# GIT - מערכת ניהול גרסאות עבור קוד

### GIT - מערכת ניהול גרסאות עבור קוד

GIT היא מערכת לניהול גרסאות המיועדת לניהול ושמירת שינויים בקוד המקור של פרויקט. באמצעות GIT ניתן לשמור את היסטוריית השינויים, לעקוב אחר התפתחות הפרויקט ולעבוד בשיתוף פעולה עם מתכנתים אחרים.

### מושגי יסוד ב-GIT

\*\*Repository (מאגר):\*\* מאגר שבו נשמר הקוד וכל השינויים שנעשו בו.

\*\*Local Repository (מאגר מקומי):\*\* מאגר שנמצא במחשב המקומי שלך.

\*\*Remote Repository (מאגר מרוחק):\*\* מאגר שנמצא בשרת מרוחק, כמו GitHub, GitLab או BitBucket.

\*\*Commit (התחייבות):\*\* פעולה שמבצעת שמירת שינוי בקוד במאגר.

\*\*Branch (סניף):\*\* עותק עצמאי של הקוד שבו ניתן לבצע שינויים מבלי להשפיע על הסניף הראשי.

\*\*Merge (מיזוג):\*\* פעולה שמבצעת איחוד של שינויים מסניף אחד לאחר.

\*\*Stage (במה):\*\* אזור ביניים שבו נשמרים הקבצים לפני ביצוע ה-commit.

### פקודות בסיסיות ב-GIT

- `git --version` - מציג את גרסת GIT המותקנת.

- `git config --global user.name "USERNAME"` - הגדרת שם משתמש.

- `git config --global user.email "EMAIL"` - הגדרת כתובת דוא"ל.

- `git init` - אתחול מאגר GIT חדש.

- `git status` - הצגת מצב הקבצים (שינויים ב-stage ולא ב-stage).

- `git add FILENAME` - הוספת קובץ ל-stage.

- `git add .` - הוספת כל הקבצים והתיקיות ל-stage.

- `git commit -m "MESSAGE"` - ביצוע commit עם הודעה.

- `git log` - הצגת היסטוריית ה-commit-ים.

- `git rm --cached FILENAME` - הסרת קובץ מה-stage.

- `git branch` - הצגת כל הסניפים.

- `git checkout BRANCH\_NAME` - מעבר לסניף מסוים.

- `git merge BRANCH\_NAME` - מיזוג סניף מסוים עם הסניף הנוכחי.

- `git remote add origin URL` - הוספת מאגר מרוחק.

- `git push -u origin master` - דחיפת השינויים למאגר המרוחק.

### דוגמת עבודה עם GIT

1. \*\*אתחול מאגר חדש:\*\*

```sh

git init

```

2. \*\*הוספת קבצים ל-stage וביצוע commit ראשון:\*\*

```sh

git add .

git commit -m "First commit"

```

3. \*\*הגדרת מאגר מרוחק ודחיפת השינויים:\*\*

```sh

git remote add origin https://github.com/USERNAME/REPOSITORY\_NAME.git

git push -u origin master

```

4. \*\*ביצוע שינויים נוספים וביצוע commit נוסף:\*\*

```sh

git add .

git commit -m "Second commit"

git push

```

### סיכום

GIT היא כלי חיוני לכל מתכנת, המאפשר ניהול גרסאות יעיל, עבודה בשיתוף פעולה ושמירה על היסטוריית השינויים בקוד. באמצעות הפקודות הבסיסיות והכלים המוצעים על ידי GIT ניתן לנהל את הפרויקט בצורה מסודרת ולוודא שכל שינוי בקוד נשמר ומתועד.