BINUS UNIVERSITY

People
Innovation
Excellence

Web Games using Python & Streamlit for Student

Dr Alexander A S Gunawan
aagung@binus.edu
08175001010

Expert Profile

Dr. Ir. Alexander A S Gunawan, S.Si, M.T, M.Sc, IPM

- Ir – UGM (2019)
- Dr – UI (2013-2016) Computer Science
- MT – ITB (2000-2003) Electrical Engineering
- MSc – Fachhochschule Darmstadt (2001-2002) Robotics
- S.Si – ITB (1996-2000) Mathematics

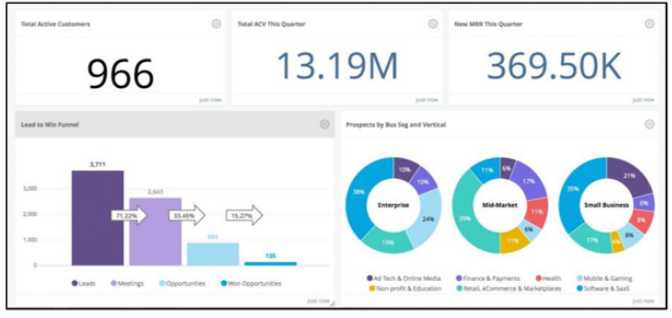
Data Scientist

Chairman ICCSCI 2018

Research Grants:
- **Ristek**
- **PKPT, PUPT**
- **Konsorsium etc**

Python

Streamlit



Streamlit

https://streamlit.io/

- Turn Data Scripts into Web Apps
- Interactive
- No Frontend Experience Required
- Easy to Deploy

Real time object detection

An image browser for the Udacity self-driving-car dataset with real-time object detection.

See on GitHub

Skills

	Flask	django	Streamlit
Python	✓	✓	✓
HTML	✓	✓	✗
CSS	✓	✓	✗
JavaScript	✓	✓	✗

Flexible

			
Learn	Moderate	Difficult	Easy
Simplicity	Moderate	Complex	Simple
Flexible	Moderate	High	Less
Scalable	Yes	Yes	Yes
Customize	Easy	Easy	Difficult
Deployment	Depends on Application	Depends on Application	Fast

Visualizations

			
Dashboard	Difficult	Difficult	Easy
Matplotlib, Seaborn	Difficult	Difficult	Easy
Plotly, Bokeh	Complicated	Complicated	Easy

Altair
Pydeck
Graphviz
Maps
Folium

Tutorial

- https://docs.streamlit.io/
- https://www.datacamp.com/tutorial/streamlit
- https://towardsdatascience.com/streamlit-hands-on-from-zero-to-your-first-awesome-web-app-2c28f9f4e214

Streamlit cheat sheet

streamlit.io

This cheat sheet is a summary of the docs

I also recommend streamlitopedia

How to install and import

```
$ pip install streamlit
```

Import convention
```
>>> import streamlit as st
```

Add widgets to sidebar

```
st.sidebar.<widget>
```

```
>>> my_val = st.sidebar.text_input('I:')
```

Command line

```
$ streamlit --help
$ streamlit run your_script.py
$ streamlit hello
$ streamlit config show
$ streamlit cache clear
$ streamlit docs
$ streamlit --version
```

Pre-release features

To access beta and experimental features

```
pip uninstall streamlit
pip install streamlit-nightly --upgrade
```

Magic commands

Magic commands allow you to implicitly `st.write()`

```
 ''' _This_ is some __Markdown__ '''

a=3
'a', a

'dataframe:', data
```

Display text

```
st.text('Fixed width text')
st.markdown('_Markdown_') # see *
st.latex(r''' e^{i\pi} + 1 = 0 ''')
st.write('Most objects') # df, err, func, keras!
st.write(['st', 'is <', 3]) # see *
st.title('My title')
st.header('My header')
st.subheader('My sub')
st.code('for i in range(8): foo()')
```

* optional kwarg unsafe_allow_html = True

Display data

```
st.dataframe(data)
st.table(data.iloc[0:10])
st.json({'foo':'bar','fu':'ba'})
```

Display charts

```
st.line_chart(data)
st.area_chart(data)
st.bar_chart(data)
st.pyplot(fig)
st.altair_chart(data)
st.vega_lite_chart(data)
st.plotly_chart(data)
st.bokeh_chart(data)
st.pydeck_chart(data)
st.deck_gl_chart(data)
st.graphviz_chart(data)
st.map(data)
```

Display media

```
st.image('./header.png')
st.audio(data)
st.video(data)
```

Display interactive widgets

```
st.button('Hit me')
st.checkbox('Check me out')
st.radio('Radio', [1,2,3])
st.selectbox('Select', [1,2,3])
st.multiselect('Multiselect', [1,2,3])
st.slider('Slide me', min_value=0, max_value=10)
st.text_input('Enter some text')
st.number_input('Enter a number')
st.text_area('Area for textual entry')
st.date_input('Date input')
st.time_input('Time entry')
st.file_uploader('File uploader')
st.beta_color_picker('Pick a color')
```

Use widgets' returned values in variables:

```
>>> for i in range(int(st.number_input('Num:'))): foo()
>>> if st.sidebar.selectbox('I:',['f']) == 'f': b()
>>> my_slider_val = st.slider('Quinn Mallory', 1, 88)
>>> st.write(slider_val)
```

Control flow

```
st.stop()
```

Display code

```
st.echo()

>>> with st.echo():
>>>     # Code below both executed and printed
>>>     foo = 'bar'
>>>     st.write(foo)
```

Display progress and status

```
st.progress(progress__variable_1_to_100)

st.spinner()

>>> with st.spinner(text='In progress'):
>>>     time.sleep(5)
>>>     st.success('Done')

st.balloons()
st.error('Error message')
st.warning('Warning message')
st.info('Info message')
st.success('Success message')
st.exception(e)
```

Placeholders, help, and options

```
st.empty()

>>> my_placeholder = st.empty()
>>> my_placeholder.text('Replaced!')

st.help(pandas.DataFrame)

st.get_option(key)
st.set_option(key)

st.beta_set_page_config(layout='wide')
```

Mutate data

```
DeltaGenerator.add_rows(data)

>>> my_table = st.table(df1)
>>> my_table.add_rows(df2)

>>> my_chart = st.line_chart(df1)
>>> my_chart.add_rows(df2)
```

Optimize performance

```
@st.cache

>>> @st.cache
... def foo(bar):
...     # Mutate bar
...     return data
...
>>> d1 = foo(ref1)
>>> # Executes as first time
>>>
>>> d2 = foo(ref1)
>>> # Does not execute; returns cached value, d1==d2
>>>
>>> d3 = foo(ref2)
>>> # Different arg, so function executes
```

Streamlit cheat sheet v1.0 | August 2020
https://github.com/daniellewisDL/streamlit-cheat-sheet

Using Template

- Basic tutorial https://github.com/amogh9594/basic-of-streamlit
- theming-showcase https://github.com/streamlit/theming-showcase
- geospatial applications https://github.com/giswqs/streamlit-template
- Dashboard https://github.com/Aonic7/Dashboard-Streamlit

import streamlit as **st**

Streamlit

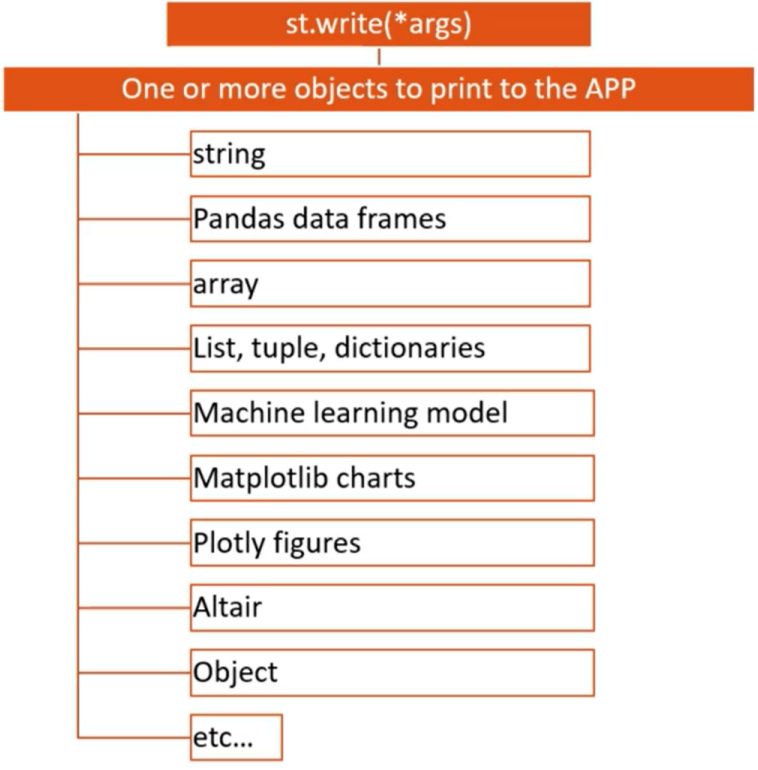
Display almost anything

st.write and magic commands

Streamlit has two easy ways to display information:

- **st.write**
- **Magic**

st.write



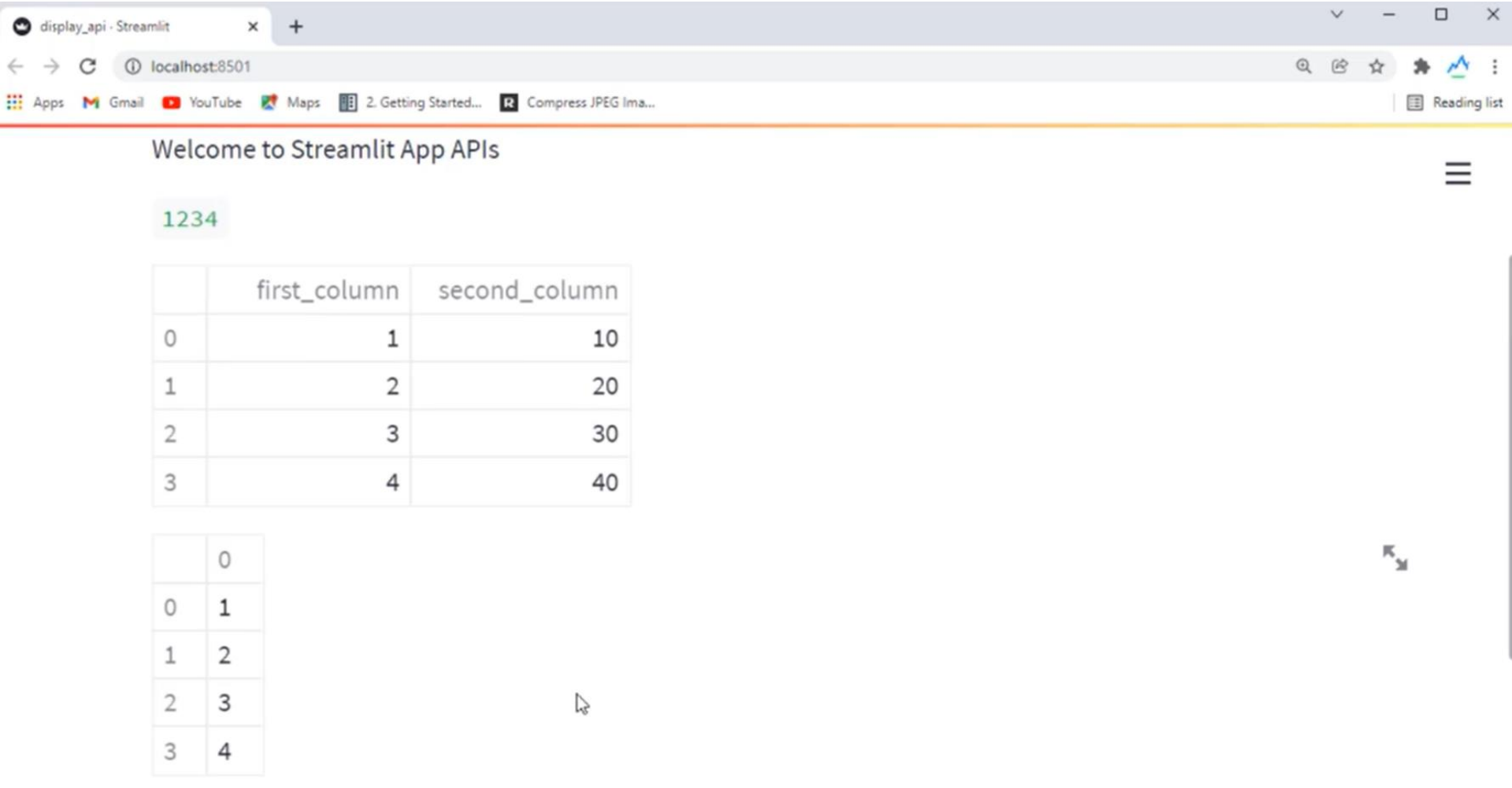
st.write

- Write arguments to the app

```
st.write(1234)
st.write(pd.DataFrame({
    'first column': [1, 2, 3, 4],
    'second column': [10, 20, 30, 40],
 }))
```

st.write

```
st.write('Welcome to Streamlit App APIs')

st.write(1234)

df = pd.DataFrame({
    'first_column':[1,2,3,4],
    'second_column': [10,20,30,40]
})

st.write(df)

## display numpy array
st.write(np.array([1,2,3,4]))
```


display_api · Streamlit
localhost:8501
Apps
Gmail
YouTube
Maps
2. Getting Started...
Compress JPEG Ima...
Reading list
Welcome to Streamlit App APIs
1234
first_column
second_column
0 1 10
1 2 20
2 3 30
3 4 40
0
0 1
1 2
2 3
3 4

Magic

- Magic commands are a feature in streamlit that allow you to write almost anything without having to type an explicit commands.

```
import pandas as pd
df = pd.DataFrame({'col1': [1,2,3]})
df  # 👈 Draw the dataframe
```

Magic

- Magic commands are a feature in streamlit that allow you to write almost anything without having to type an explicit commands.

```
st.write(df)

## display numpy array
st.write(np.array([1,2,3,4]))

## ---------------------------- MAGIC ----------------------------
st.write("Magic commands")

df1 = pd.DataFrame({'col1':[1,2,3,4]})

df1
x = 10
x
```

Streamlit

Text elements

Text elements

- St.markdown
- St.title
- St.header
- St.subheader
- St.text
- St.caption
- St.code
- St.latex

markdown.txt
text_elements_markdown.py

```python
import streamlit as st

st.markdown("""
# Markdown
# For Heading Level -1 or Title use (#)
## For Heading Level -2 or Header use (##)
### For Heading Level -3 or Subheader use (###)
#### For Heading Level -4

To create paragraphs, use a blank line to separate one or more lines of text.

---
```

Status Elements

```python
import streamlit as st
import time

## progress
st.header('st.progress')
st.caption('Display a progress bar')

my_bar = st.progress(0)

for pct_complete in range(1,101):
    time.sleep(0.5)
    my_bar.progress(pct_complete)
```

Media Elements

```
st.header('Display video')
video_file = open('./media/waterfalls.mp4','rb')
video_bytes = video_file.read()

st.video(video_bytes)

# display audio
st.header('Display audio')
audio_file = open('./media/audio.mp3','rb')
audio_bytes = audio_file.read()

st.audio(audio_bytes,format='audio/ogg')
```

Widgets

Widgets

- Button
- Checkbox
- Radio Button
- Select / Dropdown
- Multiselect
- Slider
- Text input
- Number Input
- Text area
- Date Picker
- File Uploader
- Color Picker

```
# load the data
data = pd.read_csv('tips.csv')

def display_data_random(df):
    sample = df.sample(5)
    return sample



# button widget
st.subheader('Displaying Random 5 Rows')
st.caption('click on the button below to display the row rando
button = st.button('Display random 5 rows')
if button:
    sample = display_data_random(data)
    st.dataframe(sample)
```

Button

```
# checkbox
st.markdown('---')
st.subheader('st.checkbox')
agree = st.checkbox('I agree to terms and conditions') # return boolean value
st.write('checkbox status =',agree)
```

Multiple Checkbox

```
# mutiple checkbox
with st.container():
    st.info('What technologies you know')

    python = st.checkbox('Python')
    datascience = st.checkbox('Data Science')
    ai_ml = st.checkbox('AI/ML')
    android = st.checkbox('Android')
    react = st.checkbox('React JS')
    java = st.checkbox('Core Java')
    javascript = st.checkbox('Java Script')
    tech_button = st.button('Submit')
    if tech_button:
        tech_dict = {
            'Python':python,
            'Data Science':datascience,
            'AI/ML':ai_ml,
            'Android':android,
            'React JS':react,
            'Core Java':java,
            'Java Script':javascript,
        }
        st.json(tech_dict)
```

```
# radio button
st.markdown('---')
st.subheader('st.radio')

radio_button = st.radio("what is your favorite color ?",
                        ('White','Black','Pink','Red','Blue','Green'))

st.write('Your favorite color is',radio_button)
```

```
# selectbox
st.markdown('---')
st.subheader('st.selectbox')

select_box = st.selectbox('What skill you want to learn most',
                          ('Java','Python','C','C++','JavaScript','HTML','Others'))
st.write('You selected =',select_box)
```

Selectbox

```
# multi select
st.markdown('---')
st.subheader('st.multiselect')

options = st.multiselect('What kind of movies you like',
                ['Comedy','Action','Sci-fi','Drama','Romance']
                )
st.write('you selected',options)
```

Multi Select

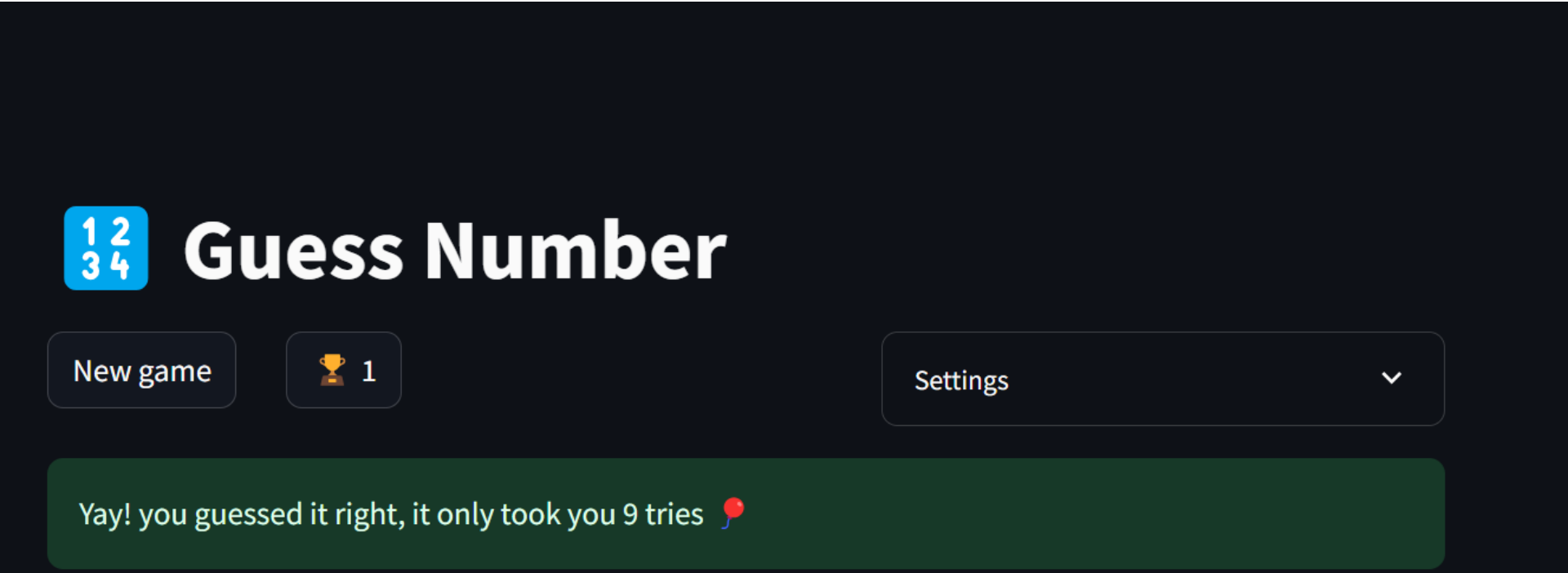
```
# slider
st.markdown('---')
st.subheader('st.slider')
loan = st.slider(
    'What is loan amout you are looking for ?',0,100000,1000,1000
)
st.write('Loan amount =',loan)
```

```
# text input
st.markdown('---')
st.subheader('st.text_input , st.number_input, st.date_input')

with st.container():
    name = st.text_input('Please enter your name')
    age = st.number_input('What is your name',min_value=0,max_value=150,value=25,step=1)
    decribe = st.text_area('Decription',height=150)
    dob = st.date_input('Select date of birth')

    submit_button = st.button('Submit Button')

    if submit_button:
        info = {
            "Name": name,
            'Age': age,
            'Date of Birth': dob,
            'About Yourself': decribe
        }
        st.json(info)
```

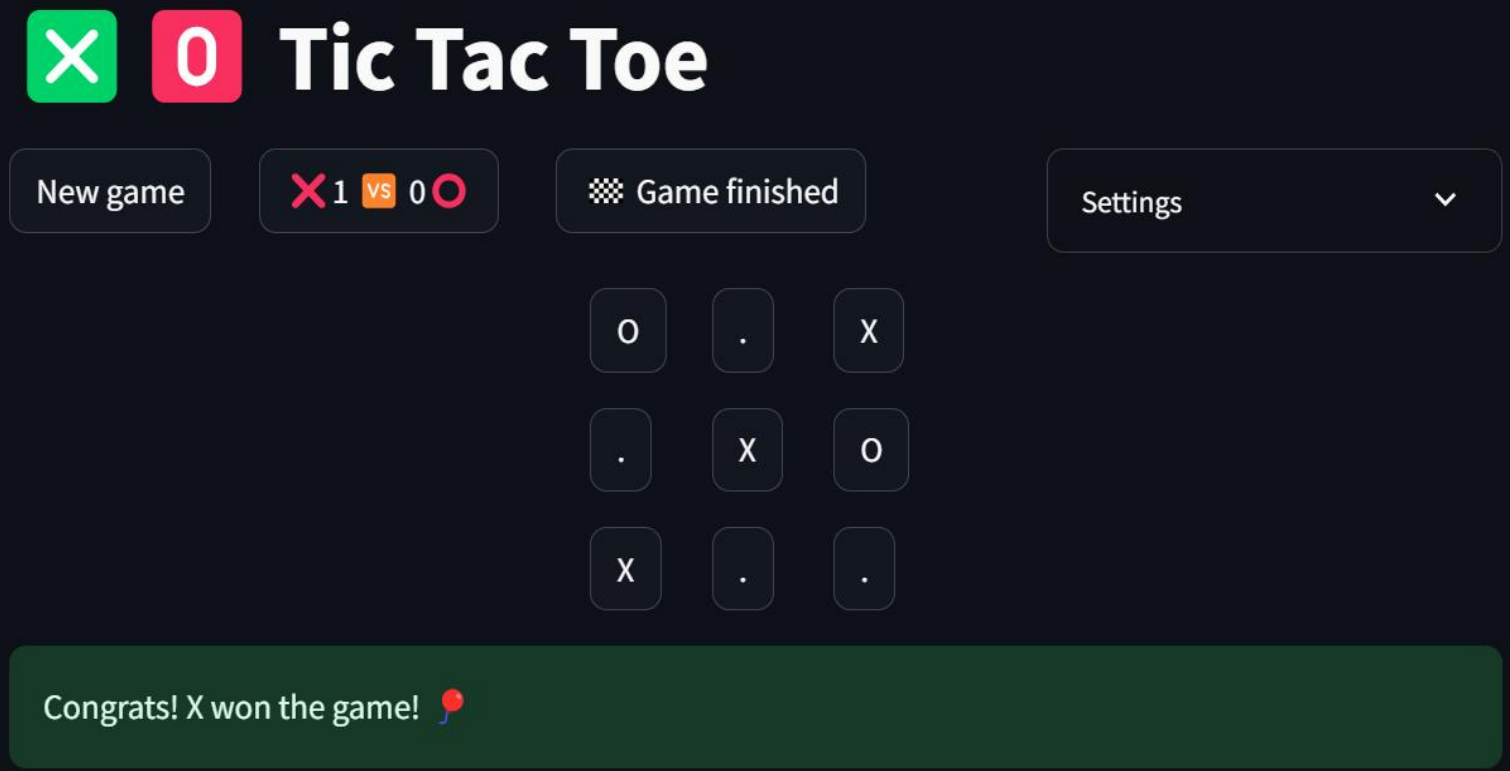
```
## fileuploader
st.markdown('---')
st.subheader('st.file_uploader')

uploaded_file = st.file_uploader('Choose a file')
save_button = st.button('save file')
if save_button:
    if uploaded_file is not None:
        with open(os.path.join("./save_folder",uploaded_file.name),mode='wb') as f:
            f.write(uploaded_file.getbuffer())

        st.success('File uploaded sucessfully')

    else:
        st.warning('Please select the file you want to upload')
```

Web Games

Guess Number Game



Tic Tac Toe


Tic Tac Toe
New game
1 vs 0
Game finished
Settings
O . X
. X O
X . .
Congrats! X won the game!

Thank You