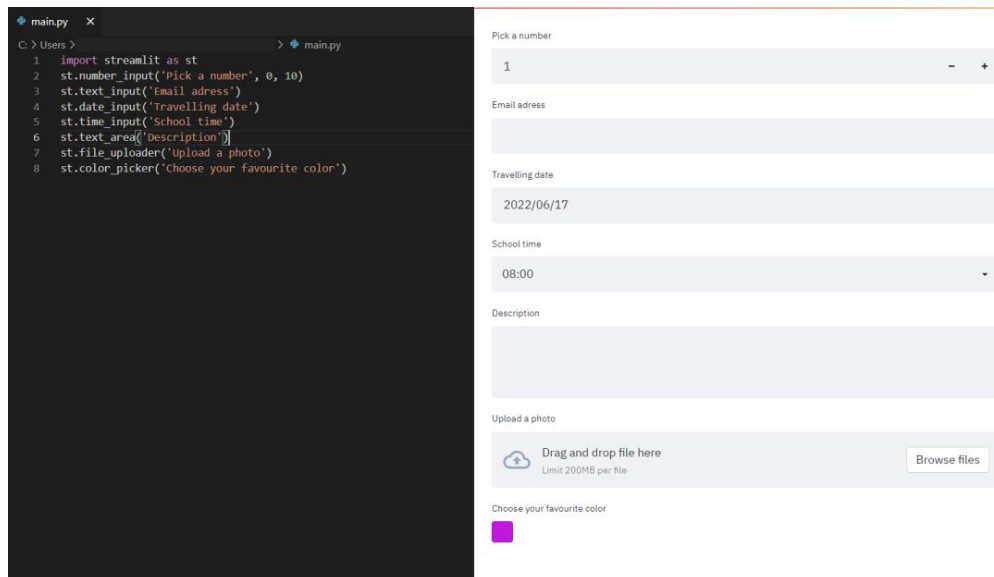
`st.text_area()`: 显示具有多行文本的文本输入小部件

`st.file_uploader()`: 显示文件上传器小部件

`st.color_picker()`: 显示颜色选择器小部件以选择颜色

```
st.number_input('Pick a number', 0, 10)
st.text_input('Email address')
st.date_input('Travelling date')
st.time_input('School time')
```

```
st.text_area('Description')
st.file_uploader('Upload a photo')
st.color_picker('Choose your favorite color')
```



## 显示状态

`st.success()`: 显示成功消息

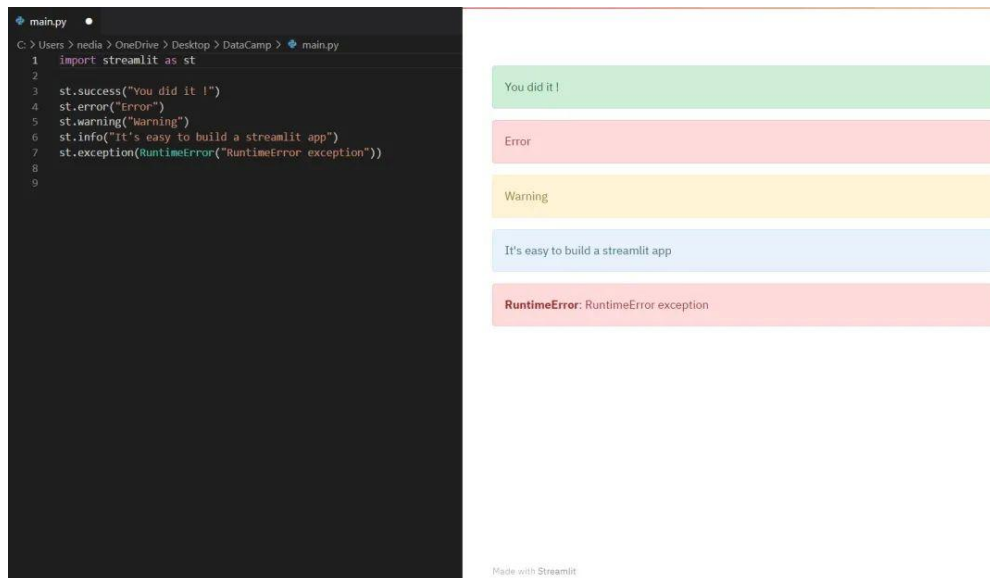
`st.error()`: 显示错误消息

`st.warnig()`: 显示警告信息

`st.info()`: 显示信息性消息

`st.exception()`: 显示异常消息。

```
st.success("You did it !")
st.error("Error")
st.warnig("Warning")
st.info("It's easy to build a streamlit app")
st.exception(RuntimeError("RuntimeError exception"))
```



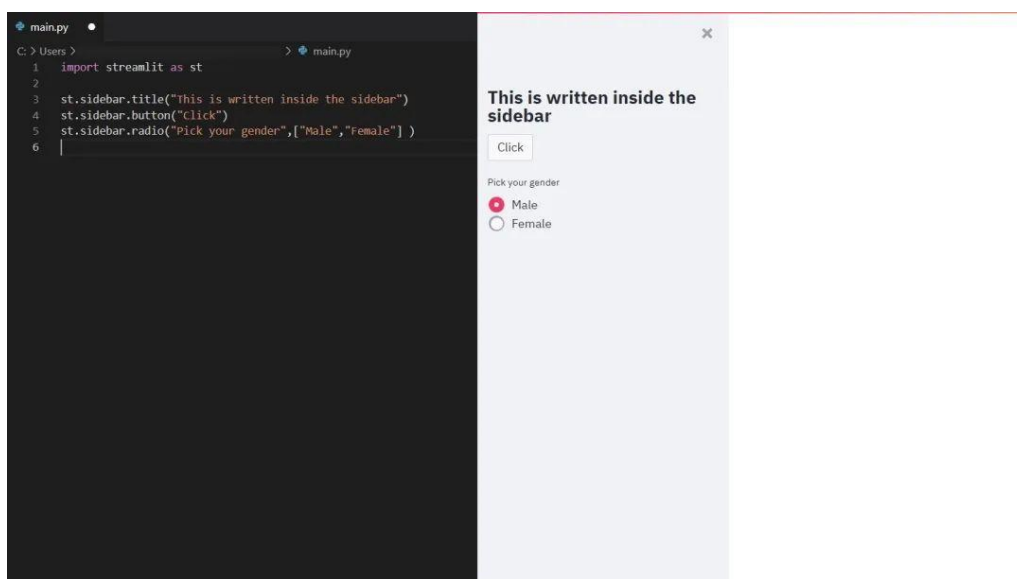
## 侧边栏

`st.sidebar()`：可以在应用程序界面中创建一个侧边栏并在其中放置元素，这将在应用程序更有条理且更易于理解。

```
st.sidebar.title("This is written inside the sidebar")
```

```
st.sidebar.button("click")
```

```
st.sidebar.radio("Pick your gender", ["Male", "Female"])
```



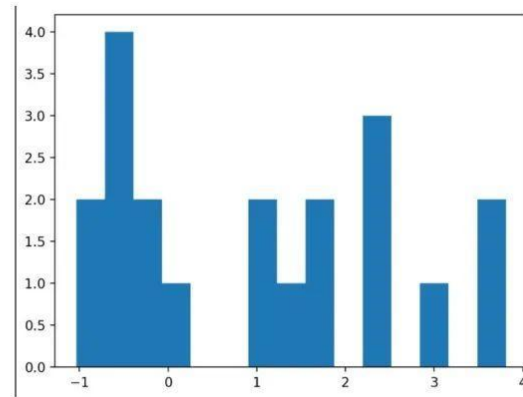
# 显示图表

`st.pyplot()`: 显示 `matplotlib.pyplot` 图。

```
import streamlit as st
import matplotlib.pyplot as plt
import numpy as np

rand=np.random.normal(1, 2, size=20)
fig, ax = plt.subplots()
ax.hist(rand, bins=15)
st.pyplot(fig)
```

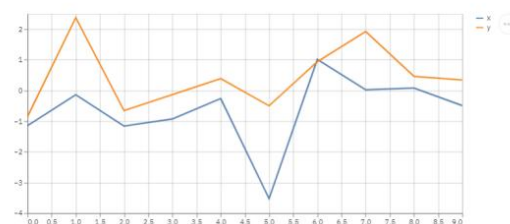
```
main.py x
C:\Users > > main.py
1 import streamlit as st
2 import matplotlib.pyplot as plt
3 import numpy as np
4
5
6 rand = np.random.normal(1, 2, size=20)
7 fig, ax = plt.subplots()
8 ax.hist(rand, bins=15) #,color="pink"
9 st.pyplot(fig)
```



`st.line_chart()`: 显示折线图

```
df= pd.DataFrame(
    np.random.randn(10, 2),
    columns=['x', 'y'])
st.line_chart(df)
```

```
main.py
C:\Users > > main.py
1 import streamlit as st
2 import pandas as pd
3 import numpy as np
4 df = pd.DataFrame(
5     np.random.randn(10, 2),
6     columns=['x', 'y'])
7 st.line_chart(df)
8 |
```



# 数据可视化

## 加载数据集

例：使用 COVID-19 数据集进行仪表板可视化。

```
@st.cache(suppress_st_warning=True)
def load_data():
    data = pd.read_csv('data.csv')
    return data
covid_data = load_data()

st.sidebar.checkbox("世卫组织冠状病毒 (COVID-19) 仪表板", True, key=1)
df = covid_data.groupby(by=['Country']).sum().reset_index()
select = st.sidebar.selectbox('选择一个国家', df['Country'])

# 获取选择框选中的状态
state_data = df[df['Country'] == select]
select_status = st.sidebar.radio("世卫组织冠状病毒 (COVID-19) 病例",
                                ('新增病例', '累计病例', '新增死亡病例', '累计死亡病例'))
```

使用复选框来选择按状态进行的分析。选择框将显示受 COVID-19 影响的状态列表。用于选择活动、已确认、死亡或已恢复案例的单选按钮。

## 绘制图形

```
def get_total_dataframe(dataset):
    total_dataframe = pd.DataFrame({
        '病例分类': ['新增病例', '累计病例', '新增死亡病例', '累计死亡病例'],
        '病例数': (dataset.iloc[0]['New_cases'],
                  dataset.iloc[0]['Cumulative_cases'],
                  dataset.iloc[0]['New_deaths'], dataset.iloc[0]['Cumulative_deaths'])})
    return total_dataframe

state_total = get_total_dataframe(state_data)

if st.sidebar.checkbox("世卫组织冠状病毒 (COVID-19) 仪表板", True, key=2):
    st.markdown("## **世卫组织冠状病毒 (COVID-19) 仪表板**")
    st.markdown("### %s 国家总新增病例、累计病例、新增死亡病例和累计死亡病例" % (select))
    if not st.checkbox('Hide Graph', False, key=1):
```



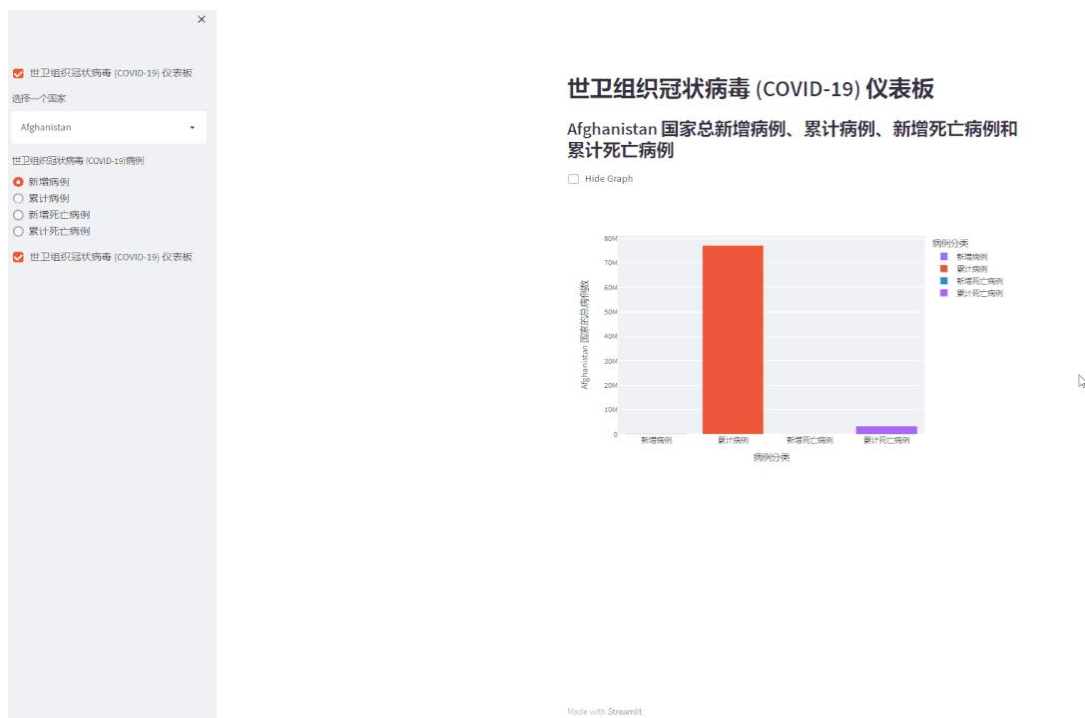
```

state_total_graph = px.bar(
    state_total,
    x='病例分类',
    y='病例数',
    labels={'病例数': '%s 国家的总病例数' % (select)},
    color='病例分类')
st.plotly_chart(state_total_graph)

```

执行此代码后，可以根据需要的状态选择案例。 `get_total_dataframe` 方法用于获取数据集以绘制所选状态的图形。

为了绘制图形，使用了 `plotly.express` 库方法。最后，使用 `st.plotly_chart()` 显示图形。



## 显示数据框或表格

使用 `st.dataframe()` 和 `st.table()` 在表格视图中查看数据框。

```

def get_table():
    datatable = df.sort_values(by=['Cumulative_cases'], ascending=False)
    return datatable

```

```
datatable = get_table()

st.markdown("### 世界各国的 Covid-19 病例分析")

st.markdown("下表为您提供了 %s 国家 Covid-19 总新增病例、累计病例、新增死亡病例和累计死亡病例的实时分析。"%(select))

st.dataframe(datatable) # will display the dataframe

st.table(state_total)# will display the table
```

世卫组织冠状病毒 (COVID-19) 仪表盘

选择一个国家

Afghanistan

世卫组织冠状病毒 (COVID-19) 病例

☒ 新增病例

☐ 累计病例

☐ 新增死亡病例

☐ 累计死亡病例

☒ 世卫组织冠状病毒 (COVID-19) 仪表盘

世卫组织冠状病毒 (COVID-19) 仪表盘

Afghanistan 国家总新增病例、累计病例、新增死亡病例和累计死亡病例

☐ Hide Graph

病例分类	病例数
新增病例	~15K
累计病例	~75K
新增死亡病例	~1K
累计死亡病例	~5K

世界各国的Covid-19病例分析

下表为您提供了 Afghanistan 国家Covid-19总新增病例、累计病例、新增死亡病例和累计死亡病例的实时分析。

	Country	New_cases	Cumulative_cases	New_deaths	Cumulative_deaths
226	United States of Ameri	85007630	27739254156	1002546	430192523
96	India	43270577	17312489716	524817	224239164
28	Brazil	31611769	11832577455	668693	304513466
72	France	29114200	6133088383	145062	69919018
210	The United Kingdom	22460453	5799914364	179472	86857764
173	Russian Federation	18391797	5244107633	380333	122230718
78	Germany	27124689	4614832898	140118	58624887