



Relax and Code on. Photo by Cody Black on Unsplash

How to write Web apps using simple

Get one more story in your member preview when you sign up. It's free.

 Sign up with Google

 Sign up with Facebook

Already have an account? [Sign in](#)

A Machine Learning project is never really complete if we don't have a good way to showcase it.

While in the past, a well-made visualization or a small PPT used to be enough for showcasing a data science project, with the advent of dashboarding tools like RShiny and Dash, a good data scientist needs to have a fair bit of knowledge of web frameworks to get along.

And Web frameworks are hard to learn. I still get confused in all that HTML, CSS, and Javascript with all the hit and trials, for something seemingly simple to do.

Not to mention the many ways to do the same thing, making it confusing for us data science folks for whom web development is a secondary skill.

So, are we doomed to learn web frameworks? Or to call our developer friend for silly doubts in the middle of the night?

This is where StreamLit comes in and delivers on its promise to create web apps just using Python.

Zen of Python: Simple is better than complex and Streamlit makes it dead simple to create apps.

This post is about understanding how to create apps that support data science projects using Streamlit.

To understand more about the architecture and the thought process that led to streamlit,



Get one more story in your member preview when you sign up. It's free.

Sign up with Google

Sign up with Facebook

Already have an account? [Sign in](#)

```
pip install streamlit
```

To see if our installation is successful, we can just run:

```
streamlit hello
```

This should show you a screen that says:

```
~/web/new_blog/post_files/streamlit ➔ streamlit hello

👋 Welcome to Streamlit!

If you are one of our development partners or are interested in
getting personal technical support, please enter your email address
below. Otherwise, you may leave the field blank.

Email:

Telemetry: As an open source project, we collect usage statistics.
We cannot see and do not store information contained in Streamlit apps.

If you'd like to opt out, add the following to ~/.streamlit/config.toml,
creating that file if necessary:

[browser]
gatherUsageStats = false

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://192.168.0.103:8501
```

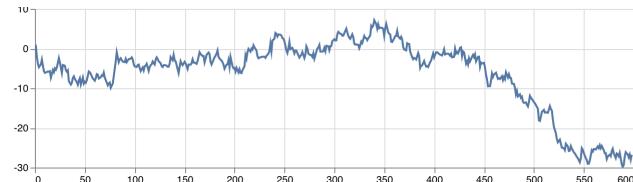
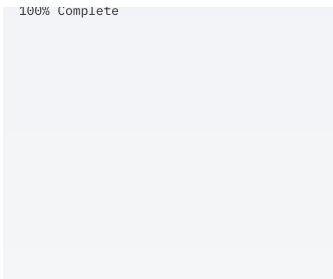


Get one more story in your member preview when you sign up. It's free.

 Sign up with Google

 Sign up with Facebook

Already have an account? [Sign in](#)



[Re-run](#)

— 0 ...

• • •

Streamlit Hello World

Streamlit aims to make app development easy using simple Python.

So let us write a simple app to see if it delivers on that promise.

Here I start with a simple app which we will call the Hello World of streamlit. Just paste the code given below in a file named `helloworld.py`

```
import streamlit as st

x = st.slider('x')
st.write(x, 'squared is', x * x)
```

And, on the terminal run:

```
streamlit run helloworld.py
```

X

Get one more story in your member preview when you sign up. It's free.

Sign up with Google

Sign up with Facebook

Already have an account? [Sign in](#)

A Simple slider widget app

It was pretty easy. In the above app, we used two features from Streamlit:

- the `st.slider` widget that we can slide to change the output of the web app.
- and the versatile `st.write` command. I am amazed at how it can write anything from charts, dataframes, and simple text. More on this later.

Important: Remember that every time we change the widget value, the whole app runs from top to bottom.

• • •

Streamlit Widgets

Widgets provide us a way to control our app. The best place to read about the widgets is the API reference documentation itself but I will describe some most prominent ones that you might end up using.

1. Slider

```
streamlit.slider(label, min_value=None, max_value=None, value=None, step=None, format=None)
```

We already saw `st.slider` in action above. It can be used with `min_value`, `max_value`, and `step` for getting inputs in a range.



Get one more story in your member preview when you sign up. It's free.

Sign up with Google

Sign up with Facebook

Already have an account? [Sign in](#)

This is how the app looks:

Enter URL

www.kaggle.com

The Entered URL is www.kaggle.com

A Simple text_input widget app

Tip: You can just change the file `helloworld.py` and refresh the browser. The way I work is to open and change `helloworld.py` in sublime text and see the changes in the browser side by side.

3. Checkbox

One use case for checkboxes is to hide or show/hide a specific section in an app. Another could be setting up a boolean value in the parameters for a function. `st.checkbox()` takes a single argument, which is the widget label. In this app, the checkbox is used to toggle a conditional statement.

```
import streamlit as st
import pandas as pd
import numpy as np

df = pd.read_csv("football_data.csv")
if st.checkbox('Show dataframe'):
    st.write(df)
```



Get one more story in your member preview when you sign up. It's free.

 Sign up with Google

 Sign up with Facebook

Already have an account? [Sign in](#)

6	177003	L. Modrić	32	https://cdn.sofifa.org...	Croatia	h
7	176580	L. Suárez	31	https://cdn.sofifa.org...	Uruguay	h
8	155862	Sergio Ramos	32	https://cdn.sofifa.org...	Spain	h
9	200389	J. Oblak	25	https://cdn.sofifa.org...	Slovenia	h
10	188545	R. Lewandowski	29	https://cdn.sofifa.org...	Poland	h

A Simple checkbox widget app

4. SelectBox

We can use `st.selectbox` to choose from a series or a list. Normally a use case is to use it as a simple dropdown to select values from a list.

```
import streamlit as st
import pandas as pd
import numpy as np

df = pd.read_csv("football_data.csv")

option = st.selectbox(
    'Which Club do you like best?',
    df['Club'].unique())

'You selected: ', option
```

Which Club do you like best?

Manchester United

You selected: Manchester United



Get one more story in your member preview when you sign up. It's free.

Sign up with Google

Sign up with Facebook

Already have an account? [Sign in](#)

```
import streamlit as st
import pandas as pd
import numpy as np

df = pd.read_csv("football_data.csv")

options = st.multiselect(
    'What are your favorite clubs?', df['Club'].unique())

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

What are your favorite clubs?



You selected:

```
▼ [
  0 : "FC Bayern München"
  1 : "Manchester United"
]
```

A Simple multiselect widget app

• • •

Creating Our Simple App Step by Step

So much for understanding the important widgets. Now, we are going to create a simple app using multiple widgets at once.

×

Get one more story in your member preview when you sign up. It's free.

G Sign up with Google

F Sign up with Facebook

Already have an account? [Sign in](#)

```

clubs = st.multiselect('Show Player for clubs?',
df['Club'].unique())

nationalities = st.multiselect('Show Player from Nationalities?',
df['Nationality'].unique())

# Filter dataframe
new_df = df[(df['Club'].isin(clubs)) &
(df['Nationality'].isin(nationalities))]

# write dataframe to screen
st.write(new_df)

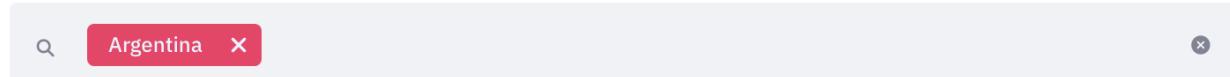
```

Our simple app looks like:

Show Player for clubs?



Show Player from Nationalities?



	0	ID	Name	Age	Photo	Nationality
23	23	153079	S. Agüero	30	https://cdn.sofifa.org...	Argentina https://
89	89	192366	N. Otamendi	30	https://cdn.sofifa.org...	Argentina https://
526	26	173373	S. Romero	31	https://cdn.sofifa.org...	Argentina https://
629	29	201862	M. Rojo	28	https://cdn.sofifa.org...	Argentina https://
15210	10	241632	B. Garré	17	https://cdn.sofifa.org...	Argentina https://

Using multiple widgets in conjunction

Get one more story in your member preview when you sign up. It's free.

Sign up with Google

Sign up with Facebook

Already have an account? [Sign in](#)

```

import streamlit as st
import pandas as pd
import numpy as np
import plotly_express as px

df = pd.read_csv("football_data.csv")

clubs = st.multiselect('Show Player for clubs?',
df['Club'].unique())
nationalities = st.multiselect('Show Player from Nationalities?',
df['Nationality'].unique())

new_df = df[(df['Club'].isin(clubs)) &
(df['Nationality'].isin(nationalities))]
st.write(new_df)

# create figure using plotly express
fig = px.scatter(new_df, x ='Overall',y='Age',color='Name')

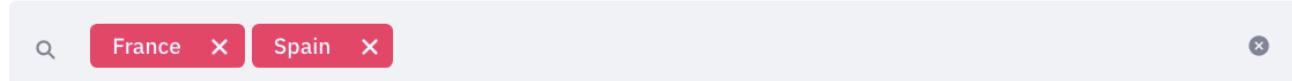
# Plot!
st.plotly_chart(fig)

```

Show Player for clubs?



Show Player from Nationalities?



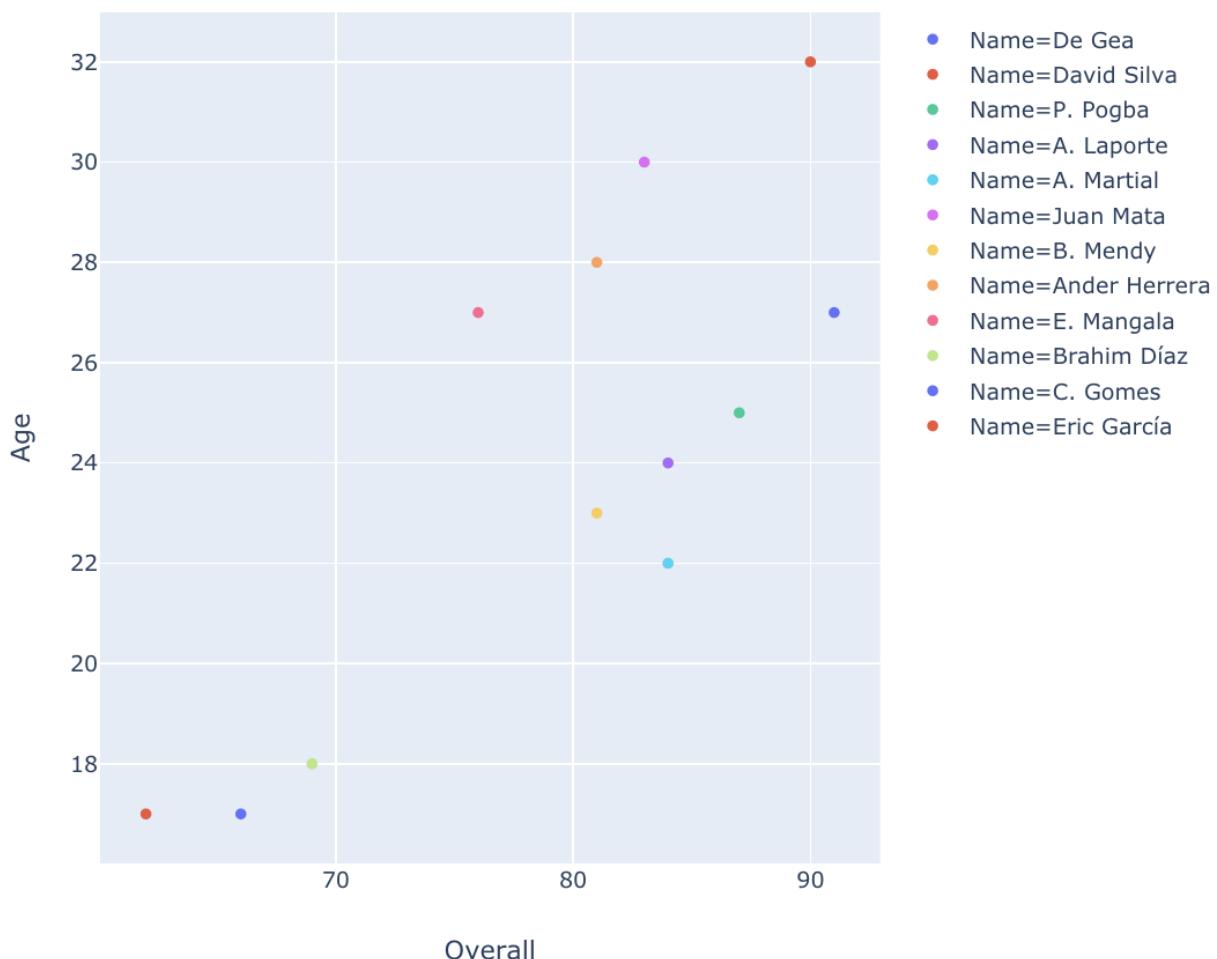
	Unnamed: 0	ID	Name	Age	Photo	National
3	3	193080	De Gea	27	https://cdn.sofifa.org...	S
13	13	168542	David Silva	32	https://cdn.sofifa.org...	S
45	45	195864	P. Pogba	25	https://cdn.sofifa.org...	Fr

Get one more story in your member preview when you sign up. It's free.

Sign up with Google

Sign up with Facebook

Already have an account? [Sign in](#)



Adding charts

• • •

Improvements

In the start we said that each time we change any widget, the whole app runs from start to end. This is not feasible when we create apps that will serve deep learning models or



Get one more story in your member preview when you sign up. It's free.

Sign up with Google

Sign up with Facebook

Already have an account? [Sign in](#)

```
import streamlit as st
import pandas as pd
import numpy as np
import plotly_express as px

df = st.cache(pd.read_csv)("football_data.csv")
```

Or for more complex and time taking functions that need to run only once(think loading big Deep Learning models), using:

```
@st.cache
def complex_func(a,b):
    DO SOMETHING COMPLEX

# Won't run again and again.
complex_func(a,b)
```

When we mark a function with Streamlit's cache decorator, whenever the function is called streamlit checks the input parameters that you called the function with.

If this is the first time Streamlit has seen these params, it runs the function and stores the result in a local cache.

When the function is called the next time, if those params have not changed, Streamlit knows it can skip executing the function altogether. It just uses the results from the cache.

2. Sidebar

Get one more story in your member preview when you sign up. It's free.

Sign up with Google

Sign up with Facebook

Already have an account? [Sign in](#)

```
clubs = st.sidebar.multiselect('Show Player for clubs?',  
df['Club'].unique())  
nationalities = st.sidebar.multiselect('Show Player from  
Nationalities?', df['Nationality'].unique())  
  
new_df = df[(df['Club'].isin(clubs)) &  
(df['Nationality'].isin(nationalities))]  
st.write(new_df)  
  
# Create distplot with custom bin_size  
fig = px.scatter(new_df, x ='Overall',y='Age',color='Name')  
  
# Plot!  
st.plotly_chart(fig)
```



X

Get one more story in your member preview when you sign up. It's free.

 Sign up with Google

 Sign up with Facebook

Already have an account? [Sign in](#)

comments. You could also have used the command `st.markdown`

```
import streamlit as st
import pandas as pd
import numpy as np
import plotly_express as px

...
# Club and Nationality App

This very simple webapp allows you to select and visualize players
from certain clubs and certain nationalities.
...
df = st.cache(pd.read_csv)("football_data.csv")

clubs = st.sidebar.multiselect('Show Player for clubs?',
                               df['Club'].unique())
nationalities = st.sidebar.multiselect('Show Player from
Nationalities?', df['Nationality'].unique())

new_df = df[(df['Club'].isin(clubs)) &
            (df['Nationality'].isin(nationalities))]
st.write(new_df)

# Create distplot with custom bin_size
fig = px.scatter(new_df, x ='Overall',y='Age',color='Name')

...
### Here is a simple chart between player age and overall
...

st.plotly_chart(fig)
```



Get one more story in your member preview when you sign up. It's free.

 Sign up with Google

 Sign up with Facebook

Already have an account? [Sign in](#)

Our final App Demo

• • •

Conclusion

Streamlit has democratized the whole process to create apps, and I couldn't recommend it more.

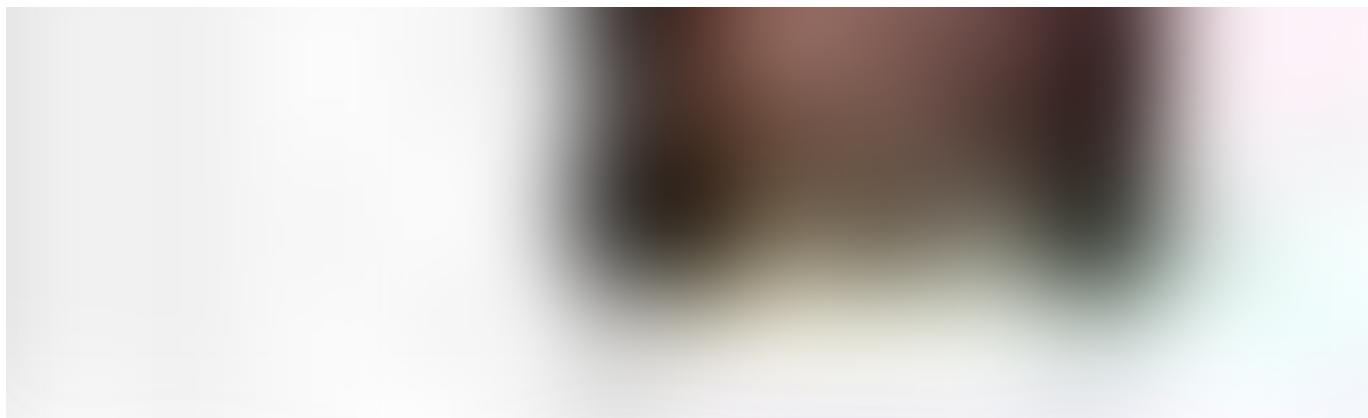
In this post, we created a simple web app. But the possibilities are endless. To give an example here is face GAN from the streamlit site. And it works by just using the same

Get one more story in your member preview when you sign up. It's free.

 Sign up with Google

 Sign up with Facebook

Already have an account? [Sign in](#)



I love the default colors and styles that the developers have used, and I found it much more comfortable than using Dash, which I was using until now for my demos. You can also include audio and video in your streamlit apps.

On top of that, Streamlit is a free and open-source rather than a proprietary web app that just works out of the box.

In the past, I had to reach out to my developer friends for any single change in a demo or presentation; now it is relatively trivial to do that.

I aim to use it more in my workflow from now on, and considering the capabilities it provides without all the hard work, I think you should too.

I don't have an idea if it will perform well in a production environment yet, but its a boon for the small proof of concept projects and demos. I aim to use it more in my workflow from now on, and considering the capabilities it provides without all the hard work, I

X

Get one more story in your member preview when you sign up. It's free.

 Sign up with Google

 Sign up with Facebook

Already have an account? [Sign in](#)

University of Michigan, which is a part of a pretty good **Data Science Specialization with Python** in itself. Do check it out.

• • •

Thanks for the read. I am going to be writing more beginner-friendly posts in the future too. Follow me up at **Medium** or **Subscribe to my blog** to be informed about them. As always, I welcome feedback and constructive criticism and can be reached on Twitter @mlwhiz.

Also, a small disclaimer — There might be some affiliate links in this post to relevant resources, as sharing knowledge is never a bad idea.

Python Data Science Programming Artificial Intelligence Machine Learning

About Help Legal



Get one more story in your member preview when you sign up. It's free.

 Sign up with Google

 Sign up with Facebook

Already have an account? [Sign in](#)