

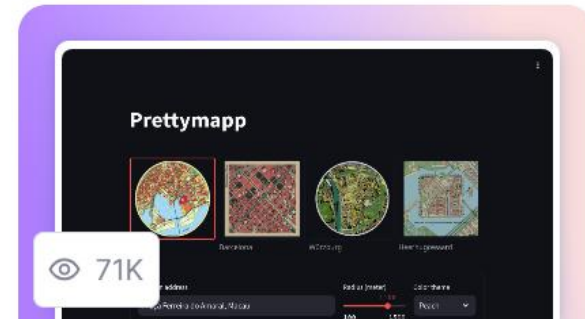
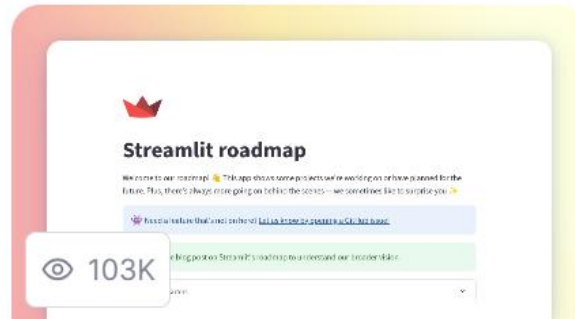
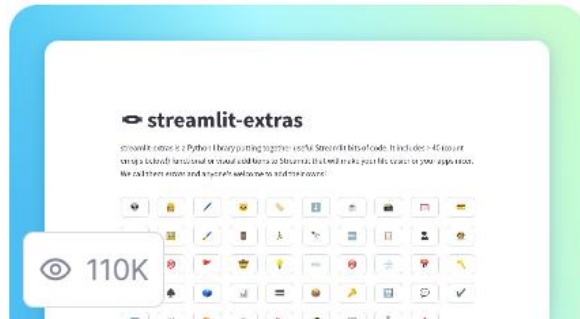
13

Interaktive Apps mit Streamlit

Streamlit ist eine Python-Bibliothek, die den Bau von Dashboards und anderen Programmen für Datenanalyse und Data Science ermöglicht.

Das Schreiben von Apps in Streamlit geht ohne OOP-Kenntnisse, da Streamlit leicht nutzbare Funktionen beim Import zur Verfügung stellt.

Auch bei Streamlit gibt es viele tolle Beispiele in einer App Gallery.



Eine erste App ist einfach zu erzeugen. Dazu legt man eine py-Datei und importiert Streamlit, ab dann werden über st die nötigen Funktionen genutzt:

```
import streamlit as st

st.title('Hello, Streamlit World!')
st.text('Das ist ein Text mit st.text')
...
```

Das Programm (hier app.py) wird über das Terminal gestartet:

```
streamlit run app.py
```

Die App ist standardmäßig unter Port 8051 erreichbar und kann im Terminal mit Strg + C beendet werden.

Im Gegensatz zu Dash, gibt es keine Layout-Methode. Die App ergibt sich Zeile für Zeile aus den einzelnen aufgelisteten Funktionen.

Hier eine App mit Plotly-Grafik und eingebundenem Dataframe:

```
import streamlit as st
import pandas as pd

customers = pd.read_csv('Customers.csv', index_col=0)
st.title('Kundenanalyse-App')

scores_by_profession = (customers.groupby('Profession')
                        ['Spending Score (1-100)'].mean()
                        .sort_values(ascending=False))

score_profession_barchart = px.histogram(scores_by_profession,
                                          scores_by_profession.index,
                                          'Spending Score (1-100)')

st.plotly_chart(score_profession_barchart)
st.dataframe(customers)
```

Bei jeder Nutzerinteraktion führt Streamlit die py-Datei neu aus. Um ständiges Laden zu vermeiden, werden sogenannte Forms eingerichtet, bei denen Nutzereingaben gebündelt auf Knopfdruck übergeben werden.

```
categorical_cols = ['Gender', 'Profession', 'Work Experience', 'Family Size']
```

```
with st.form('analysis-form'):  
    cat_selection = st.selectbox("Welche Kategorie interessiert dich?",  
                                categorical_cols)
```

```
target_selection = st.radio(  
    'Von was willst du die Mittelwerte sehen?',  
    ['Annual Income ($)', 'Spending Score (1-100)', 'Age']  
)
```

```
st.form_submit_button('Diagramm erstellen', type='primary')
```

Da das Laden von Daten, Berechnungen sowie Modellbildungen die Arbeit der App verlangsamen, gibt es mit Caching einen Mechanismus, um einmal generierte Daten und Ressourcen zu speichern und schnell zu laden.

```
@st.cache_data
def load_customers():
    # Mit sleep, damit der Effekt von Caching spürbar wird:
    time.sleep(5)
    return pd.read_csv('Customers.csv', index_col=0)
```

```
customers = load_customers()
```

Caching von Machine-Learning-Modellen erfolgt mit `st.cache_resource`.

Eine Session ist eine Streamlit-App, die in einem Browsertab läuft. Bei jeder Interaktion außerhalb von Formen läuft das ganze Skript komplett ab. Session State ermöglicht es, Variablen von einer Ausführung zur nächsten weiterzureichen, indem diese in einem Dictionary gespeichert werden.

```
# Zugriff auf session_state per Dictionary-Notation:
```

```
if 'key' not in st.session_state:  
    st.session_state['key'] = 'value'
```

```
# Zugriff auf session_state mit Attribut-Schreibweise:
```

```
if 'key' not in st.session_state:  
    st.session_state.key = 'value'
```

Um neue Werte von Widgets in `st.session_state` zu speichern, ordnet man diesen einen Key zu. Damit bei Veränderungen auch etwas geschieht (z.B. eine Berechnung), setzt man *callback functions* auf `on_change` oder `on_click`.

```
kwh_consumed = st.number_input(
    'Vorjahresverbrauch kWh',
    min_value=0.0,
    value=10_000.0,
    key='kwh_consumed',
    on_change=calculate_cost
)
```

Callback functions stellen dabei ganz gewöhnliche Python-Funktionen dar, die vor dem Rest des Skripts ablaufen:

```
def calculate_cost():
    return round(st.session_state['kwh_consumed'] *
                st.session_state['preceding_kwh_price'],
                2)
```