# Natural Language Processing Lab
# Week 6

Streamlit Tutorial

# What is Streamlit?

- a free, open-source, python-based framework that allows you to quickly build interactive dashboards and machine learning web apps

Streamlit

# If you…

- know Python
- have zero front-end web development knowledge
- want something quick and simple to show off your results

# Streamlit is for you!

# Streamlit’s great because…

- it requires no front-end (HTML, JavaScript, CSS) experience or knowledge.
- you only need a few lines of code to create amazing web apps.
- everything’s Python, so it’s compatible with most Python libraries (e.g. pandas, matplotlib, seaborn, plotly, Keras, PyTorch).
- it provides a caching mechanism to speed up your app and save it from expensive computations

# How to install

1. Make sure you have PIP and Python 3.7 - 3.10
2. Create a new virtual environment using your prefered managenment tool.
3. Go to terminal and type `pip install streamlit` to install.
4. Use `streamlit hello` to check if the installation worked, Streamlit's Hello app should appear in a new tab in your default web browser.

Click here for more detailed instructions.

# Getting Started

Once you have Streamlit installed, you can add `import streamlit as st` to your Python script and start using Streamlit commands.

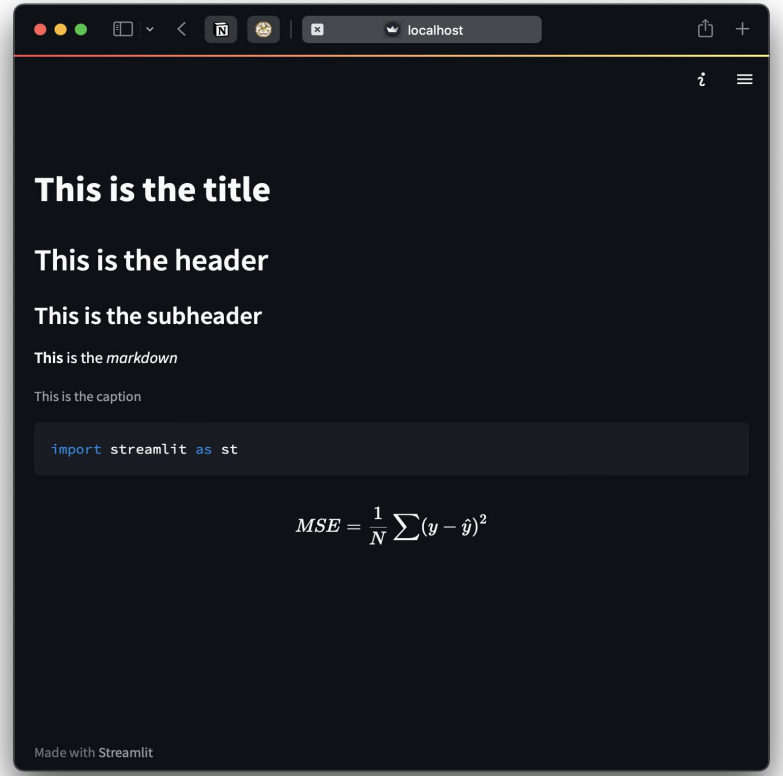
To run your app, use the command `streamlit run script_name.py`, a local Streamlit server will spin up and your app will open in a new tab in your default web browser.

To update your app, save the source file. Streamlit detects if there is a change and asks you whether you want to rerun your app. Choose "Always rerun" at the top-right of your screen to automatically update your app every time you change its source code.

# Text Elements

`st.title()`: This function allows you to add the title of the app.
`st.header()`: This function is used to set header of a section.
`st.subheader()`: This function is used to set sub-header of a section.
`st.markdown()`: This function is used to set a markdown of a section.
`st.caption()`: This function is used for captions.
`st.code()`: This function is used to display code.
`st.latex()`: This function is used to show mathematical expressions formatted as LaTeX.

```
st.title("This is the title")
st.header("This is the header")
st.subheader("This is the subheader")
st.markdown("**This** is the *markdown*")
st.caption("This is the caption")
st.code("import streamlit as st")
st.latex(r''' MSE = \frac 1 N\sum(y - \hat y)^2 ''')
```



# Input Widgets

`st.checkbox()`: This function returns a Boolean value. When the box is checked, it returns a True value, otherwise a False value.

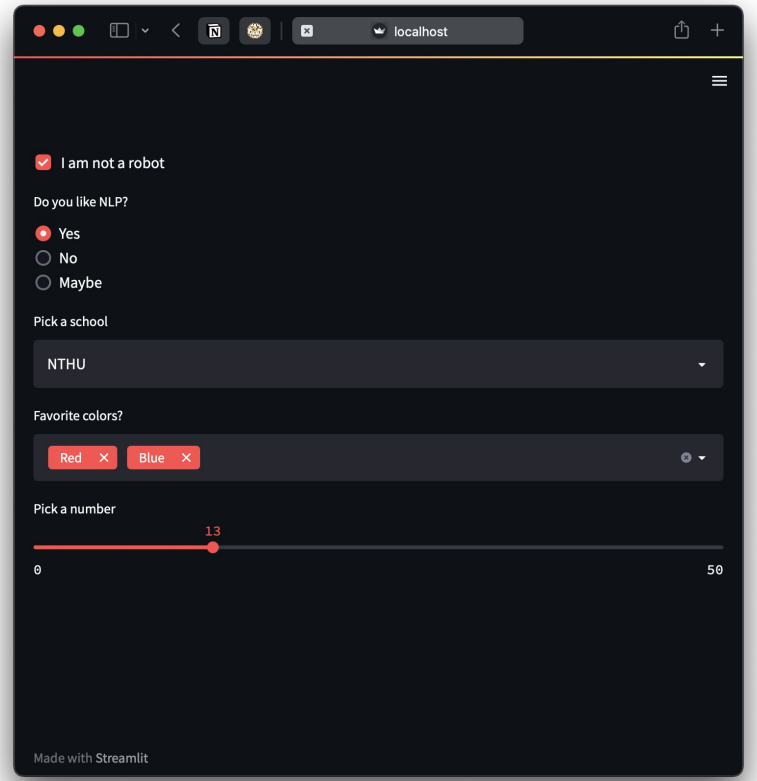
`st.radio()`: This function is used to display a radio button widget.

`st.selectbox()`: This function is used to display a select widget.

`st.multiselect()`: This function is used to display a multiselect widget.

`st.slider()`: This function is used to display a slider widget.

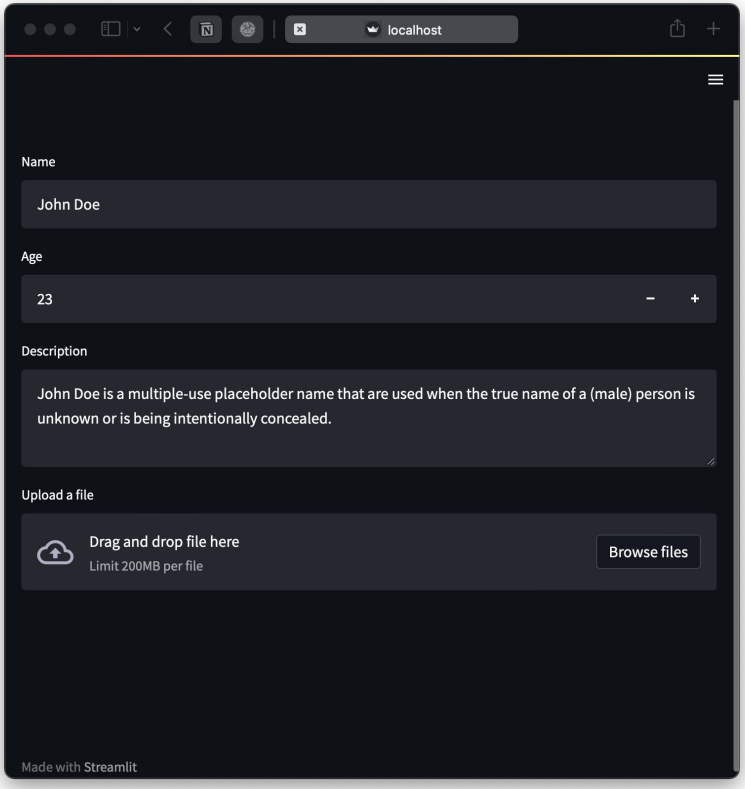
```
st.checkbox('I am not a robot')
st.radio('Do you like NLP?',['Yes','No', 'Maybe'])
st.selectbox('Pick a school',['NTHU','NYCU'])
st.multiselect("Favorite colors?",['Red','Blue','White'])
st.slider('Pick a number', 0,50)
```



# More Input Widgets

`st.text_input()`: This function is used to display a text input widget.
`st.number_input()`: This function is used to display a numeric input widget.
`st.text_area()`: This function is used to display a text input widget with more than a line text.
`st.file_uploader()`: This function is used to display a file uploader widget.

```
st.text_input('Name')
st.number_input('Age', 0,100)
st.text_area('Description')
st.file_uploader('Upload a file')
```

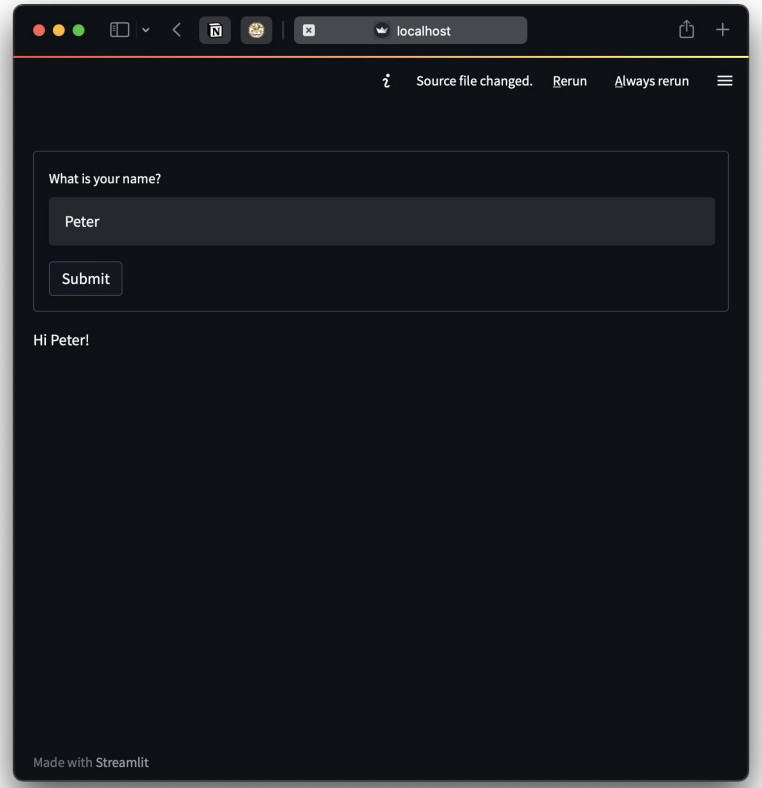


# Forms

`st.form_submit_button()`: This function creates the submit button for a form, returning a boolean to indicate whether the form was submitted or not. This allows for building additional logic upon submit button.

```
form = st.form(key='form-name')
name = form.text_input('What is your name?')
submit = form.form_submit_button('Submit')

if submit:
    st.write(f'Hi {name}!')
```



# Other Features

With Streamlit, you can also:

- use layouts and containers
- add a side bar
- display audio, images, and videos
- show tables, dataframes, charts, and graphs
- make a multi-page app

For more information, check out the official documentation here.

# Deploying the App

1. Create a new public Github repository and upload your files.
2. Set up a Streamlit Cloud Account here.
3. Log in to Streamlit Cloud and click “New app”.
4. Fill in your repo, branch, and file path, and click "Deploy".
5. Wait a few minutes for your app to launch.

The default URL of your deployed app will be:

`https://[user name]-[repo name]-[branch name]-[app path]-[short hash].streamlitapp.com`

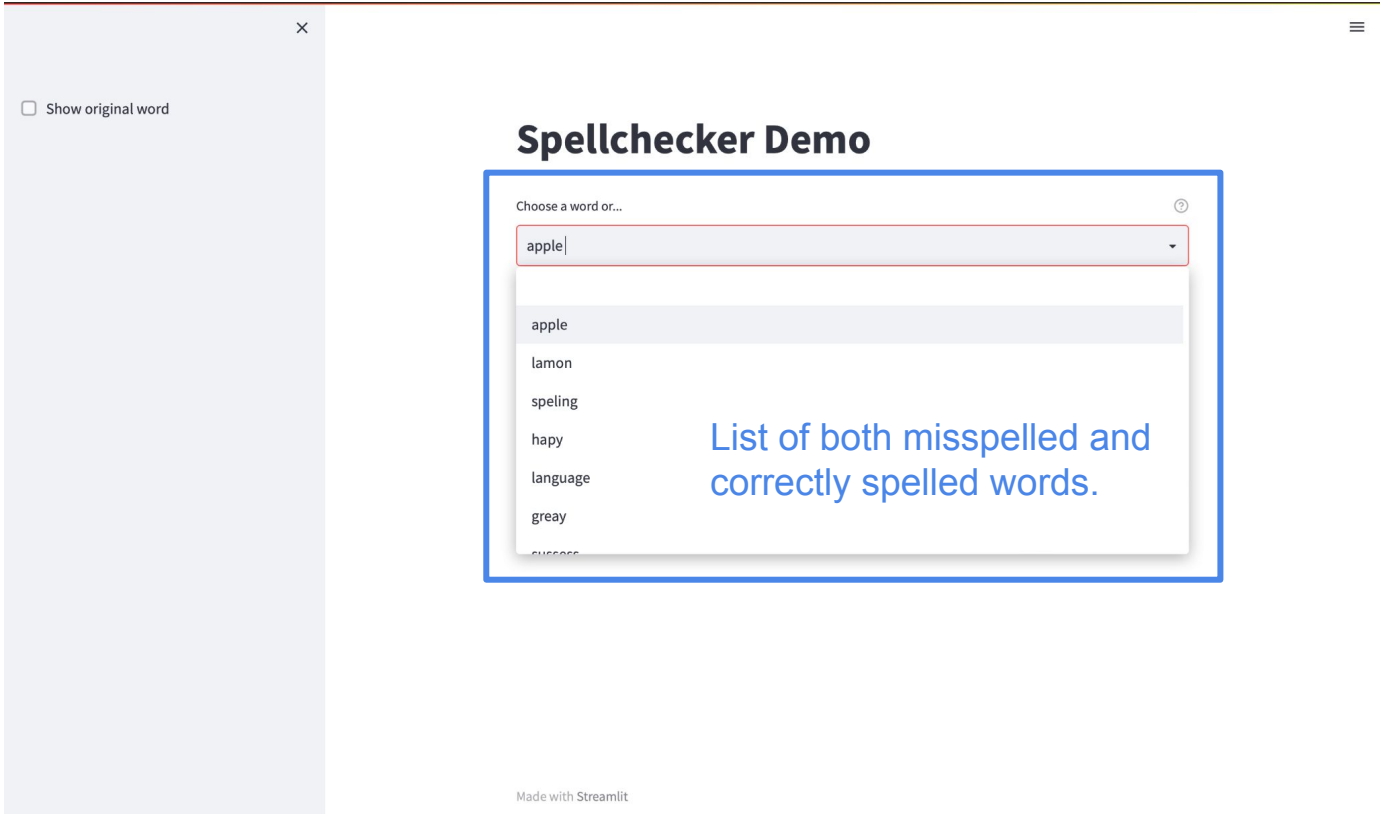
You can also use a custom subdomain to change it to:

`https://<your-custom-subdomain>.streamlitapp.com`

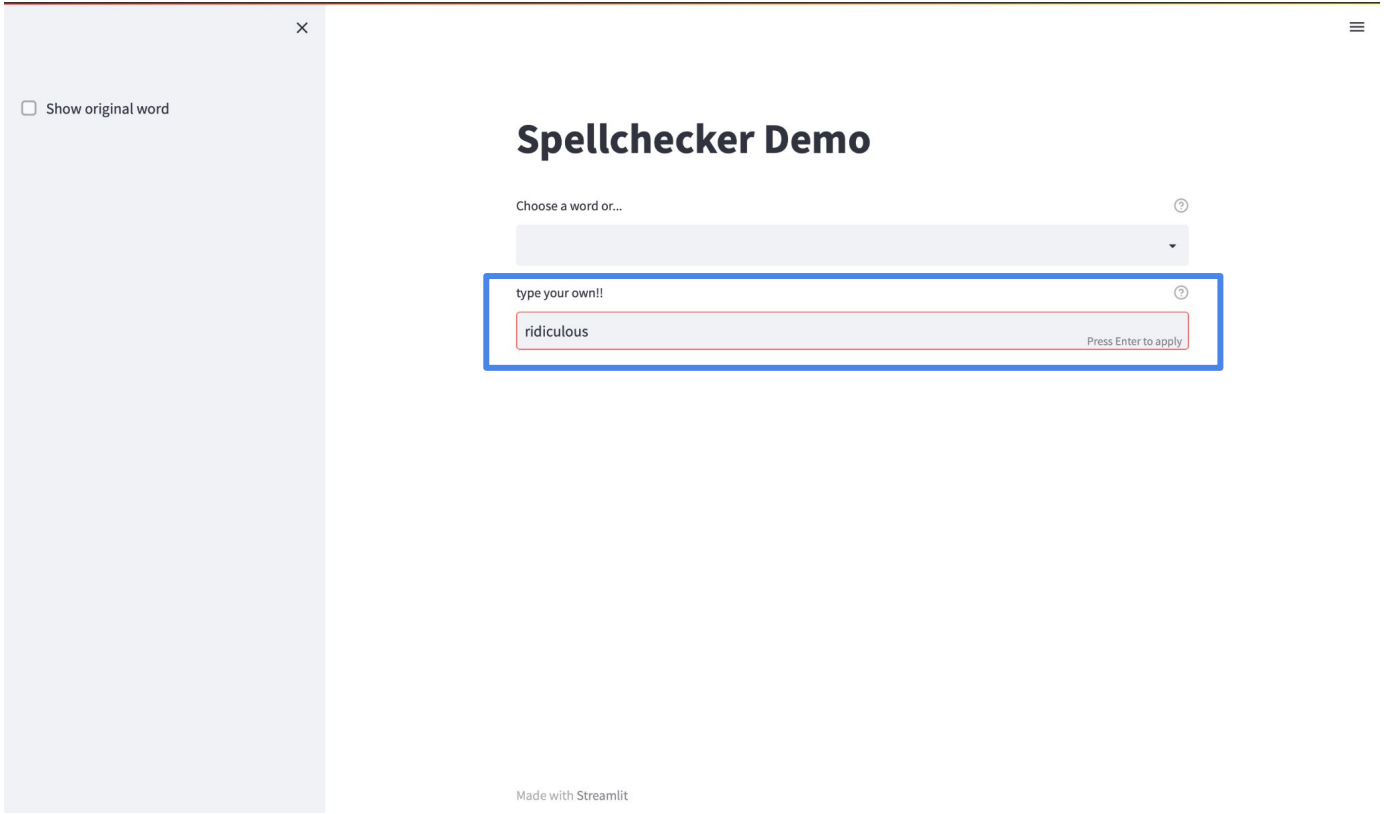
# Your Turn

- Create a Streamlit app to show off your spellchecker from Week 1
- It should include
  - a drop-down list of sample words ⇒ +20
  - a text field for users to type their own word ⇒ +20
  - a sidebar containing a checkbox to choose whether to display the original word ⇒ +20
  - the corrected word
    - inside a green box if the word is spelled correctly ⇒ +10
    - red if the word is misspelled ⇒ +10
- Deploy your app using Streamlit Cloud, and change the URL to `https://<your-studentID>.streamlitapp.com`, or if it's taken, `https://<week7-your-studentID>.streamlitapp.com` ⇒ +20
- Hint: just add the Streamlit code to your `spell.py`

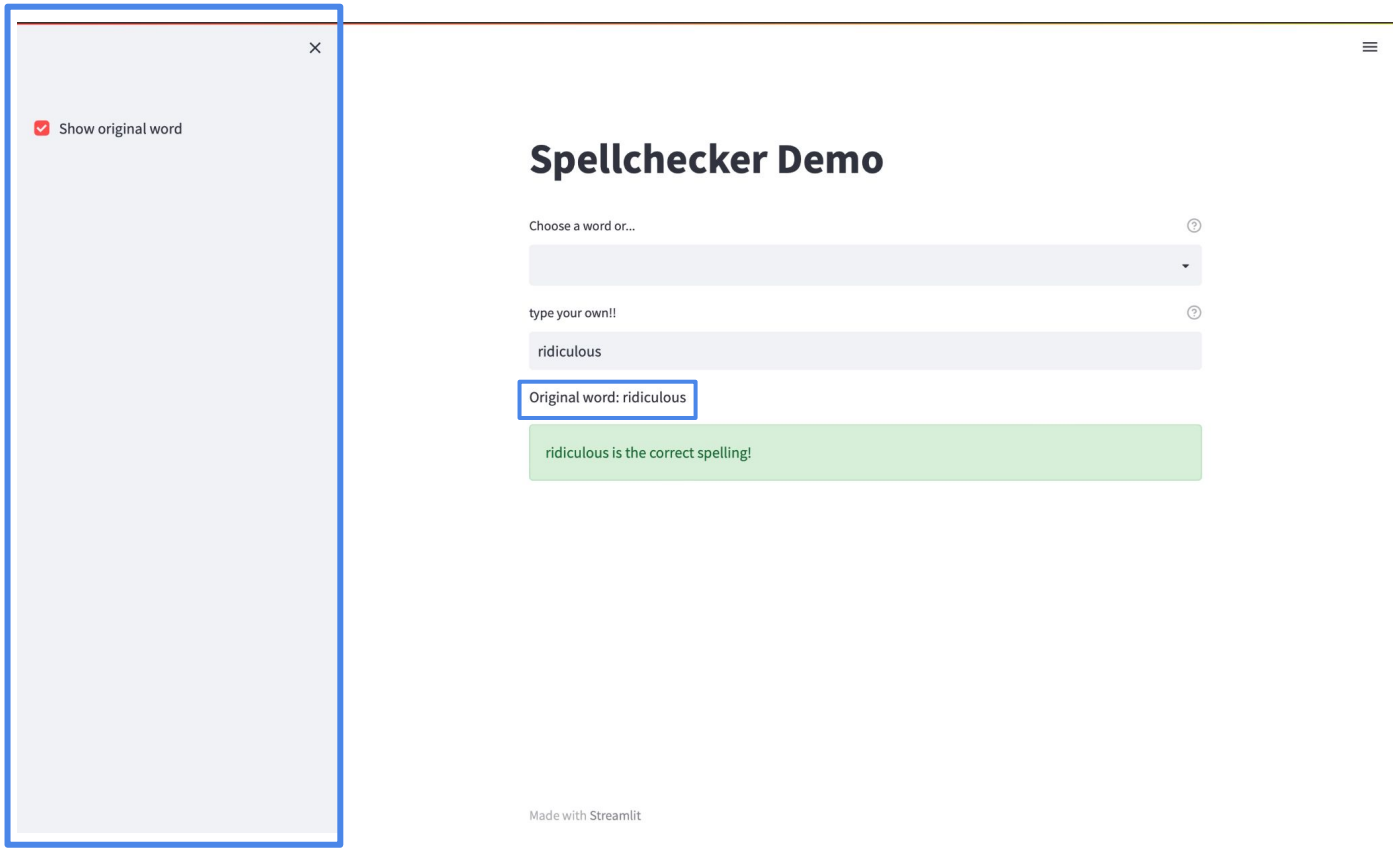
# Example - sample words



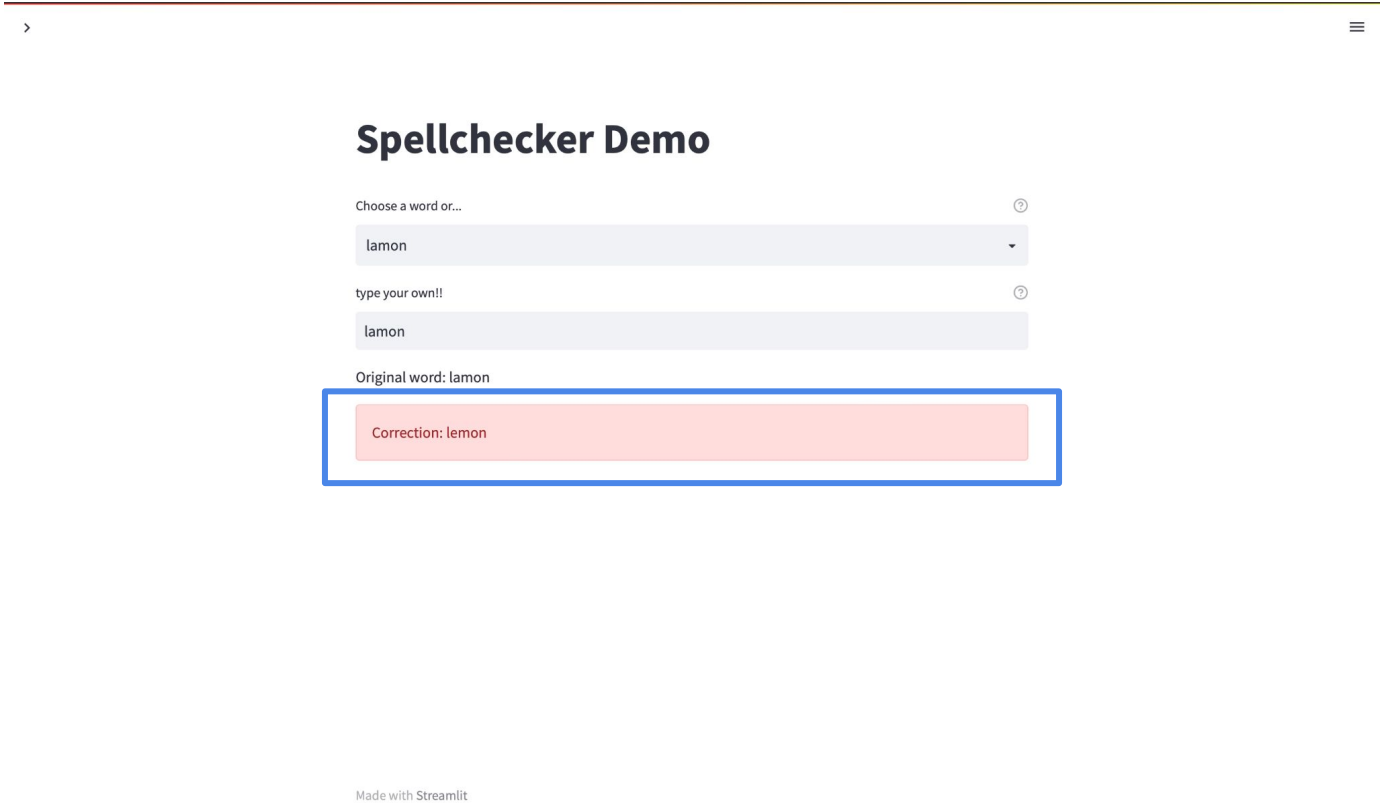
# Example - text box



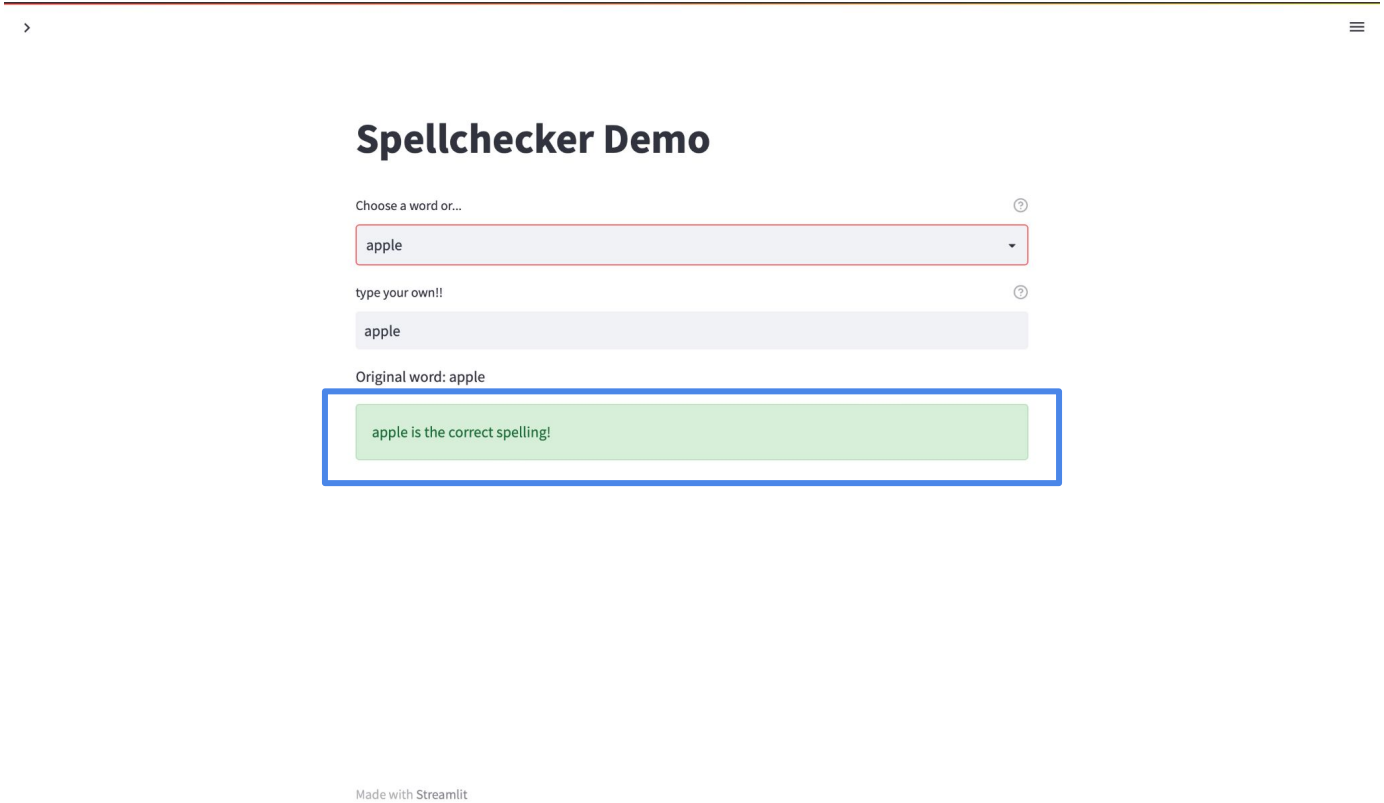
# Example - sidebar



# Example - output (red)



# Example - output (green)



# Reminders

- Hand in **your code** `{student id}.py` and **URL** on **elearn**.
- Make an appointment to show your work here.
- Deadline for assignment 7: 15:29 a.m. Nov 3 (Thu)