

GitHub ועבודה עם Git מרחוק

מה זה GitHub?

- פלטפורמה לניהול ושיתוף קוד
- מאפשר עבודה משותפת על פרויקטים
- מאחסן קוד בענן
- תומך בשיתוף ושיתוף פעולה על קוד

Git מול GitHub

Git

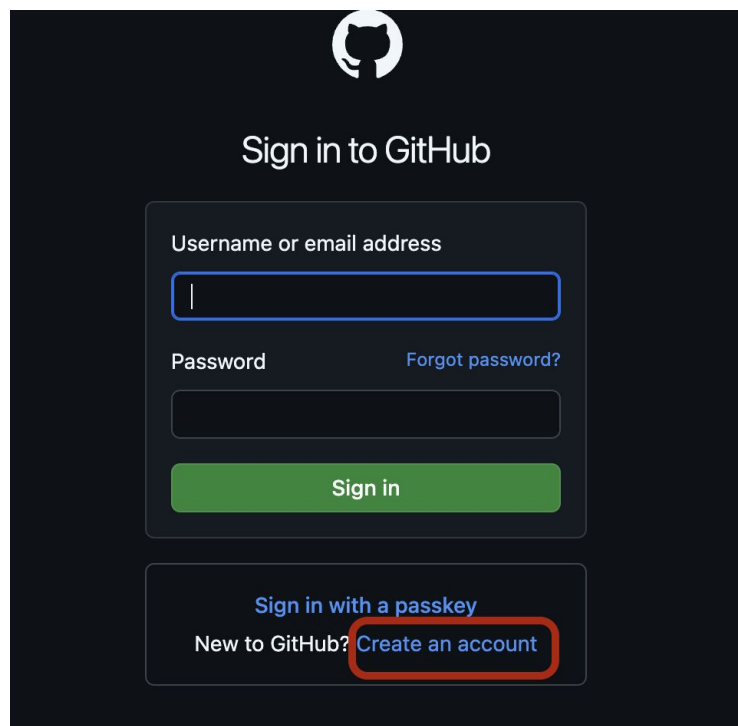
- מערכת בקרת גרסאות מקומית
- פועלת על המחשב האישי

GitHub

- פלטפורמה לשיתוף קוד בענן
- מאפשר ניהול ושיתוף פעולה בפרויקטים

יצירת חשבון ב-GitHub

- התחברו ל-GitHub
- בחרו "Create an account"



The image shows the GitHub sign-in interface on a dark background. At the top is the GitHub logo. Below it, the text "Sign in to GitHub" is centered. The main form contains two input fields: "Username or email address" and "Password". Below the password field is a "Forgot password?" link. A green "Sign in" button is positioned below the password field. At the bottom of the form, there is a section with the text "Sign in with a passkey" and "New to GitHub?". The "Create an account" link is highlighted with a red rectangle.

Sign in to GitHub

Username or email address

Password [Forgot password?](#)

Sign in

Sign in with a passkey

New to GitHub? [Create an account](#)

יצירת מאגר חדש ב-GitHub

- התחברו ל-GitHub
- לחצו על `New Repository`
- במסך הבא תתבקשו לתת שם - העתיקו את שם הספרייה בה יצרתם את הפרויקט שלכם בגיט.
זכרו כי בשקפים הקודמים התייחסנו לשם הספרייה כשם הפרויקט. בדוגמה שנתנו - myTestRepo1

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

Owner *



ronen-shachar

Repository name *

myTestRepo1

1 - בחרו שם לrepo

✔ myTestRepo1 is available.

Great repository names are short and memorable. Need inspiration? How about [stunning-train](#) ?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:



Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)



You are creating a public repository in your personal account.

2 - צרו את ה-repo

Create repository


חיבור מאגר Git מקומי ל-GitHub


- נעבור לספרית הפרויקט, בה הרצנו `git init`
- נריץ את הפקודות הנ"ל
- כאשר ניצור `repo` חדש בגיטהאב נקבל את הפקודות המדויקות


```
git remote add origin <repo-url>
```


```
git push -u origin main
```


העתיקו את הפקודות כפי שיופיעו אצלכם


 **myTestRepo1** Public


 Pin

 Unwatch 1


 Fork 0

 Star 0

 **Start coding with Codespaces**
Add a README file and start coding in a secure, configurable, and dedicated development environment.
[Create a codespace](#)

 **Add collaborators to this repository**
Search for people using their GitHub username or email address.
[Invite collaborators](#)

Quick setup — if you've done this kind of thing before

 Set up in Desktop

 or

HTTPS

SSH

<https://github.com/ronen-shachar/myTestRepo1.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# myTestRepo1" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/ronen-shachar/myTestRepo1.git
git push -u origin main
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/ronen-shachar/myTestRepo1.git
git branch -M main
git push -u origin main
```

הריצו את הפקודות הבאות

שכפול מאגר קיים (Clone)

- שכפול repo - הורדת repo מגיטהב למיקום חדש
 - לסיפריה נוספת במחשב שלנו
 - למחשב אחר לצורך עבודה משותפת

```
git clone <repo-url>
```

משיכת שינויים (pull)

```
git pull origin main
```

- משיכה של כל השינויים שעודכנו בגיטהאב לrepo המקומי שלנו.
זכור שניתן להריץ git pull אך ורק אם אנו נמצאים בספריה בה הוגדר ה-repo.

דחיפת שינויים (push)

- לאחר ביצוע שינויים בענף המקומי, יש לדחוף אותם (push) ל-GitHub כדי לשתף אותם עם הצוות או לשמור אותם בענן.

`git push`

הדגמה - clone

- נבצע clone של repo מגיטהב.
- בדיקת הסטטוס מראה שאין שינויים במחשב המקומי - לא עשינו שום שינוי אחרי הclone

```
➤ workspace git clone https://github.com/ronen-shachar/myTestRepo1.git
Cloning into 'myTestRepo1'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
➤ workspace cd myTestRepo1
➤ myTestRepo1 git:(main) git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

הדגמה - הוספת קובץ

- ניצור קובץ חדש בשם file1.txt
- בדיקת הסטטוס מראה שיש קובץ חדש שעדיין לא הוסף (add) לגיט

```
> myTestRepo1 git:(main) echo Hello world > file1.txt
> myTestRepo1 git:(main) ✕ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   file1.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

הדגמה - commit

- נבצע commit
- בדיקת הסטטוס מראה שהענף המקומי main (הסבר על ענפים בהמשך) מכיל commit אחד בהשוואה לגיטהאב - origin/main

Your branch is ahead of 'origin/main' by 1 commit.

```
> myTestRepo1 git:(main) ✕ git commit -m "adding files"
[main d61d843] adding files
 1 file changed, 1 insertion(+), 1 deletion(-)
> myTestRepo1 git:(main) git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

הדגמה - push

- נבצע push
- בדיקת הסטטוס מראה שתוכן הענף המקומי main עודכן לגיטהאב - origin/main

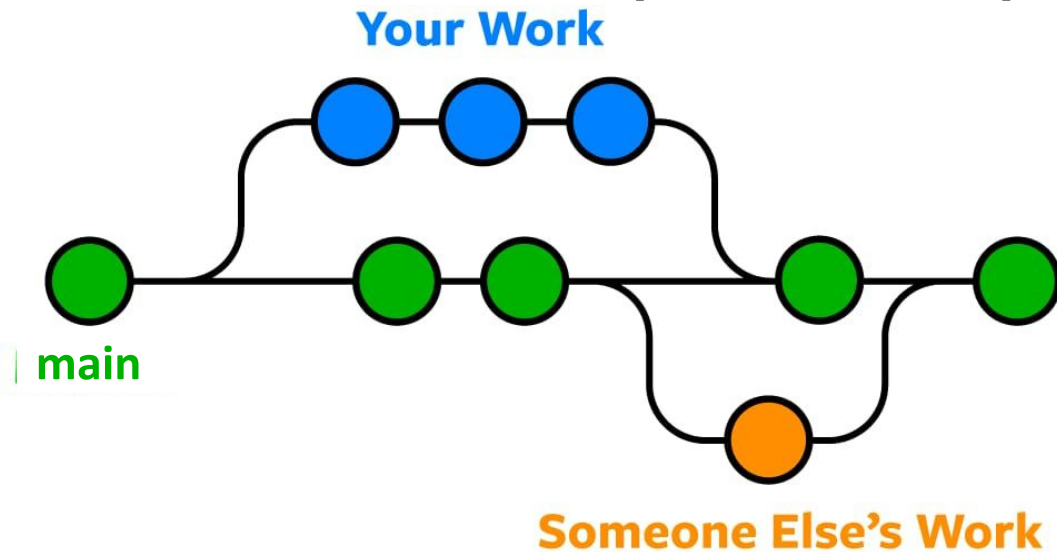
Your branch is up to date with 'origin/main'.

```
➤ myTestRepo1 git:(main) git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Writing objects: 100% (3/3), 257 bytes | 257.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/ronen-shachar/myTestRepo1.git
   897dce0..d61d843  main -> main
➤ myTestRepo1 git:(main) git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
```

ניהול ענפים (Branches)

- ענפים מאפשרים עבודה על פיצ'רים חדשים ללא השפעה על הקוד הראשי
- ענפים מאפשרים לכמה מפתחים לעבוד במקביל על הקוד
- כברירת מחדל הענף הראשי נקרא **main**



ניהול ענפים (Branches)

- יוצר ענף חדש, אך לא עובר אליו

```
git branch <branch-name>
```

- עובר לענף קיים

```
git checkout <branch-name>
```

- יוצר ענף חדש וגם עובר אליו

```
git checkout -b <new-branch>
```

מיזוג ענפים (Merge)

- מיזוג קוד מענף משני לענף הראשי - אחרי שסיימנו לעבוד על השינויים בענף נפרד, נמזג אותם לענף הראשי

`git pull`

`git checkout main`

`git merge <branch-name>`

טיפול בקונפליקטים במיזוג

- קונפליקטים קורים כאשר יש שינויים סותרים באותו קובץ
- אנו עלולים להתקל בקונפליקטים בהמשך כאשר נעבוד במשותף על הקוד
- יש לערוך את הקובץ ידנית ולפתור את הקונפליקט
- הערה: בשלב זה כאשר יש קונפליקטים הדרך הפשוטה ביותר לטיפול היא ביצוע clone לספריה חדשה והעתקת השינויים הנדרשים אליה ידנית.

סיכום

- GitHub מאפשר עבודה משותפת בענן
- ניתן לשכפל מאגרים, למשוך שינויים ולדחוף עדכונים
- עבודה עם ענפים משפרת את ניהול הפיתוח

תרגול

1. צרו חשבון ב-GitHub
2. צרו מאגר חדש והעלו אליו קוד
3. שכפלו את המאגר שהעלתם אל ספריה אחרת שתצרו במחשב שלכם
4. צרו ענף חדש, עברו לענף החדש והוסיפו לענף קובץ חדש