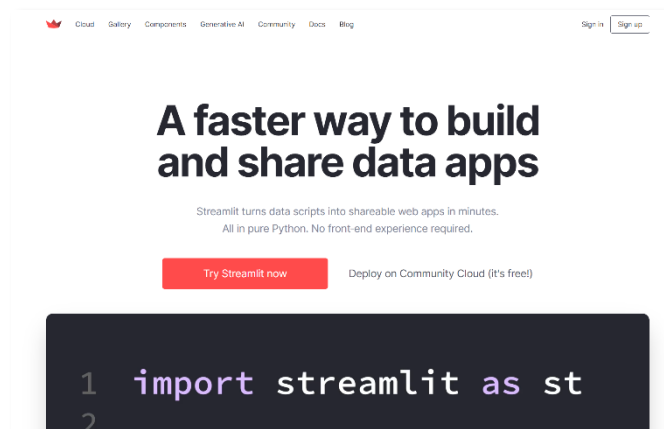


Streamlit



Python で手軽に Web アプリを作成できるフレームワーク。データサイエンスの分野でよく使われる。簡単で、直感的にコードが書け、サーバなどの知識がなくても実行ができる。

■Streamlit の特徴

	特徴	適しているアプリ
Streamlit	コードは直感的。HTML は不要	データサイエンス分野
Flask	シンプルで軽量。大規模開発にも向いている	本格的なアプリ開発
Django	機能が豊富。その分学習コストが高い	多機能のアプリ

◆Streamlit が適している場面

- ・データサイエンス分野で WebUI 付きのアプリを作成する
- ・データの可視化や分析結果を複数人で共有する場合
- ・動作する結果を確認したい
- ・プロトタイプの Web アプリを作成する

◆Flask との違い

Flask はちょっと大変。

- ・HTML とか、SQL とかも必要
- ・環境設定が必要

◆インストール

streamlit は対応ソフトのバージョンが厳格なので、まずは、Python のバージョンを確認する。¹

```
$python --version
```

また、インストール用の pip も最新²のものが良い。

```
$pip install --upgrade pip
```

streamlit のインストール

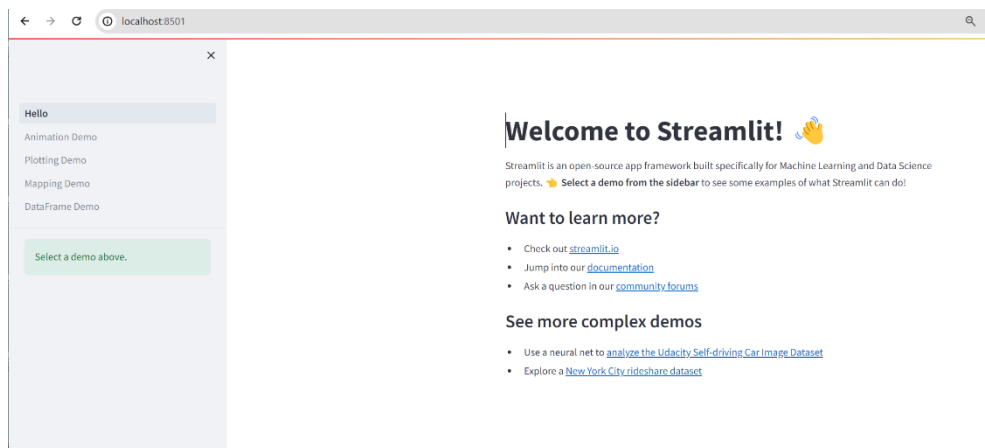
```
$pip install --streamlit
```

◆インストールの確認

インストールが完了したら、コマンドラインから「hello」を命令すると、クライアントの HP が表示される。

```
$streamlit hello
```

hello で表示されるページ



¹ Python のディレクトリがわからない場合は「where python」で調べる。または、Python 自体に聞く。「import sys
print(sys.executable」移動は「cd」

² 1.30.0 (2024.1.23)

■はじめての streamlit

streamlit が以下に簡単に体験してもらいます。³

ブラウザで結果表示するのに、HTML を一切書かなくて済みます。

次のコードを記述して「app.py」として保存します。

```
import streamlit as st
st.title('タイトル')
st.caption('キャプション')
```

・ファイルの実行

```
$streamlit run <ファイル名>
```

このファイルを実行するとブラウザが自動的に表示される

また、同時に「ローカルホストとポート番号」、「IP アドレス」がコマンドラインに表示される。

```
Welcome to Streamlit. Check out our demo in your browser.
Local URL: http://localhost:8501
Network URL: http://192.168.3.22:8501
```

■表示

◆テキスト表示関連

st.title("title")	: タイトル
st.header("header")	: ヘッダー
st.write("write")	: 表示
st.markdown("# markdown")	: マークダウンで表示
st.text("text")	: テキスト表示

³ 普段の授業では Jupyter を使うのですが、拡張子や、実行、ブラウザの表示まで手間がかかるので、今回は VSCode などが扱いやすいかと思います。

◆コードの表示

st.code の第 1 引数に「code」を記載した文字列を指定し、language でプログラミング言語を指定する。コードがそのまま表示される。

```
code='''
import streamlit as st
st.title('ABC')
'''

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

◆画像の埋め込み⁴

画像の表示には Pillow ライブラリを使う。

app.py と同じディレクトリに、表示するイメージをアップロードしておく

Pillow の Image を有効にするために、ヘッダーに PIL と Image を記述する

```
import streamlit as st
from PIL import Image
```

本体の画像読み込みの記述⁵

```
image = Image.open('digi.jpeg')
st.image(image, width=200)
```

Web 上の画像を使いたい場合には、その URL を埋め込むことで表示される。

◆動画の埋め込み

```
video_file = open('way.mp4', 'rb')
video_bytes = video_file.read()
st.video(video_bytes)
```

「rb」モードは、「読み込み専用」(read only) かつ「バイナリー」モードでファイルを開くための指定。

⁴ 画像の表示には Pillow ライブラリを使う

⁵ OpenCV で読み込んだ ndarray オブジェクトの表示も可能

◆データ入力と表示

画面上にテキストボックスを表示して、入力された値を Python 側で受ける。

```
name = st.text_input('名前')
print(name)
```

画面がリロードされたタイミングに戻り値に値が入る。ボックスからカーソルが離れてもリロードされる。

出力は、「ターミナル」に出力されている。

・送信ボタンを加える

普通は、入力後に「ボタン」で送信するので、次に、ボタンを加える。

さらに、ボタンの状態をターミナルに表示させる。この場合、押されたときに True を返す。

```
submit_btn = st.button('送信')
cancel_btn = st.button('キャンセル')
print(f'submit_btn: {submit_btn}')
print(f'cancel_btn: {cancel_btn}')
```

ボタンの状態は、何もしない場合には「False」になっている。

一端、送信ボタンを押すと submit_btn の内容が「True」に変化する。

・入力文字の画面表示

送信ボタンが押されたとき、入力された値を画面に表示する

```
submit_btn = st.button('送信')
cancel_btn = st.button('キャンセル')
if submit_btn:
    st.text(f'こんにちは！ {name}さん！')
```

「ボタンが押されたとき・・・」は、if 文を使う

・複数のテキストボックスがある場合

「名前」と「住所」など複数の入力ボックスがある場合、その2つの入力が終わるまでリロードをしないようにする必要がある。(カーソルを移動しただけでリロードされるので)

リロードされないためには、with 文の枠組みの中にすべて入れる

```
with st.form(key='profile_form'):
    #textbox
    name = st.text_input('名前')
    address = st.text_input('住所')
    print(name)

    submit_btn = st.form_submit_button('送信')
    cancel_btn = st.form_submit_button('キャンセル')
    if submit_btn:
        st.text(f'こんにちは！{name}さん！郵送先は{address}ですね')
```

◆さまざまなエレメンツ

・セレクトボタン

キャンセルボタンの内容も付け加えた

```
with st.form(key='profile_form'):
    # テキストボックス
    name = st.text_input('名前')
    address = st.text_input('住所')
    # セレクトボックス
    age_category = st.selectbox('年齢層', ('子供', '大人'))
    # 送信ボタン
    submit_btn = st.form_submit_button('送信')
    cancel_btn = st.form_submit_button('キャンセル')

    if submit_btn:
        st.text(f'こんにちは！{name}さん！郵送先は{address}ですね。')
        st.text(f'年齢層：{age_category}')
    if cancel_btn:
        st.text('キャンセルされました。')
```

「selectbox」を「radio」にすればラジオボタンになる。

・複数の選択

各項目はタプルで抱える

```
hobby = st.multiselect('趣味', ('スポーツ', '読書', '映画'))
```

表示では、選択された空く数の値を「、」で区切りながら join で繋いで行きます。

```
if submit_btn:
    st.text(f'こんにちは！{name}さん！郵送先は{address}ですね。')
    st.text(f'年齢層：{age_category}')
    st.text(f'趣味：{"，".join(hobby)}')
```

◆さまざまなウィジェット

```
st.radio("radiobutton", ("radio1", "radio2")) # ラジオボタン
st.text_input("text input") # 文字入力(1行)
st.text_area("text area") # 文字入力(複数行)
st.slider("slider", 0, 100, 50) # スライダー
st.file_uploader("Choose file") # ファイ
```

■サイドバー表示

左袖に必要なウィジェットを簡単に表示することができる。

```
import streamlit as st

st.button("button")
st.selectbox("selectbox", ("select1", "select2"))
st.multiselect("multiselectbox", ("select1", "select2"))
st.radio("radiobutton", ("radio1", "radio2"))
# 以下をサイドバーに表示
st.sidebar.text_input("text input")
st.sidebar.text_area("text area")
st.sidebar.slider("slider", 0, 100, 50)
st.sidebar.file_uploader("Choose file")
```

■pandas と matplotlib を使ったグラフ表示

pandas を利用した表、グラフの表示も可能です

サンプルでは、2020 年と 2021 年の「平均気温」の csv ファイルを読み込み表として表示します。

・表での表示

```
import streamlit as st
import pandas as pd

st.title('あぶり')
st.text('本日はお日柄も良く')

df = pd.read_csv('temp.csv', index_col='mon')
st.dataframe(df)
```

・グラフでの表示

```
import streamlit as st
import pandas as pd

st.title('あぶり')
st.text('本日はお日柄も良く')

df = pd.read_csv('temp.csv', index_col='mon')
st.line_chart(df)
```

◆matplotlib での表示

```
import streamlit as st
import pandas as pd
import matplotlib.pyplot as plt

st.title('あぶり')
st.text('本日はお日柄も良く')

df = pd.read_csv('temp.csv', index_col='mon')
#matplotlib
```


(用語)

・デプロイ

deploy「常駐する」、「配置する」

「デプロイ」とは、ソフトウェア開発プロセスの一環で、完成したアプリケーションやソフトウェアをユーザーが実際に使用できる環境に配置・公開することを指します。

デプロイには以下のような要素が含まれます：

公開：

ソフトウェアをユーザーがアクセス可能な環境に公開します。これはウェブサーバー、クラウドサービス、またはユーザーのデバイス（例：モバイルアプリのインストール）などが該当します。

設定：

ソフトウェアが正しく機能するように必要な設定を行います。これには環境変数の設定やデータベースの接続設定などが含まれます。

依存関係の管理：

ソフトウェアが依存するライブラリやモジュールを適切に配置し、利用可能な状態にします。

バージョン管理：

ソフトウェアの異なるバージョンを管理し、必要に応じて更新やロールバック（以前のバージョンに戻すこと）を行います。

セキュリティ：

ソフトウェアを安全に公開するために、セキュリティ対策を講じます。これには暗号化、アクセス権限の設定、セキュリティパッチの適用などが含まれます。

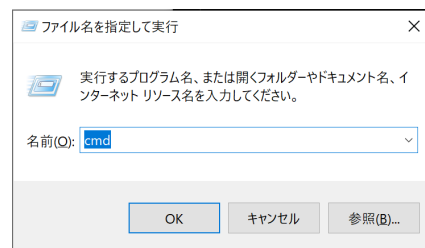
デプロイは、単にソフトウェアをアップロードするだけでなく、安定して効率的に動作するようにするための一連のプロセスです。特に大規模なアプリケーションや多くのユーザーがいる環境では、デプロイのプロセスは非常に複雑になることがあります。

・ウィジェット

部品

コマンドラインの出し方

「Win キー + R」で cmd を指定



または、Anaconda Power Shell などを使う