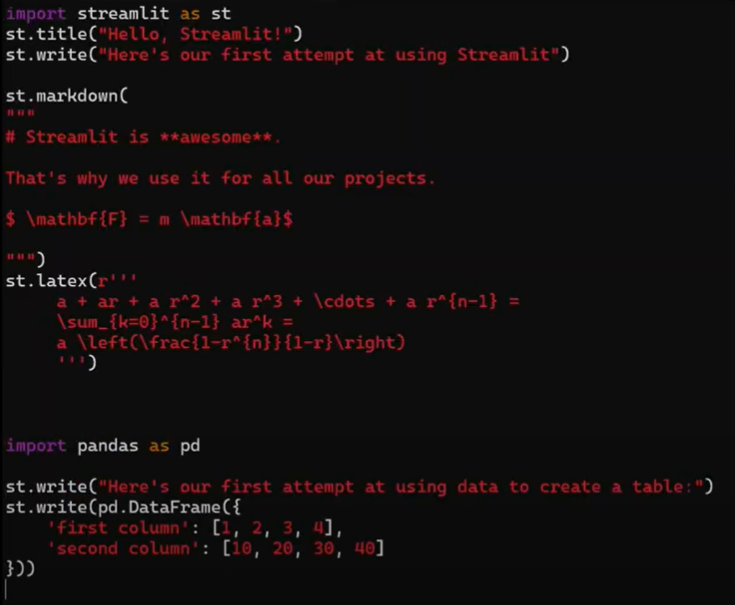
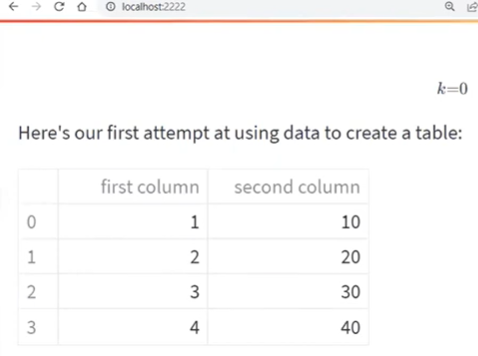
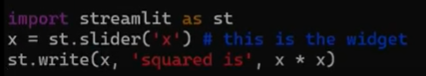
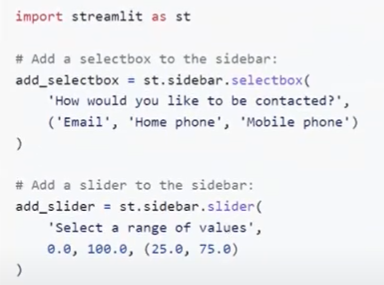
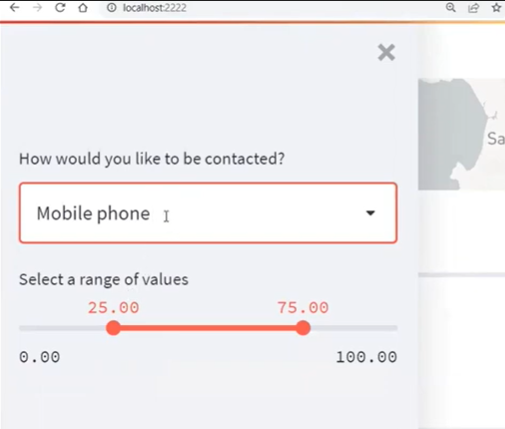
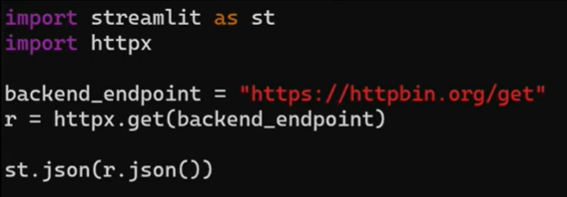
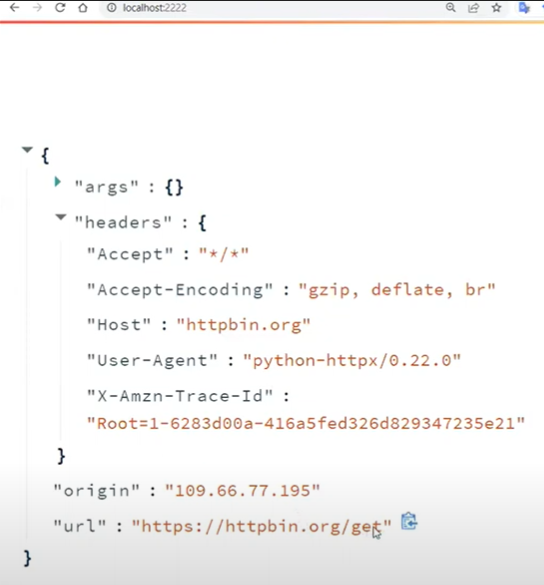
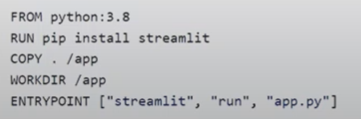
**EASS שיעור 11- עבודה עם GITHUB, המשך STREAMLIT,  
 CMD VS ENTRYPOINT**

הערות לעבודה עם github:

1. קבצים שלא נרצה שיוצגו בגיטהאב יש לשים בקובץ gitignore (כמו .DS\_Store , תיקיית .vscode , cache וכו').  
   כדי לכלול בקובץ את כל הקבצים שיש להם סיומת זהה ניתן להשתמש בכוכבית, לדוגמה: \*.cache  
   ניצור קובץ .gitignore ונרשום בו בכל שורה את השמות של הקבצים/תיקיות המיותרים.   
   ניתן לחפש באינטרנט תבניות מוכנות template של קבצי gitignore להשראה.
2. קובץ readme איכותי. קישור להסברים והדגמות:   
   <https://docs.github.com/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax>
3. בכל העלאת קבצים לגיטהאב ניצור commit לכל שינוי עם שם משמעותי.
4. עבור כל מטלה/פרוייקט ניצור repository נפרד.
5. ישנה תוכנה בשם github desktop שבה ניתן לערוך ולהעתיק repositories.
6. עבודה עם branch: ניצור branch שבו נעלה שינויים, כשנאחד את השינויים עם הbranch הראשי נמחק את הbranch המשני. מצב שבו לא נמחק branches זה כשיש גרסאות שונות אז לכל גרסה יהיה branch.

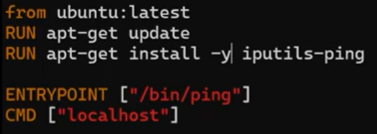
המשך Streamlit:

* בפקודת write ניתן ליצור ולהדפיס טבלה בעזרת pandas dataframe  
    
  
* שימוש ב-widget כדי לקבל קלט מהמשתמש. דוגמה:  
    
  
* Layout – הוספת אלמנטים לפריסת המסך כמו תפריט צד, תא שניתן להרחיב ולצמצם אותו. קישור לדוגמאות נוספות: <https://docs.streamlit.io/library/api-reference/layout>  
    
  
* תקשורת עם ה-backend : נשתמש בבקשות HTTP בעזרת ספריית httpx  
    
  
* שימוש ב-Streamlit באמצעות Dockerfile:  
  

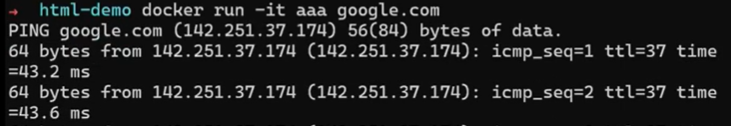
הבדל בין ENTRYPOINT ל-CMD:

שתי הפקודות הללו מגדירות מה ירוץ כשהקונטינר יתחיל.

ההבדל בינהן הוא ש-ENTRYPOINT נותן פרמטרים שלא "נדרסים" אוטומטית (אלא "נדרסים" רק אם נשים --entrypoint עם פרמטר חדש בשורת ה-run), לעומת זאת CMD נדרס אוטומטית אם נשים פרמטר (ללא צורך בקידומת).



נוסיף את פרמטר google.com שיחליף את localhost אוטומטית:



נוסיף את קידומת --entrypoint עם הפרמטר החדש שלה כדי לדרוס את /bin/ping :

