

# Streamlit



# Streamlit

<https://docs.streamlit.io/>

# Inhaltsverzeichnis

- / Text Elemente
- / Caching
- / Buttons & Slider
- / Sidebar
- / State
- / Maps
- / Placeholders
- / Containers
- / Columns

# Intallation von Streamlit

- / Öffne das Terminal
- / Python –version
- / pip install streamlit

# Starten von Streamlit

/ Datei erzeugen: (bspw. streamlitexample.py)

/ Code:

```
import streamlit as st  
st.title('hello world')
```

/ Zum Pfad wechseln, wo das File ist

/ Im Terminal (cmd)

/ Streamlit run main.py

/ Ctrl + C -> stop run

→ / **Streamlit hello** -> demo

# Text & Daten

## Sales Report

### Q1 Result

January was the start of the year

```
{
  "January" : 100
  "February" : 110
  "March" : 115
}
```

### Q2 Results

Q2 had better results 😊

```
{
  "April" : 150
  "May" : 200
  "June" : 250
}
```

	Month	Amount
0	April	150
1	May	200
2	June	250

	Month	Amount
0	April	150
1	May	200
2	June	250

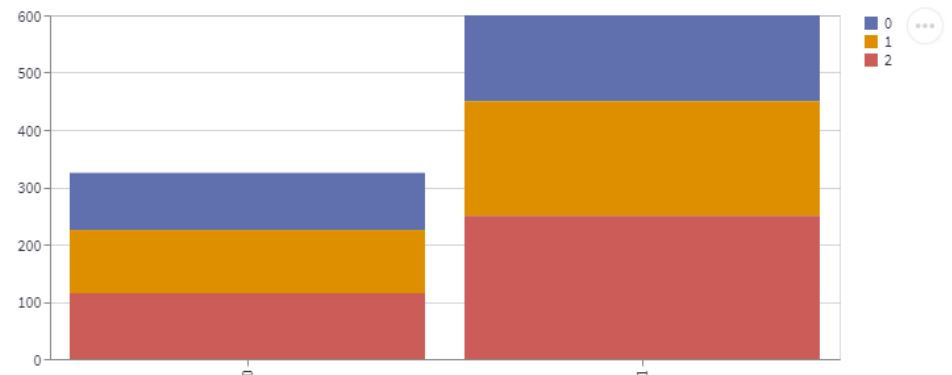
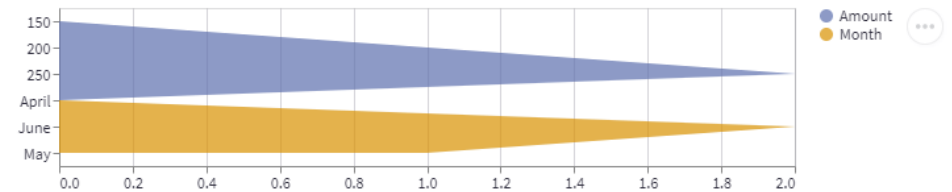
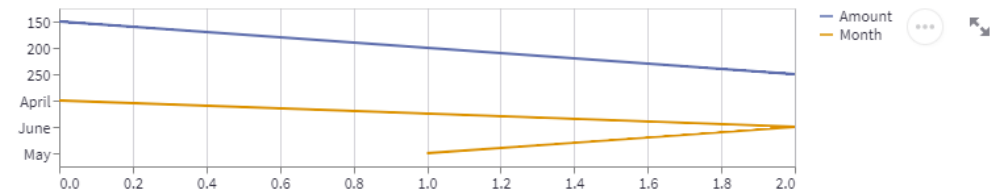
```
1 import streamlit as st
2 import pandas as pd
3
4 st.title('Sales Report')
5
6 st.header('Q1 Result')
7
8 q1_sales = {
9     'January': 100,
10    'February': 110,
11    'March': 115
12 }
13
14 st.write('January was the start of the year')
15 st.write(q1_sales)
16 st.header('Q2 Results')
17 q2_sales = {
18     'April': 150,
19     'May': 200,
20     'June': 250
21 }
22
23 'Q2 had better results:smile:#streamlit even has emojis like :smile:'
24 q2_sales
25
26 q2_df = pd.DataFrame(q2_sales.items(),
27                      columns=['Month', 'Amount']
28                      )
29 st.table(q2_df)
30
31 st.dataframe(q2_df)
```

# Diagramm & Bilder

```

1  import streamlit as st
2  import pandas as pd
3
4  q1_sales = {
5      'January': 100,
6      'February': 110,
7      'March': 115
8  }
9
10 q2_sales = {
11     'April': 150,
12     'May': 200,
13     'June': 250
14 }
15
16 q2_df = pd.DataFrame(q2_sales.items(),
17                      columns=['Month', 'Amount'])
18
19
20 q2_df2=q2_df.astype(str)
21
22 st.line_chart(q2_df2)
23
24 st.area_chart(q2_df2)
25
26 st.bar_chart([q1_sales.values(), q2_sales.values()])
27
28 from PIL import Image
29 st.image(image=Image.open('santa.png'),
30         caption='santa'
31         )
32

```



# Button & Slider

Please select a range of values



(40.0, 80.0)

Choose quarters

Choose an option

▶ []

Which quarter

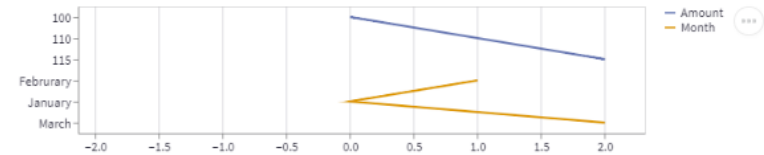
1

1

Show Q2 Data

	Month	Amount
0	January	100
1	February	110
2	March	115

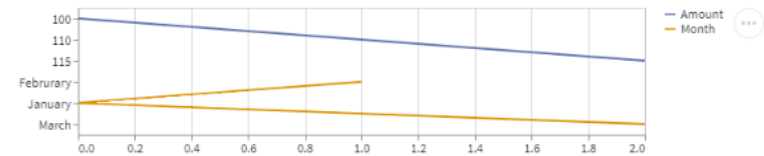
☐ Show Q2 Data



Which quarter?

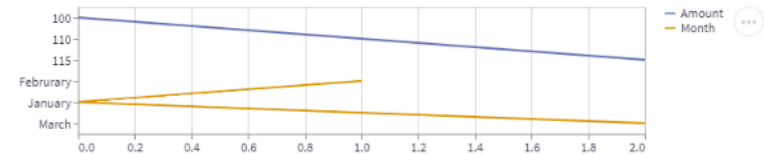
☒ Q1

☐ Q2



Which quarter?

Q1



Which quarters?



2

Please select a range of values



# Button & Slider

```

1 import streamlit as st
2 import pandas as pd
3
4 q1_sales = {
5     'January': 100,
6     'February': 110,
7     'March': 115
8 }
9
10 q2_sales = {
11     'April': 150,
12     'May': 200,
13     'June': 250
14 }
15
16 q2_df = pd.DataFrame(q2_sales.items(),
17                     columns=['Month', 'Amount'])
18
19 q1_df = pd.DataFrame(q1_sales.items(),
20                     columns=['Month', 'Amount'])
21
22 q2_df2 = q2_df.astype(str)
23 q1_df2 = q1_df.astype(str)
24
25 if st.button('Show Q2 Data'):
26     st.table(q2_df)
27 else:
28     st.table(q1_df)
29
30 if st.checkbox('Show Q2 Data'):
31     st.line_chart(q2_df2)

```

```

34 else:
35     st.line_chart(q1_df2)
36
37 quarter = st.radio('Which quarter?', ('Q1', 'Q2'))
38
39 if quarter == 'Q1':
40     st.line_chart(q1_df2)
41 elif quarter == 'Q2':
42     st.line_chart(q2_df2)
43
44 select_quarters = st.selectbox('Which quarter?', ('Q1', 'Q2'))
45
46 if select_quarters == 'Q1':
47     st.line_chart(q1_df2)
48 elif select_quarters == 'Q2':
49     st.line_chart(q2_df2)
50
51 st.write(st.slider('Which quarters?', 1, 4, (2)))
52
53 values = st.slider('Please select a range of values', 0.0, 100.0, (40.0, 80.0))
54
55 st.write(values)
56
57 st.write(st.multiselect('Choose quarters',
58                        ['Q1', 'Q2', 'Q3', 'Q4']))
59
60 st.write(st.number_input('Which quarter', 1, 4))

```

# Caching

Latency

Which quarter?

1

Sales amount 100

```
1 import streamlit as st
2 import time
3
4 @st.cache #add anotation at the end in order to show the 3sec delay @st.cache(max_entries=2, ttl=
5 def download_data(quarter):
6     sales = [100, 200, 300, 400]
7     time.sleep(3) # to simulate latency add a sleep
8     return sales[int(quarter - 1)]
9
10 st.write('Latency')
11
12 quarter = st.number_input('Which quarter?', 1, 4)
13
14 sales_amount = download_data(quarter)
15
16 st.write('Sales amount', sales_amount)
17
```

# Sidebars

×

Which section?

☒ Text

☐ Charts

☐ Widgets

## Sidebar

Lorem ipsum

```
1 import streamlit as st
2
3 st.title('Sideban')
4
5 button = 1
6
7 section = st.sidebar.radio('Which section?',
8                             ('Text', 'Charts', 'Widgets'))
9
10 q1_sales = {
11     'January': 100,
12     'February': 110,
13     'March': 115
14 }
15
16 if section == 'Text':
17     st.write('Lorem ipsum')
18 elif section == 'Charts':
19     st.write(q1_sales)
20 elif section == 'Widgets':
21     st.write(st.slider('Simple Slider', 1, 4, 0))
```

# Exkurs Projektarbeit

## Projektarbeit

Gruppeneinteilungen  
Beschreibung des Auftrages

# Auftrag

- / Eure Aufgabe ist es, ein interaktives Web-Dashboard zur Gebäudeautomatisierung zu entwickeln. Das Dashboard soll es Benutzern ermöglichen, etwas bestimmtes in einem Gebäude zu steuern
- / Ihr setzt das ganze erlernte Wissen ein

# Aufgabe 1 – 30min

- / - Erstelle ein UI mit 4 Seiten (Navigations Bar)
- / - Erstelle für jede Seite einen Titel
- / Seite 1
  - / Erstelle eine Klassenliste mit Vor und Nachnamen.
  - / Stelle die Liste auf der Webseite dar.
- / Seite 2
  - / Erstelle einen Slider der den aktuellen Wert ausgibt.
- / Seite 3
  - / Erstelle eine Liste mit 10 Zahlenwerten.
  - / Benutze ein Chart (bsp line.chart) um die Werte darzustellen.
- / Seite 4
  - / Erstelle einen Butten mit dem man eine Variabel hochzählt.

# State

```

1  import streamlit as st
2  import os
3  import json
4
5  if not os.path.isfile('2.WebApp.json'):
6      with open('2.WebApp.json', 'w') as f:
7          json.dump({'clicks': 0}, f)
8
9  with open('2.WebApp.json') as f:
10     counter = json.load(f)['clicks']
11
12     if st.button('Click me with 2.WebApp'):
13         counter += 1
14
15     st.write(f'The button was clicked {counter} times')
16
17     with open('2.WebApp.json', 'w') as f:
18         json.dump({'clicks': counter}, f)
19

```

Click me with 2.WebApp

The button was clicked 6 times

Erstelle ein .json  
File in dem  
gleichen Ordner

2.WebApp.json

```
{ "clicks": 6 }
```

## Aufgabe 2 - 15min

/ - Erstelle auf Seite 4 einen Button mit Counter



# Maps

	lat	lon
0	46.9479	7.4446
1	47.3686	8.5392



```

1 import streamlit as st
2 import pandas as pd
3
4 coordinates = pd.DataFrame({
5     'lat': [46.947922, 47.3686498],
6     'lon': [7.444608, 8.5391825],
7 })
8
9 st.write(coordinates)
10
11 st.map(coordinates)
12
13 import pydeck as pdk
14
15 @st.cache
16 def from_data_file(filename):
17     url = ("https://raw.githubusercontent.com/streamlit/example-data/master/hello/v1/%s" % filename)
18     return pd.read_json(url)
19
20 st.write(from_data_file("bart_stop_stats.json"))
21 layer = pdk.Layer(
22     "TextLayer",
23     data=from_data_file("bart_stop_stats.json"),
24     get_position=["lon", "lat"],
25     get_text="name",
26     get_color=[0, 0, 0, 200], #310, 0,0 200 should be red
27     get_size=15, # increase text size
28     get_aligment_baseline="bottom",
29 )
30
31 st.pydeck_chart(pdk.Deck(
32     map_style="mapbox://styles/mapbox/light-v9",
33     initial_view_state={"latitude": 37.76,
34                        "longitude": -122.4,
35                        "zoom": 11, "pitch": 50},
36     layers=[layer],
37 ))

```

# Placeholders

```

1  import streamlit as st
2      from time import sleep
3      from random import randint
4
5  def read_sensor():
6      sleep(0.2)
7      return randint(1, 10)
8
9  for i in range(20):
10     sensor = read_sensor()
11     st.write(f'Sensor data: {sensor}')
12
13     place = st.empty()
14
15     for i in range(20):
16         sensor = read_sensor()
17         place.write(f'Sensor data: {sensor}')
18
19     place2 = st.empty()
20     data = []
21     for i in range(20):
22         sensor = read_sensor()
23         data.append(sensor)
24         place2.line_chart(data)
25
26     place2.write('Done')

```

Sensor data: 8

Sensor data: 7

Sensor data: 3

Sensor data: 10

Sensor data: 9

Sensor data: 9

Sensor data: 6

Sensor data: 8

Sensor data: 6

Sensor data: 2

Sensor data: 2

Sensor data: 7

Sensor data: 6

Sensor data: 1

Sensor data: 5

Sensor data: 9

Sensor data: 6

Sensor data: 1

Sensor data: 6

Sensor data: 1

Sensor data: 4

Done

## Aufgabe 3 - 40min

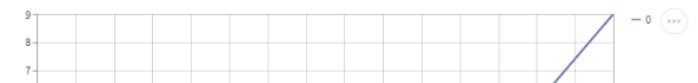
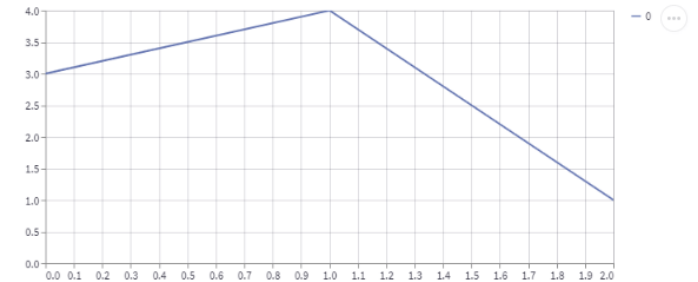
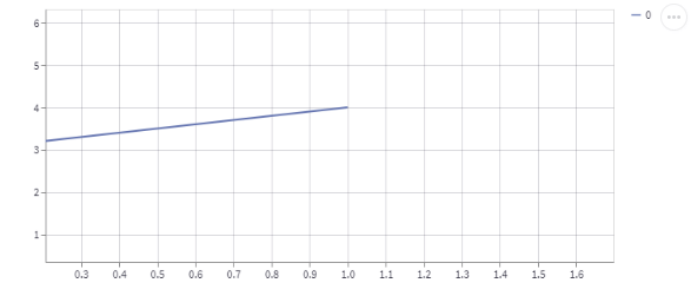
- / - Erstelle eine weitere Seite in der Navigations Bar (Seite 5).
- / - Erstelle einen Placeholder.
- / - Stelle die Namen der Klassenkameraden in einem 2sec abstand in dem Placeholder dar.

# Container

```

1 import streamlit as st
2 from time import sleep
3 from random import randint
4
5 def read_sensor():
6     sleep(0.2)
7     return randint(1, 10)
8
9 place = st.container()
10 st.write('Some more content')
11
12 data = []
13 for i in range(20):
14     sensor = read_sensor()
15     data.append(sensor)
16     place.line_chart(data)
17
18 place.write('Done')
19

```



# Columns Beispiel 1

/

```
1 import streamlit as st
2 col1, col2, col3 = st.columns(3)
3
4 with col1:
5     st.header("A cat")
6     st.image("https://static.streamlit.io/examples/cat.jpg")
7
8 with col2:
9     st.header("A dog")
10    st.image("https://static.streamlit.io/examples/dog.jpg")
11
12 with col3:
13     st.header("An owl")
14     st.image("https://static.streamlit.io/examples/owl.jpg")
```

A cat



A dog



An owl



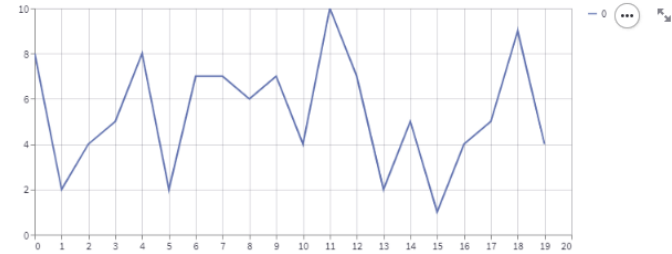
<https://docs.streamlit.io/library/api-reference/layout/st.columns>

# Columns Beispiel 2

```

1 import streamlit as st
2 from time import sleep
3 from random import randint
4
5 def read_sensor():
6     sleep(0.2)
7     return randint(1, 10)
8
9 place = st.container()
10 st.write('Some more content')
11
12 left_column, right_column = st.columns(2)
13
14 data = []
15
16 for i in range(20):
17     sensor = read_sensor()
18     data.append(sensor)
19
20 place.line_chart(data)
21 place.write('Done')
22
23 left_column.write('Data:')
24 left_column.write(data)
25
26 with right_column:
27     st.write('Chart:')
28     st.area_chart(data)

```



Done

Some more content

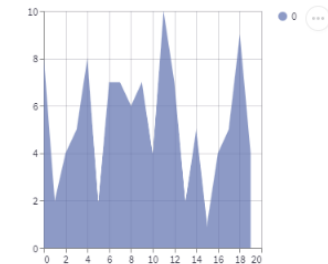
Data:

```

[
  0 : 8
  1 : 2
  2 : 4
  3 : 5
  4 : 8
  5 : 2
  6 : 7
  7 : 7
  8 : 6
  9 : 7
 10 : 4
 11 : 10
 12 : 7
 13 : 2
 14 : 5
 15 : 1
 16 : 4
 17 : 5
 18 : 9
 19 : 4
]

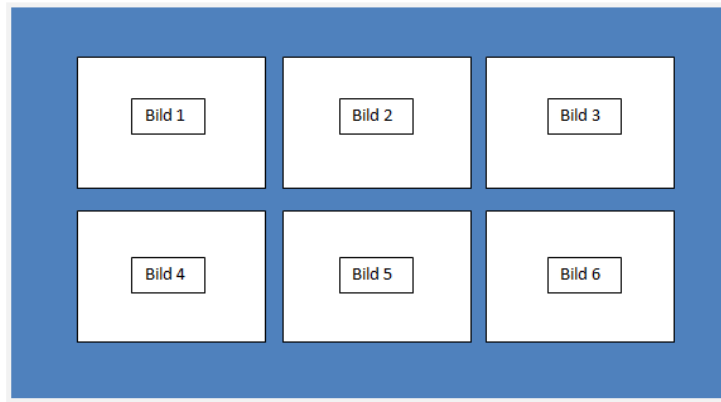
```

Chart:



## Aufgabe 4 - 30min

- / - Wähle 6 Bilder
- / - Versuche folgendes Layout nachzubauen



/ Zusatzaufgabe:

/ Lass das Bild 1 alle 2 sekunden durch ein anderes Bild ersetzen

<https://blog.jcharistech.com/2020/10/10/how-to-add-layout-to-streamlit-apps/>

## Aufgabe 6 - 40min

- / - Erstelle Seite 7 in der Navigations Bar
- / - Erstelle einen Placeholder
- / - Der Placeholder soll mit dem aktuellen Wert von S8 befüllt werden.



## Aufgabe 7 – 30 min

/ Aufgabenstellung:

/ Erstellen Sie eine Benutzeroberfläche, welches einen Regler besitzt, um die Meilen auszuwählen, für die man gerne die kmh Wert auslesen möchte. Für das Zufallszahlenspiel, sollte der Benutzer eine Eingabe machen, danach auf einen Button klicken (Spielen) und ein Ergebnis kriegen.