In [ ]:

```
# https://docs.streamlit.io/library/get-started/main-concepts
import streamlit as st
import pandas as pd
import numpy as np
import os
import time



## Menu
menu_options = ("Dataframe", "Widget", "Layout", "Theme", "Cache", "Misc")
default_ix = menu_options.index("Cache")
menu_item = st.sidebar.selectbox("Pick a Concept:", menu_options, index=default_ix)
st.sidebar.write(
"""
Since Streamlit runs script from top to bottom, we use menu-item to break out
the whole script into sections, so each rerun occurs for a selected section only
"""
)

## UI Layout
st.sidebar.header('Layout - sidebar')

# Add a selectbox to the sidebar:
add_selectbox = st.sidebar.selectbox(
    'How would you like to be contacted?',
    ('Email', 'Home phone', 'Mobile phone')
)
st.sidebar.write(f"You selected: {add_selectbox}")

# Add a slider to the sidebar:
add_slider = st.sidebar.slider(
    'Select a range of values',
    0.0, 100.0, (25.0, 75.0)
)
st.sidebar.write(f"Range: {add_slider}")

add_slider2 = st.sidebar.slider(
    'Select a values',
    0.0, 100.0, 50.0
)
st.sidebar.write(f"Value: {add_slider2}")



if menu_item == "Dataframe":
    ## Display Data
    st.header('Dataframe')
    st.subheader('st.write anything')
    st.write(pd.DataFrame({
        'first column': list(range(5)),
        'second column': [100*i for i in range(5)]
    }))


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

```
    df  # st.write(df)

    st.subheader('st.dataframe makes interactive table')
    df = np.random.randn(10, 20)
    st.dataframe(df)

    dataframe = pd.DataFrame(
        np.random.randn(10, 20),
        columns=('col %d' % i for i in range(20)))

    st.dataframe(dataframe.style.highlight_max(axis=0))

    st.subheader('st.table makes static table')
    st.table(dataframe)

    st.header('Charts and Maps')
    chart_data = pd.DataFrame(
         np.random.randn(20, 3),
         columns=['a', 'b', 'c'])

    st.line_chart(chart_data)

    map_data = pd.DataFrame(
        np.random.randn(1000, 2) / [50, 50] + [37.76, -122.4],
        columns=['lat', 'lon'])

    st.map(map_data)

if menu_item == "Widget":

    ## UI control
    st.header('Widgets ')
    x = st.slider('x')  # 👈 this is a widget
    st.write(x, 'squared is', x * x)

    st.text_input("Your name", key="name")
    # You can access the value at any point with:
    st.session_state.name

    df = pd.DataFrame({
        'first column': [1, 2, 3, 4],
        'second column': [10, 20, 30, 40]
        })
    option = st.selectbox(
        'Which number do you like best?',
         df['first column'])

    'You selected: ', option

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

    if st.checkbox('Show dataframe'):
        chart_data = pd.DataFrame(
           np.random.randn(20, 3),
           columns=['a', 'b', 'c'])

        chart_data
```

```
    st.subheader('st.progress ...')
    'Starting a long computation...'

    # Add a placeholder
    latest_iteration = st.empty()
    bar = st.progress(0)

    for i in range(100):
      # Update the progress bar with each iteration.
      latest_iteration.text(f'Iteration {i+1}')
      bar.progress(i + 1)
      time.sleep(0.1)

    '...and now we\'re done!'

if menu_item == "Layout":


    ## Layout
    st.header('Layout')

    left_column, right_column = st.columns(2)
    # You can use a column just like st.sidebar:
    left_column.button('Press me!')

    # Or even better, call Streamlit functions inside a "with" block:
    with right_column:
        chosen = st.radio(
            'Sorting hat',
            ("Gryffindor", "Ravenclaw", "Hufflepuff", "Slytherin"))
        st.write(f"You are in {chosen} house!")

if menu_item == "Theme":
    ## Theme
    st.header('Theme')


if menu_item == "Cache":

    ## Caching
    st.header('Caching')
    @st.cache  # 👈 This function will be cached
    def Fibonacci(n):
        # Function for nth Fibonacci number
        # Check if input is 0 then it will
        # print incorrect input
        if n < 0:
            print("Incorrect input")

        # Check if n is 0
        # then it will return 0
        elif n == 0:
            return 0

        # Check if n is 1,2
        # it will return 1
        elif n == 1 or n == 2:
            return 1

        else:
```

```
            return Fibonacci(n-1) + Fibonacci(n-2)

    num = st.slider("num", 1, 100, 5)
    ts_start = time.time()
    fib_num = Fibonacci(num)
    ts_stop = time.time()
    duration = ts_stop - ts_start
    st.write(f"Fib({num}) = {fib_num} \n calculated in {duration:.3f} sec")

    st.button("Rerun")

    st.write("Notice that (1) calculating Fib of the same number takes constant 0.001 o


if menu_item == "Misc":
    st.header('Misc')

    st.write(f"os.getcwd() = {os.getcwd()}" )
```