

מבוא לGit ולבקרת גרסאות

מהי בקרת גרסאות?

- ניהול שינויים בקוד לאורך זמן
- מאפשר חזרה לגרסאות קודמות
- תומך בעבודה משותפת על פרויקטים
- נמנע מאובדן מידע חשוב

למה צריך בקרת גרסאות?

- מאפשר שמירה על היסטוריית שינויים
- עבודה משותפת עם מפתחים אחרים
- עבודה על כמה גרסאות במקביל
- במקרה של שינוי קוד שיצר בעיה - החזרה למצב שקדם לשינוי בקלות
- ניהול גרסאות בפרויקט גדול

מה זה Git?

- מערכת לבקרת גרסאות מבוזרת
- שומר היסטוריית שינויים מקומית
- מאפשר עבודה במקביל על קבצים ופתרון קונפליקטים
- פותח ע"י לינוס טורבאלדס

Git לעומת GitHub

Git

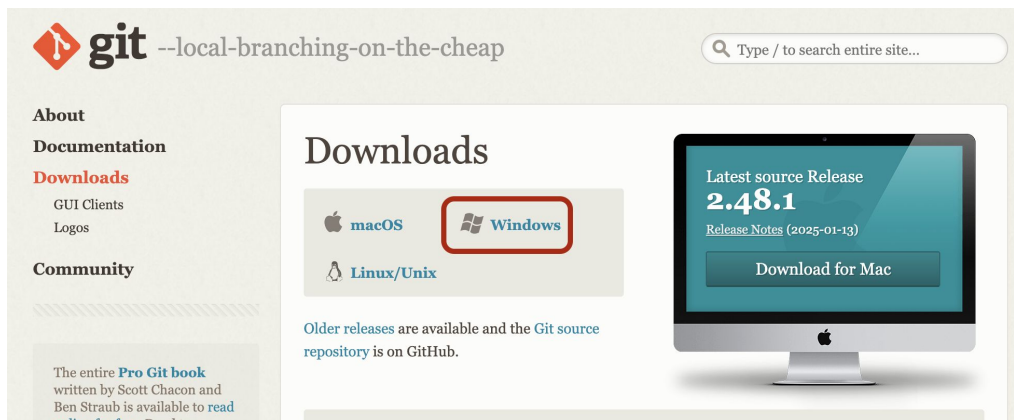
- מערכת בקרת גרסאות
- מקומי על המחשב

GitHub

- פלטפורמה לאירוח קוד ובקרת גרסאות.
- בענן, נגישה מכל מקום
- תומכת בשיתוף

התקנת Git Windows

ניתן להוריד מאתר git-scm.com
ניכנס לקישור <https://git-scm.com/downloads>



התקנת Git

Mac

- ב-Mac: התקנה דרך ``brew install git``

הגדרת Git

```
git config --global user.name "השם שלך"  
git config --global user.email "האימייל שלך"  
git config --list
```

• נסו בעצמכם

יצירת מאגר (repo) חדש

```
mkdir myTestRepo1  
cd myTestRepo1  
git init
```

- שימו לב - שם הספרייה שאתם יוצרים יהיה שם הפרויקט. במקרה שלנו - myTestRepo1
בהמשך נמשיך ונתיחס לפרויקט בשם שיצרנו כאן.
- שימו לב - את כל פקודות ה-git נריץ כאשר אנו בתוך הספרייה
בה הוגדר ה-repo

```
> workspace mkdir myTestRepo1  
> workspace cd myTestRepo1  
> myTestRepo1 git init  
Initialized empty Git repository in /Users/ronenshachar/workspace/myTestRepo1/.git/  
> myTestRepo1 git:(main)
```

סטטוס המאגר

`git status`

```
→ workspace mkdir myTestRepo1
→ workspace cd myTestRepo1
→ myTestRepo1 git init
Initialized empty Git repository in /Users/ronenshachar/workspace/myTestRepo1/.git/
→ myTestRepo1 git:(main) git status
On branch main

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

הוספת שינויים למאגר

- הוספת הקובץ `myFile.txt`

```
git add myFile.txt
```

- הוספת כל הקבצים בספריה הנוכחית וזו שתחתיה
- הנקודה בסוף הפקודה מציינת ספריה נוכחית
ומטה.

```
git add .
```

```
> myTestRepo1 git:(main) git status
On branch main

No commits yet

nothing to commit (create/copy files and use "git add" to track)
> myTestRepo1 git:(main) echo Hello > file2.txt
> myTestRepo1 git:(main) × git add .
> myTestRepo1 git:(main) × git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   file2.txt
```

פקודת commit

- פקודת ה-commit מקבעת את השינויים שהוספנו ב
git add -
- כל השינויים שהוספנו ישמרו כיחידה אחת עם
מזהה יחודי (sha1), ופרטים נוספים

"הודעה שמתארת את השינוי" -m git commit

```
> myTestRepo1 git:(main) ✕ git commit -m 'Adding file'
[main (root-commit) b030007] Adding file
1 file changed, 1 insertion(+)
create mode 100644 file2.txt
```

הצגת היסטוריית commits

`git log`

```
commit b0300078aec04b5184894d67bb0164bde9fff4f4 (HEAD -> main)
```

```
Author: Ronen Shachar <ronen.shachar@gmail.com>
```

```
Date: Sun Feb 9 00:42:19 2025 +0200
```

```
    Adding file
```

```
(END)
```

סיכום

- בקרת גרסאות שומרת על היסטוריית שינויים
- Git מאפשר ניהול גרסאות מקומי
- ניתן להוסיף, לשמור ולבטל שינויים
- ה-commits יוצרים היסטוריה מסודרת
- `git log` מציג את כל ה-commits

תרגול

- 1. התקינו את Git במחשב שלכם
- 2. הגדירו שם ואימייל באמצעות ``git config``
- 3. צרו תיקייה חדשה, הוסיפו בה קובץ והריצו ``git``
``init``
- 4. בצעו שינויים, השתמשו
ב``git add`` ו``git commit``
- 5. הציגו את היסטוריית ה-commits עם
``git log``