

# ShachiPy Streamlit ワークショップ

Onur Boyar

2024年3月2日

## セッション 1: Streamlit の紹介

### 1.1 Streamlit の概要

Streamlit とは?

- Streamlit は、データサイエンスや機械学習プロジェクトのためのウェブアプリを作成するためのオープンソースPythonライブラリです。
- 対話型データアプリケーションの迅速なプロトタイピングとデプロイメントを可能にします。

主な特長と利点:

- シンプルなPythonスクリプトでアプリを作成できます。
- フロントエンドの経験は不要です。
- ライブデータとプロットのためのインタラクティブなウィジェット。
- アプリの簡単なデプロイメント。

Example applications:

- Image background remover.
- Chat application
- Molecule optimizer.

### 1.2 インストールとセットアップ

Streamlit のインストール:

```
1 pip install streamlit
```

基本的な Streamlit アプリのセットアップ:

app.py という名前のファイルを作成し、次のコードを追加します:

```

1 import streamlit as st
2
3 st.write('Hello, Streamlit!')

```

アプリを実行するには:

```

1 streamlit run app.py

```

### 1.3 はじめての Streamlit アプリの作成

**Streamlit スクリプトの構造:**

Streamlit スクリプトは Streamlit コマンドを含むPythonスクリプトです。

基本的なコンポーネント:

タイトル:

```

1 st.title('My First Streamlit App')

```

ボタン:

```

1 if st.button('Say hello'):
2     st.write('Hello, World!')

```

テキストとデータの表示:

```

1 st.write('This is a simple text display.')

```

データ表示:

```

1 import pandas as pd
2
3 data = pd.DataFrame({'Column A': [1, 2, 3], 'Column B': [4, 5, 6]})
4 st.write(data)

```

### 1.4 インタラクティブなウィジェット

**ウィジェットの紹介:**

ウィジェットを使用すると、ユーザーはアプリケーションと動的に対話できます。

スライダー:

```

1 age = st.slider('How old are you?', 0, 130, 25)
2 st.write("You're ", age, 'years old')

```

ボタン:

```

1 if st.button('Click me'):
2     st.write('Button clicked!')

```

## セッション 2: Streamlit の高度な概念とアプリケーション

### レイアウトのカスタマイズ

列を使用する:

```
1 col1, col2 = st.columns(2)
2
3 with col1:
4     st.header('Column 1')
5     st.write('Hello, Column 1!')
6
7 with col2:
8     st.header('Column 2')
9     st.write('Hello, Column 2!')
```

### 実際のアプリケーション例

Streamlit での電卓アプリケーションの作成:

```
1 st.title('Simple Calculator')
2
3 # Inputs
4 num1 = st.number_input('Enter first number', format='%f')
5 num2 = st.number_input('Enter second number', format='%f')
6
7 # Operations
8 operation = st.selectbox('Choose an operation:', ['Add', 'Subtract',
9     'Multiply', 'Divide'])
10
11 # Calculate
12 if st.button('Calculate'):
13     if operation == 'Add':
14         result = num1 + num2
15     elif operation == 'Subtract':
16         result = num1 - num2
17     elif operation == 'Multiply':
18         result = num1 * num2
19     elif operation == 'Divide':
20         if num2 != 0:
21             result = num1 / num2
22         else:
23             result = "Error: Division by zero"
24     st.success(f'Result: {result}')
```

### 高度なアプリケーションと Streamlit アプリギャラリーの探索のヒント

- キャッシング、ファイルアップロード、カスタムコンポーネントの作成など、Streamlit の高度な機能についてのドキュメントを参照してください。

- 実際のアプリケーションを見るために Streamlit アプリギャラリーを見てみてください。

## 追加のアプリケーション例

### データ可視化アプリ

```
1 import streamlit as st
2 import pandas as pd
3 import numpy as np
4 import matplotlib.pyplot as plt
5
6 # Set the title of the app
7 st.title('Data Science App with Streamlit')
8
9 # Generate sample data
10 @st.cache # Use caching to generate the data only once
11 def generate_data(n_rows, n_cols):
12     """Generates a DataFrame with random data"""
13     dates = pd.date_range(start="2021-01-01", periods=n_rows, freq="D")
14     data = np.random.randn(n_rows, n_cols)
15     columns = [f"Column_{i}" for i in range(1, n_cols + 1)]
16     return pd.DataFrame(data, index=dates, columns=columns)
17
18 # User input for the size of the dataset
19 n_rows = st.sidebar.slider("Number of rows", min_value=10,
20                             max_value=1000, value=100, step=10)
21 n_cols = st.sidebar.slider("Number of columns", min_value=1,
22                             max_value=20, value=5, step=1)
23
24 # Generate and display the dataframe
25 df = generate_data(n_rows, n_cols)
26 st.write("### Generated Data", df)
27
28 # Show column statistics
29 if st.sidebar.checkbox('Show Column Statistics'):
30     st.write("### Column Statistics", df.describe())
31
32 # Visualization
33 if st.sidebar.checkbox('Show Histogram'):
34     column_to_plot = st.sidebar.selectbox('Select Column to Visualize', df.columns)
35     fig, ax = plt.subplots()
36     df[column_to_plot].hist(bins=20, ax=ax)
37     ax.set_title(f'Histogram of {column_to_plot}')
38     st.pyplot(fig)
39
40 st.write("This app demonstrates Streamlit's capability for data science tasks using generated data.")
```

### テキスト分析アプリ

```
1 import streamlit as st
2
3 st.title('Text Analysis App')
4
5 # Text input
```

```

6 user_input = st.text_area("Enter your text here:")
7
8 # Analysis
9 if st.button('Analyze'):
10     char_count = len(user_input)
11     word_count = len(user_input.split())
12
13     st.write(f"Character Count: {char_count}")
14     st.write(f"Word Count: {word_count}")
15
16 st.write("This app counts the number of words and characters in
    your text.")

```

## ユーザーからフィードバックを受け付けるフォーム

```

1 import streamlit as st
2
3 st.title('User Feedback Form')
4
5 # User input fields
6 name = st.text_input('Name')
7 rating = st.select_slider('Rating', options=['Poor', 'Fair', 'Good',
    'Excellent'])
8 comments = st.text_area('Comments')
9
10 # Submit button
11 if st.button('Submit'):
12     st.success(f'Thank you {name}, for your feedback!')
13
14 st.write("This app collects user feedback.")

```

## 画像反転アプリ

```

1 import streamlit as st
2 from PIL import Image, ImageOps
3
4 st.title('Image Flipper')
5
6 # File uploader
7 uploaded_file = st.file_uploader("Choose an image...", type=['jpg',
    'jpeg', 'png'])
8 if uploaded_file is not None:
9     image = Image.open(uploaded_file)
10
11     # Display original image
12     st.image(image, caption='Uploaded Image', use_column_width=True)
13
14     # Options for flipping
15     flip_type = st.radio("Flip Type", ('Normal', 'Horizontal', '
    Vertical'))
16
17     if flip_type == 'Horizontal':
18         flipped_image = ImageOps.mirror(image)

```

```
19     elif flip_type == 'Vertical':
20         flipped_image = ImageOps.flip(image)
21     else:
22         flipped_image = image
23
24     # Display flipped image
25     st.image(flipped_image, caption='Flipped Image',
26             use_column_width=True)
27 st.write("This app flips the uploaded image based on the selected
28         option.")
```