Chapter 2: Components - Starting with Write and Magic

In these chapters, we are going to explore many components. For this chapter, we will focus on 'Write and Magic' components.

No description has been provided for this image

What's the meaning of 'Write and Magic'?

- `Write and Magic` refers to components to display text and beautiful, informative widgets. These are an important part in Streamlit.

What are the components inside this collection?

- There are **3** components inside the `Write and Magic` collection. They are:
    - st.write
    - st.write_stream
    - Magic

Here, `st` refers to Streamlit imported. This is a common convention, in the form of:

In [ ]:
```
import streamlit as st
# more code ...
```

`st.write`

- Used to render text in the body of the web app. This can optionally run HTML by setting `unsafe_allow_html` to `True`, though you CAN use `st.html` for rendering HTML or CSS but without Markdown text.

- Function signature: st.write(`*args` , `unsafe_allow_html=False`)

- A basic example of this is as follows:

In [ ]:
```python
import streamlit as st
import pandas as pd

st.write("Hello, *World!* :sunglasses:") # Writing some text
st.write(pd.DataFrame({
    "first column": [1, 2, 3, 4],
    "second column": [10, 20, 30, 40],
})) # Now a Pandas DataFrame
```

No description has been provided for this image

`st.write_stream`

- Used to stream a generator or an iterable. Best used for AI-based applications.

- A good fact is that this can render different sorts of data, which is pretty useful.

- Function signature: st.write_stream(stream)

- A basic example is as follows:

In [ ]:
```python
import time
import numpy as np
import pandas as pd
import streamlit as st

_LOREM_IPSUM = """
Lorem ipsum dolor sit amet, **consectetur adipiscing** elit, sed do eiusmod te
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis
nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
"""


def stream_data(): # The generator yielding data
    for word in _LOREM_IPSUM.split(" "):
        yield word + " "
        time.sleep(0.02)

    yield pd.DataFrame(
        np.random.randn(5, 10),
        columns=["a", "b", "c", "d", "e", "f", "g", "h", "i", "j"],
```

```
    )

    for word in _LOREM_IPSUM.split(" "):
        yield word + " "
        time.sleep(0.02)

st.write_stream(stream_data)
```

▶ Watch the video

`Magic`

- It is a handy feature in Streamlit that allows you to create any type of data without specifying a proper command.
- You just have to type in the data or literal and it will magically appear in your app's body.
- An explanation is given below:

In [ ]:
```
import pandas as pd
df = pd.DataFrame({'col1': [1,2,3]}) # Instantiating a simple DataFrame
df # This is where the magic happens
```

No description has been provided for this image

In the above code, in the 3rd line with only the literal `df`, Streamlit automatically detects it and renders it without any member functions to make it appear, with the help of `st.write`. This makes code neater and cooler.

- You can always turn Magic off by going to the directory `~/.streamlit/config.toml` with this setting:

```
[runner] magicEnabled = false
```

Summary:

---

st.write: We learnt how to write text and DataFrames with st.write. st.write_stream: We learnt how to stream data from generators using st.write_stream. Magic: We learnt how to use Streamlit's native `Magic` feature to create neater code.