**מה זה גיט ?**

**מדריך גיט - Git Tutorial**

היא מערכת בקרת גרסאות המשמשת למעקב אחר שינויים בקוד המקור במהלך פיתוח תוכנה.

**שימושים עיקריים :**

1. **בקרת גירסא –**

מאפשרת למפתחים לעקוב אחר שינויים בבסיס הקוד שלהם לאורך זמן. כל שינוי שנעשה בקוד נשמר כ-commit, יוצר היסטוריית גרסאות. מאפשר למפתחים לחזור לגרסאות קודמות, לשתף פעולה ביעילות ולשמור תיעוד של כל השינויים.

1. **ארכיטקטורה מבוזרת-**

לכל מפתח בתוך הפרויקט יש עותק מלא של המאגר, כולל ההיסטוריה שלו. מאפשר עבודה במצב לא מקוון ושיתוף פעולה בצורה חלקה עם אחרים מבלי לדרוש חיבור מתמיד לשרת מרכזי.

1. **Branching and Merging-**

Git מאפשרת למפתחים ליצור branches ("ערוצים" (לעבודה על תכונות או תיקוני באגים באופן עצמאי מבסיס הקוד הראשי.

מאפשר ניסויים ופיתוח מקביל. יהיה ניתן למזג branches בחזרה לתוך בסיס הקוד הראשי.

1. **Repositories-**

Repository Git הוא אוסף של קבצים והיסטוריית הגרסאות שלהם. זה יכול לשבת בשרת מרוחק (כמו GitHub, GitLab או Bitbucket) או באופן מקומי במחשב של מפתח.

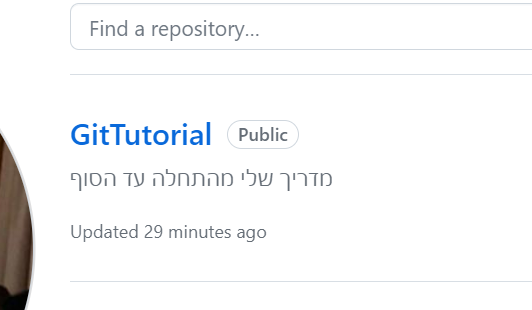
5 **- Pull Requests**

בפלטפורמות אירוח של Git כמו GitHub, מפתחים יכולים להציע שינויים למאגר על ידי יצירת בקשות משיכה. בקשות משיכה מאפשרות לאחרים לסקור את השינויים לפני שהם מתמזגים לתוך בסיס הקוד הראשי.

**תהליך עבודה:**

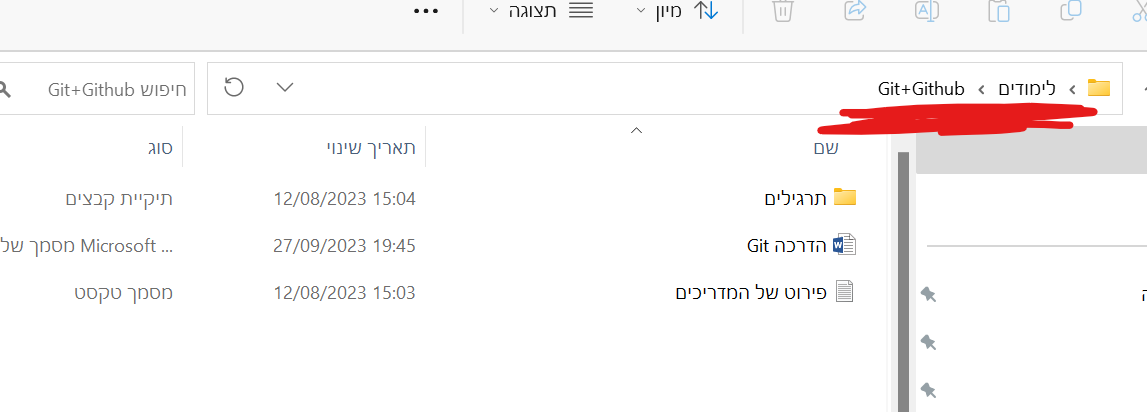
**פתיחת פרויקט ומיקומו-**

כאשר אני פותח פרויקט חדש בGIT, אני קודם ניגש לGITHUB ופותח פרויקט חדש עם השם שאני רוצה.



**1.** בוחר תיקייה במחשב שבה אני רוצה שהפרויקט ימוקם על ה\*מחשב האישי\*

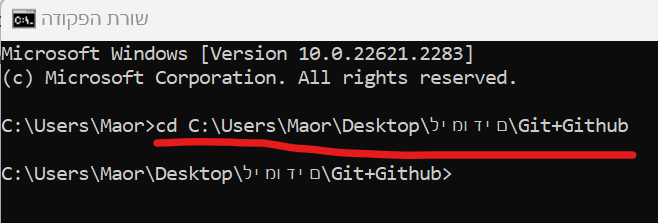
ומעתיק את הנתיב של התיקייה (ניתן להעתיק למעלה ליד שורת החיפוש).



**2.** פותח את הCMD ומשתמש בפקודת cd בשביל להיכנס לתיקייה.

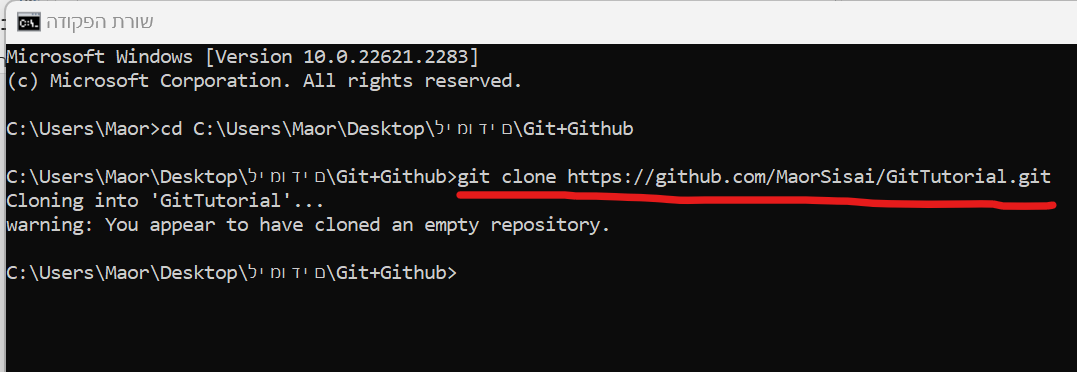
מיקום תיקיה cd / cd <folder path>

**הערה:** אפשר גם להיכנס לתיקיה הלוקאלית ולכתוב בpath של התיקייה cmd וזה יבצע את אותה הפעולה



**3.** כעת כשאני במיקום שרציתי, אני מבצע git clone לשם הפרויקט מהGIT שרציתי לשים על המחשב עם הלינק שלו ב GitHub שים לב שהסיומת צריכה להיות .git

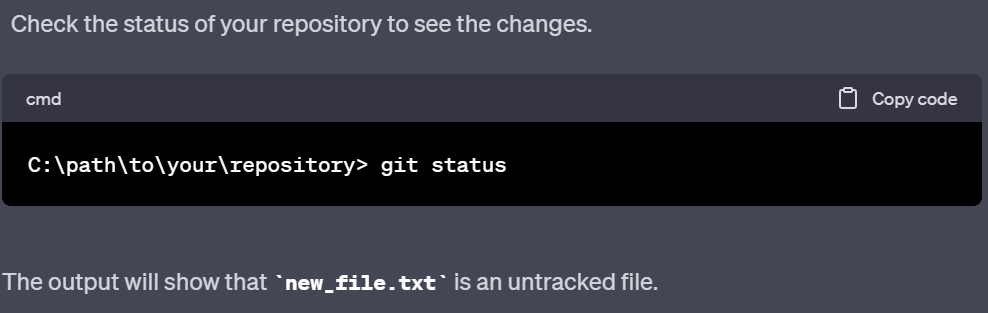
דוגמא: git clone https://github.com/MaorSisai/GIT-toturial.git



כעת נוצרת לי תיקייה חדשה בשם הפרויקט GIT בתוך הנתיב שבחרתי לוקאלית, תיקייה זאת מקושרת על GITHUB ועליה נרצה לעבוד. על מנת לעבוד עליה ולבצע פעולות שקשורות לGit אני צריך להיכנס לתיקייה עצמה כי היא זאת שבעצם מקושרת לGit, ניתן להיכנס שוב בעזרת פקודת cd פלוס הנתיב של התיקייה.

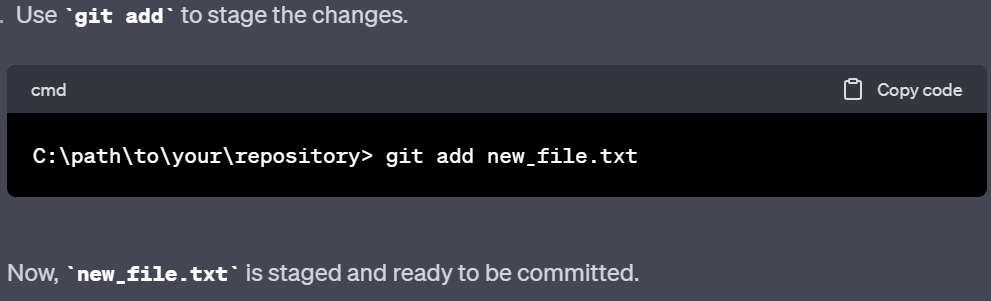
**פעולות בסיסיות:**

1. - **git status** מראה את הסטאטוס של התיקייה. ניתן לראות אם התבצעו שינויים בתיקייה כגון הוספת קבצים, שינויים בקבצים קיימים וכן הלאה.

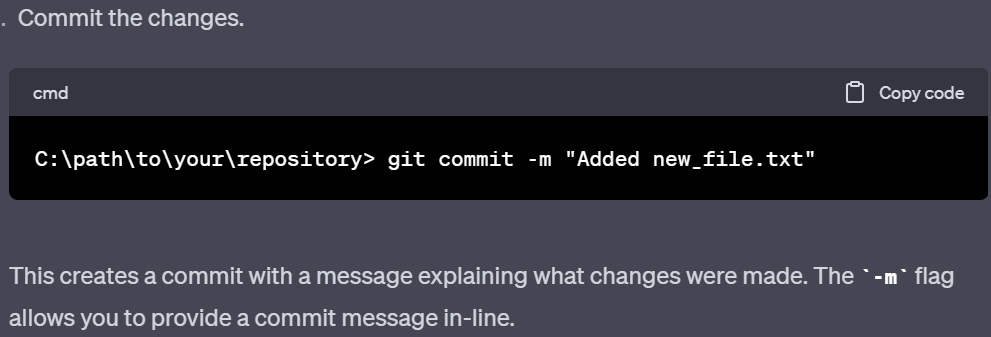
דוגמא קטנה, יצרתי קובץ חדש מחוץ לתיקיה ופשוט משכתי אותו לתיקייה הנוכחית, מתי שאני אבדוק את הסטאטוס של התיקייה אני אוכל לראות שהתווסף קובץ ואת שמו.

1. **git add .** – הוספה של \*כל\* השינויים לתיקייה. בשביל שהשינויים ישמרו בתיקיה אני צריך להשתמש בפקודה הזו (הכנה לcommit ).

אם אני רוצה שינוי ספציפי אני פשוט בוחר את השינוי הרלוונטי.

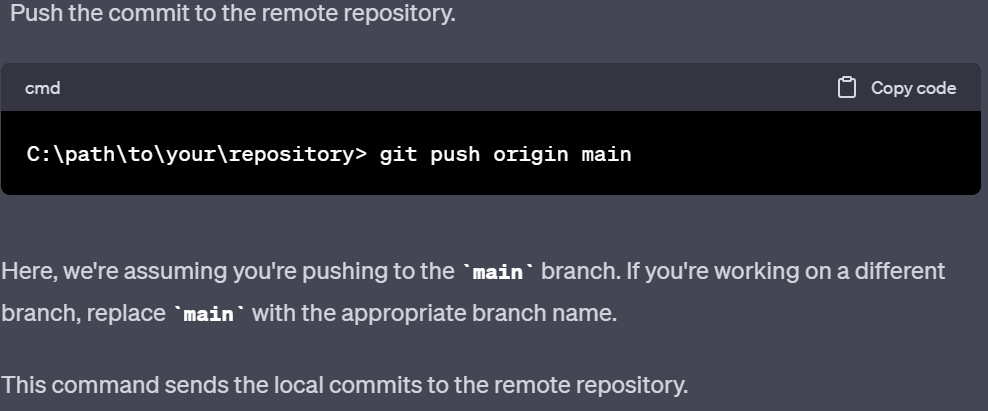


1. **git commit -m "comment"** – מבצע את השינויים האלה עם הערה קצרה לבחירתנו



1. **git push**- "דוחף" את השינויים ששמרתי לGitHub החיצוני

סעיפים 2,3,4 ביחד מסכמים איך מוסיפים דברים לGit (עריכות/קבצים/תיקיות)



1. **git push origin <branch name**>- בהמשך נלמד על ענפים, כשיש לי מספר ענפים אני צריך גם לדעת לבחור לאן אני דוחף שינויים.

צד שמאל זה היעד אליו אני דוחף, צד ימין זה ממי אני דוחף.

(מופיע בתמונה למעלה)

**ענף/Branch-**

כאשר אני רוצה לעבוד על "ענף" משלי, כלומר גירסה נוספת של הקוד שעליה אני אבצע שינויים והיא תישמר בערוץ מקביל לקוד הראשי ואז במידת הצורך אם ארצה לשלב את הקוד הראשי עם הענף המקביל שיצרתי אני אוכל לעשות זאת.

**מה זה נותן לנו ?**

כמה אנשים יכולים לעבוד במקביל על אותו הקוד.

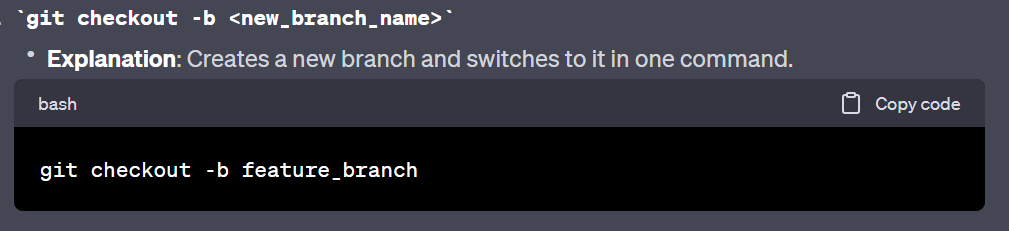
יש בקרה על שינויים ספציפיים שביצעתי בלי פגיעה בקוד המקורי.

ניתן לחזור לנקודות ספציפיות במהלך העבודה.

וכו....

**פתיחת ענף/Branch ועבודה עליו-**

**1.** **git checkout -b <branch-name>**-

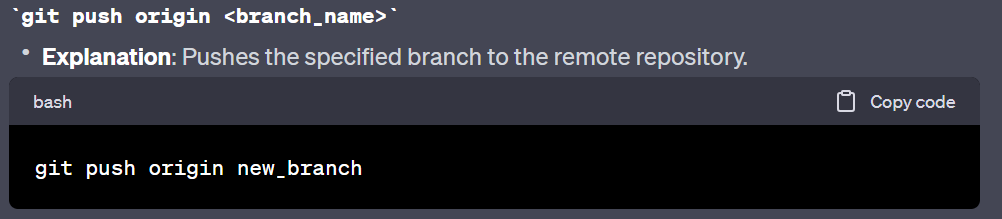


כאשר אני על התיקייה שמקושרת לGit, אני משתמש בפקודה הזאת. השם אמור להופיע עם מכפים ומרכאה בהתחלה ובסוף. דוג' git checkout -b 'new-feature' (אולי לא חייב מירכאות). כלומר הפקודה פלוס הכותרת שבחרנו לענף הזה.

(כרגע זה רק אצלי לוקאלית על המחשב כי עוד לא "דחפתי")

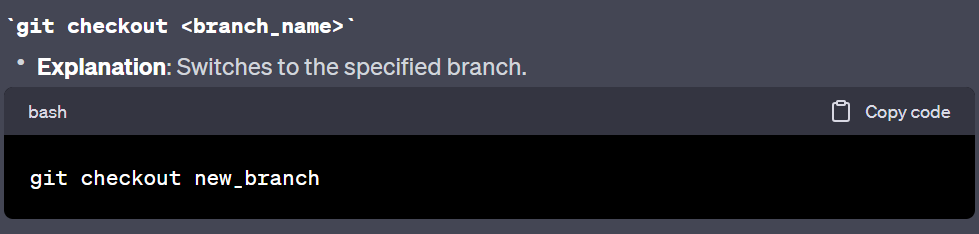
**הערה-** כעת הפעולות שאני מבצע יהיו רק על הענף שבחרתי במקביל לשאר הענפים. הmain הוא הענף הראשי.

**2**.**git push origin <branch-name>** - כעת יש לנו מספר ענפים שאנחנו צריכים לבחור איזה מהם "לדחוף" לכן כעת כשאנו מוסיפים את השינויים נצטרך להשתמש בפקודה הזאת בסוף התהליך במקום פוש רגיל.



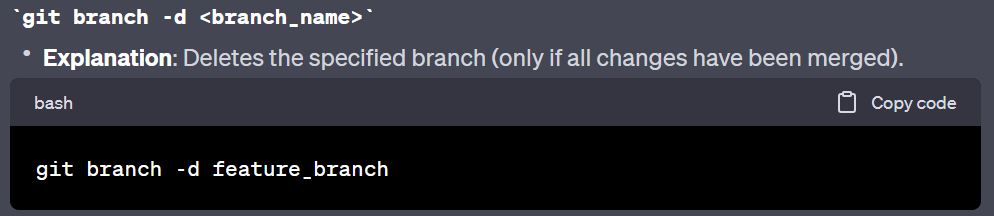
(שים לב ש origin זה שם הענף שאתה נמצא עליו ואליו אתה רוצה לדחוף, כלומר אפשר לשחק עם זה. צד שמאל שם הענף שאתה נמצא עליו ואליו אתה רוצה לדחוף, ואילו בצד ימין זה שם הענף שממנו אתה לוקח את השינויים)

3. **git checkout <main>** - מעבר בין ענפים קיימים, מה שבסוגריים זה פשוט השם של הענף שאליו את רוצה לעבור (מיין זה הענף הראשי)

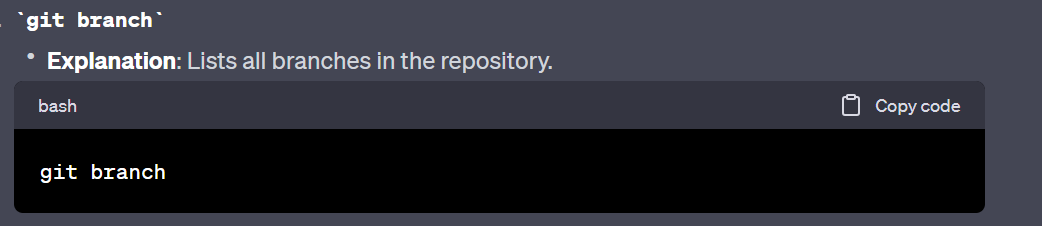


4.**git branch -d <branch-name>** - מחיקת ענף, בסוף פשוט לבחור את השם של הענף שאותו אני רוצה למחוק.

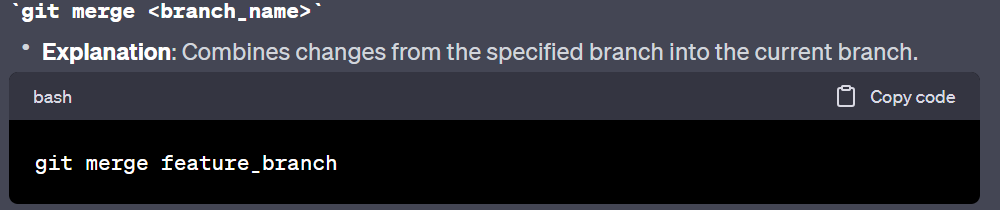
(הכל לוקאלי עד שאני עושה פוש)



5. **git branch**- מציג את רשימת הענפים שיש לי



6. **git merge <branch-name>** - שילוב של הענפים.



קודם כל אני אמור להיות בענף שאליו אני רוצה לאחד את השינויים, לאחר מכן אני משתמש בפקודה הזאת ובסופה את שם הענף הספציפי שאני רוצה לקחת לשם האיחוד

לדוג', אני נמצא על הענף הראשי ואליו אני רוצה להוסיף את new-branch

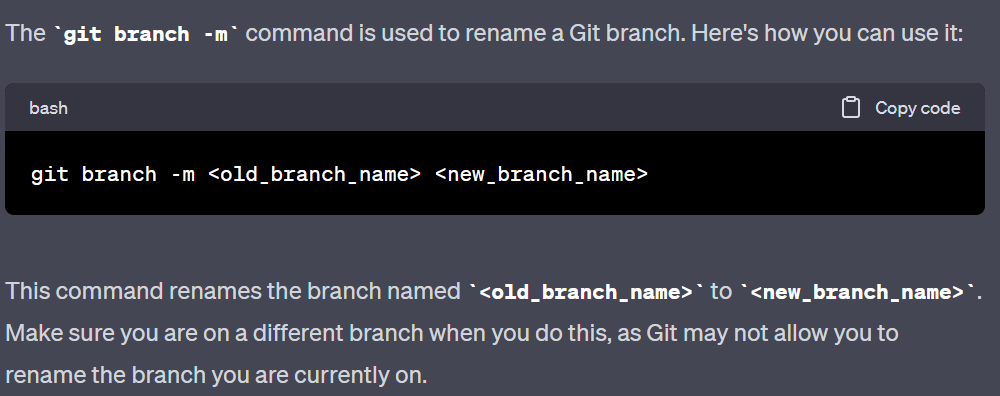
git merge new-branch

(שוב, הכל לוקאלי עד שאני עושה פוש)

כשאתה עובד בצוות שים לב שאם אתה משתמש בפקודה הזאת אז אתה צריך לראות שלא יהיו לך התנגשות עם ערבוב של כמה שינויים במקביל.

**git branch -m <old-branch-name> <new-branch-name> .7**

משנה שם של branch קיים מ-<old-branch-name> ל-<new>



8. **git stash**- שומר באופן זמני שינויים בספריית העבודה שאינם מוכנים להתחייב, ומאפשר לך להחליף סניפים מבלי לבצע עבודה לא גמורה.

**Pull/Fetch –**

שתי הפעולות האלו נועדו לשמירה/הורדה של עדכונים מרחוק. כלומר פעולות לעבודה רבת משתמשים תוך עדכון השינויים בrepository המקומי שלי.

קודם על אני כמובן צריך להיות מחובר לפרויקט עם git clone.

\*\*\*ברמה הלוקאלית של הCMD-

**git fetch/git fetch <name-of-repo>**- כאשר אני מבצע את הפקודה הזאת (אחרי שהעתקתי את הפרויקט) אז אני בעצם לוקח את השינויים שביצעו המתכנתים האחרים בפרוייקט ומעתיק אותם למכונה הוירטואלית שלי (cmd), הם עדיין לא נמצאים בתיקייה שלי לוקאלית על המחשב, שימושי לקבלת סקירה כללית של שינויים \*לפני\* החלטה של מיזוג.

השם בסוגריים לצורך העניין יהיה origin במידה וזה המקור או ענף כלשהו אם זה מה שאני רוצה למשוך אבל גם ללא השם הפעולה תתבצע.

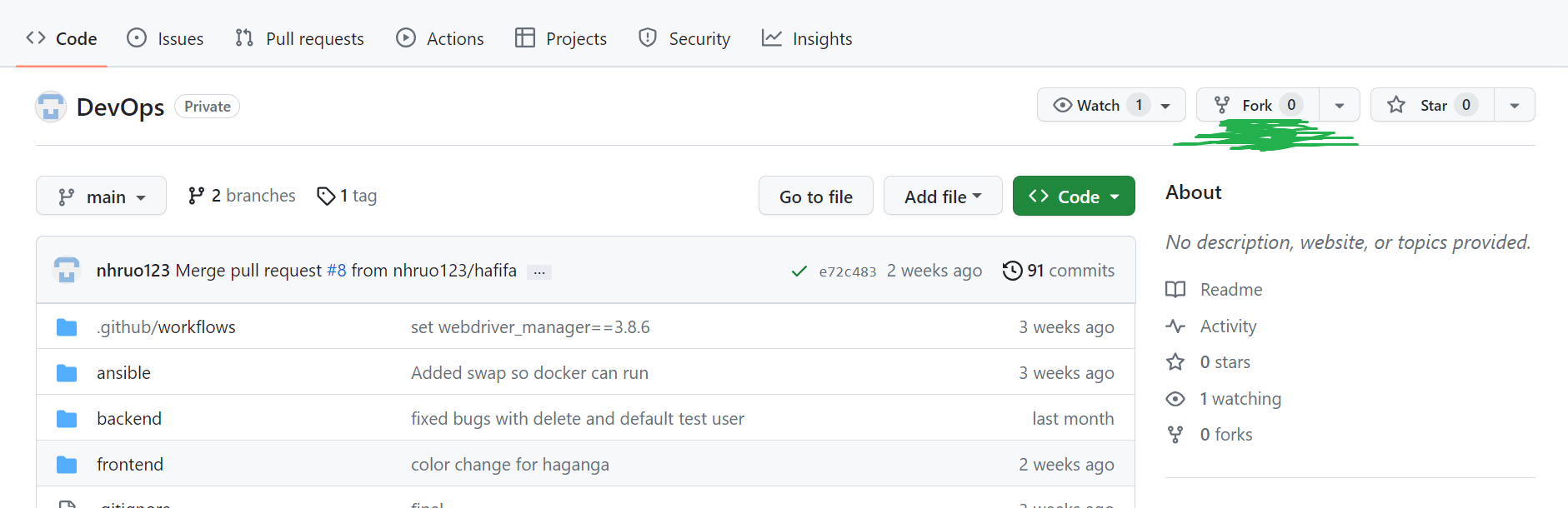
**git pull/ git pull <remote> <branch>**- משיכה של השינויים והשמה של המשיכה בתיקייה המקומית שלי במחשב וגם בrepository . כך קיימים לי כל השינויים ואני יכול להמשיך לעבוד משם.

הסבר על משיכה מדויקת יותר ,

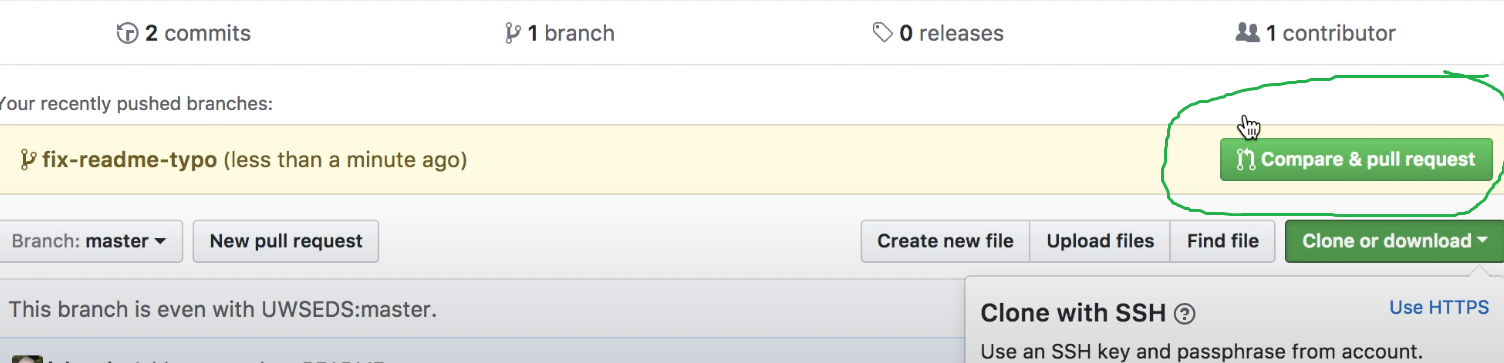
remote זה השם של ה repository המרוחק שממנו אנחנו ניקח את השינויים

ואילו branch זה השם של הענף הספציפי בתוך הrepository שאותו נרצה למשוך ולשלב.

הערה: כאשר אני רוצה להעתיק איזה שהוא repository שראיתי בgithub אני יכול ליצור העתק שלו דרך הפקודה fork



עכשיו כאשר אני מבצע שינויים ומעלה אותם, בין אם זה קבצים קיימים שפשוט עדכנתי ובין אם זה יצירת ענף חדש שאותו הוספתי, תיפתח לי אופציה שתשאל אותי אם אני רוצה ליצור pull request



כשאני לוחץ על זה, זה יפתח לי חלון שבו למעשה אני ארצה לשלב את השנויים שביצעתי ולהכניס אותם לגרסה הראשית/מאסטר, אני ארשום תיאור קצר שבוא אני אתאר אילו שינויים ביצעתי או כל פירוט שנראה לי שדרוש מפני שמי שמקבל את הבקשה הוא למעשה בעל הrepository והוא זה שצריך לנהל את כל השינויים והכל אמור לעבור דרכו בין אם זה אני או מישהו אחר שמנהל.