

Interactive Presentation with Python Streamlit

Wei (Tony) Zhang, Pfizer (China) Research and Development Co.,Ltd.

ABSTRACT

In today's digital age, effective communication often requires more than just static presentations. Interactive presentations offer a dynamic way to engage and captivate audiences, fostering deeper understanding and collaboration. Python Streamlit is a powerful tool for creating interactive web applications with ease. Various features of Streamlit includes text inputs, sliders, buttons, and image embedding, to create interactive elements that respond to user input in real-time. Additionally, advanced techniques such as data visualization with interactive charts and integration with machine learning models, enables us to elevate the presentations with rich, dynamic content. Whether displaying data science dashboard, presenting research findings, showcasing product demos, or facilitating workshops, Python Streamlit empowers presenters to deliver immersive experiences that leave a lasting impression on their audience.

INTRODUCTION

Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science. It allows you to build interactive, data-driven applications with minimal code. Here's a breakdown of why Streamlit is an excellent choice for creating interactive presentations:

1. Simplicity and Ease of Use:

- Minimal Coding: Streamlit apps are straightforward to write. You can transform a Python script into an interactive web app without needing extensive knowledge of web development.
- Quick Prototyping: You can build and deploy a web app in minutes, making it ideal for quick prototyping and iterative development.

2. Interactivity:

- Widgets and Inputs: Streamlit provides a variety of widgets (sliders, text inputs, buttons, etc.) that make your presentations interactive and engaging.
- Real-Time Updates: Changes to the code are reflected in real-time, allowing for an interactive experience that can adapt to user input dynamically.

3. Integration with Data Science Tools:

- Seamless Integration: Streamlit works well with popular data science libraries like pandas, NumPy, Matplotlib, Plotly, and more. This allows you to leverage your existing data analysis skills to create rich visualizations.
- Machine Learning Integration: You can easily integrate machine learning models into your Streamlit app, providing interactive model predictions and insights.

4. Deployment and Sharing:

- **Easy Deployment:** Streamlit apps can be easily deployed and shared with others, whether it's within your team or with a broader audience. Streamlit provides hosting services that streamline the deployment process.
- **GitHub Integration:** You can deploy apps directly from a GitHub repository, making it easy to maintain and share your projects.

Creating an interactive presentation using Python and Streamlit can be an engaging way to present data, showcase applications, or provide a user-friendly interface for various projects. The main steps for using Streamlit are:

- 1. Install Streamlit**
- 2. Basic Streamlit App**
- 3. Run the Streamlit App**
- 4. Enhance Your Presentation**
- 5. Add More Interactivity**
- 6. Deploy Your App**

INSTALL STREAMLIT

Run your Streamlit app from the command line:

```
pip install streamlit
```

BASIC STREAMLIT APP

Create a simple Streamlit app to get familiar with its structure. Create a Python file, say app.py, and add the following code:

```
import streamlit as st

st.title('Interactive Presentation with Streamlit')
st.header('Introduction')
st.write('This is a simple interactive presentation using Streamlit.')

st.header('Data Visualization')
st.write('Here you can add some data visualization.')

# Example: Interactive widget
number = st.slider('Pick a number', 0, 100)
st.write(f'You selected: {number}')
```

RUN THE STREAMLIT APP

Run your Streamlit app from the command line:

```
streamlit run app.py
```

CONVERT AN EXISTING APP INTO A MULTIPAGE APP

We can convert the existing app to be a multipage app by the following steps:

1. Create a new pages folder in the same folder where the app.py lives
2. Rename app.py to any name.py, so that the title in the sidebar can be the same as the py file name.
3. Create three new files inside of pages folder for each page.
4. Run streamlit run the updated name py file to view your newly converted multipage app.

ENHANCE YOUR PRESENTATION

You can enhance your presentation by adding more features like charts, interactive widgets, and importing data. Here's an example:

st.write Write arguments to the app. <pre>st.write("Hello **world**!") st.write(my_data_frame) st.write(my_mpl_figure)</pre>	st.write_stream Write generators or streams to the app with a typewriter effect. <pre>st.write_stream(my_generator) st.write_stream(my_llm_stream)</pre>	Magic Any time Streamlit sees either a variable or literal value on its own line, it automatically writes that to your app using <code>st.write</code> <pre>"Hello **world**!" my_data_frame my_mpl_figure</pre>
Markdown Display string formatted as Markdown. <pre>st.markdown("Hello **world**!")</pre>	Title Display text in title formatting. <pre>st.title("The app title")</pre>	Header Display text in header formatting. <pre>st.header("This is a header")</pre>

Rank	Preview	App name	Author	Owner	URL	Views	GitHub	Views (past 30 days)
1		GPT2Zero	cladward	https://bit.ly/403k...	403k	0		
2		HuggingFace	napolita uach	https://bit.ly/403k...	79k	0		
3		KnowledgeGPT	mimo001	https://bit.ly/403k...	15k	883		
4		Tweet generator	kinosol	https://bit.ly/403k...	15k	81		
5		BERT Semantic	searchovd	https://bit.ly/403k...	12k	0		
6		GPT4E	stephenburger	https://bit.ly/403k...	7k	124		
7		GPT Lab Lounge	dcin	https://bit.ly/403k...	4k	0		

Dataframes

Display a dataframe as an interactive table.

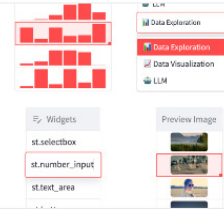
```
st.dataframe(my_data_frame)
```

Ty	Name	Avatar	Ty	Age	Ty	Gender	Ty	Is Active?	Ty	Homepage
	Gary Cross			76 years		None		☐		http://www.netra.com/
	Denise Thornton			33 years		None		☐		http://bit.ly/403k...
	Christina Saunders			24		None		☐		http://www.netra.com/
	Kathleen Collins			31 years		female		☐		https://www.rose-freeson.net/
	Timothy Parker			1 years		male		☐		https://thompson.com/
	Dr. Loreta Garcia DDS			69 years		female		☐		https://www.chart.com/
	John Gibbs			28 years		male		☐		http://www.simmons.org/

Data editor

Display a data editor widget.

```
edited = st.data_editor(df, num_rows="dynam
```



Column configuration

Configure the display and editing behavior of dataframes and data editors.

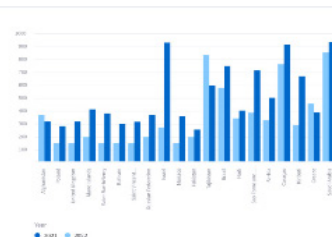
```
st.column_config.NumberColumn("Price (in US
```



Simple area charts

Display an area chart.

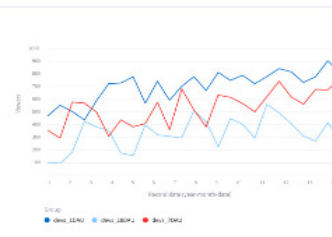
```
st.area_chart(my_data_frame)
```



Simple bar charts

Display a bar chart.

```
st.bar_chart(my_data_frame)
```



Simple line charts

Display a line chart.

```
st.line_chart(my_data_frame)
```

ADD MORE INTERACTIVITY

Streamlit supports a variety of interactive widgets such as sliders, text inputs, buttons, and more. You can also integrate with other libraries for more advanced features. Here's an example of using more widgets and features:

Button elements

Click me

Button

Display a button widget.

```
clicked = st.button("Click me")
```

Download file



Download button

Display a download button widget.

```
st.download_button("Download fi
```

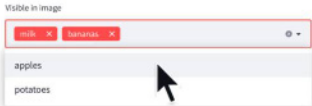
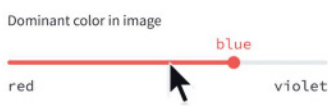
Submit

Form button

Display a form submit button. For use with `st.form`.

```
st.form_submit_button("Sign up"
```

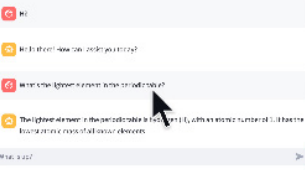
Selection elements

		
Checkbox Display a checkbox widget.	Toggle Display a toggle widget.	Radio Display a radio button widget.
<pre>selected = st.checkbox("I agree</pre>	<pre>activated = st.toggle("Activate</pre>	<pre>choice = st.radio("Pick one", [</pre>
		
Selectbox Display a select widget.	Multiselect Display a multiselect widget. The multiselect widget starts as empty.	Select slider Display a slider widget to select items from a list.
<pre>choice = st.selectbox("Pick one</pre>	<pre>choices = st.multiselect("Buy",</pre>	<pre>size = st.select_slider("Pick a</pre>

Numeric input elements

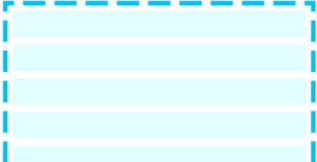
	
Number input Display a numeric input widget.	Slider Display a slider widget.
<pre>choice = st.number_input("Pick</pre>	<pre>number = st.slider("Pick a numb</pre>

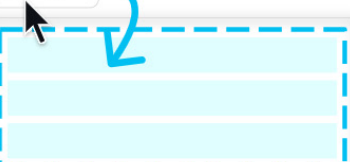
Text input elements

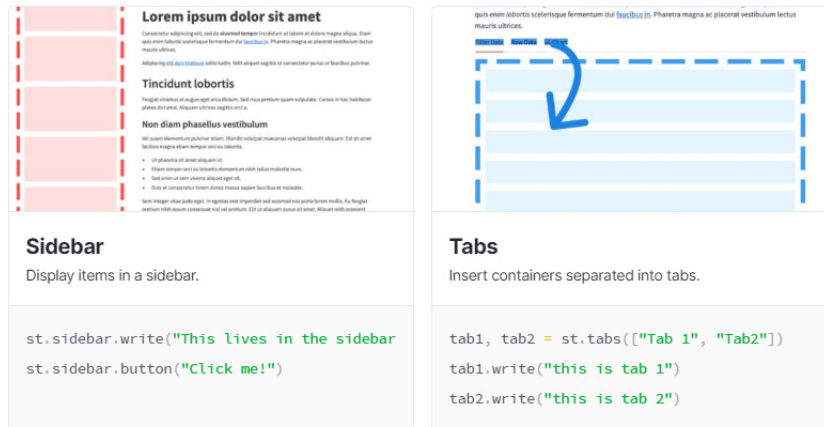
		
Text input Display a single-line text input widget.	Text area Display a multi-line text input widget.	Chat input Display a chat input widget.
<pre>name = st.text_input("First name")</pre>	<pre>text = st.text_area("Text to analyze")</pre>	<pre>prompt = st.chat_input("Say something") if prompt: st.write(f"The user has sent: {prompt}")</pre>

LAYOUTS AND CONTAINERS

Streamlit provides several options for controlling how different elements are laid out on the screen.

		
Columns Insert containers laid out as side-by-side columns.	Container Insert a multi-element container.	Modal dialogs Insert a modal dialog that can rerun independently from the rest of the script.
<pre>col1, col2 = st.columns(2) col1.write("this is column 1") col2.write("this is column 2")</pre>	<pre>c = st.container() c.write("This will show last") c.write("This will show first") c.write("This will show second")</pre>	<pre>@st.experimental_dialog("Sign up") def email_form(): name = st.text_input("Name") email = st.text_input("Email")</pre>

		
Empty Insert a single-element container.	Expander Insert a multi-element container that can be expanded/collapsed.	Popover Insert a multi-element popover container that can be opened/closed.
<pre>c = st.empty() c.write("This will show last") c.write("This will be replaced") c.write("This will show first")</pre>	<pre>with st.expander("Open to see more"): st.write("This is more content")</pre>	<pre>with st.popover("Settings"): st.checkbox("Show completed")</pre>



DEPLOY YOUR APP

Once you are happy with your app, you can deploy it using Streamlit sharing or other hosting services like Heroku, AWS, or Google Cloud. To deploy on Streamlit sharing, you need to:

- Push your code to a GitHub repository.
- Sign up on Streamlit sharing and link your GitHub repository.
- Follow the instructions to deploy your app

CONCLUSION

Streamlit is a powerful and versatile tool that revolutionizes the way we create and deliver interactive presentations. Its ability to transform Python scripts into engaging, interactive web applications makes it an invaluable resource for data scientists, educators, developers, and anyone looking to present data dynamically.

Interactive presentations with Streamlit offer a fresh approach to sharing data and insights. By combining the flexibility of Python with the interactivity of web applications, Streamlit enables you to create presentations that are not only informative but also engaging and interactive. Whether you are presenting at a conference, teaching a class, or sharing results with stakeholders, Streamlit provides the tools you need to make a lasting impact.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Wei (Tony) Zhang

Enterprise: Pfizer (China) Research and Development Co.,Ltd.

E-mail: wei.zhang18@pfizer.com