

תוכנה 1

סביבת הפיתוח

מנהלות

אתר הקורס: ■

<http://courses.cs.tau.ac.il/software1/2122b/>

מרצה: לנה דנקין ■

מתרגלים: ■

אלה גולדשמידט ■

אמיר הרץ ■

סביבת המחשוב באוניברסיטה היא Linux ■

תנאי קדם: פתיחת חשבון אישי במחשבי האוניברסיטה ■

הנחיות לפתיחת חשבון והכרת סביבת העבודה באתר הקורס. ■

סביבת פיתוח והרצה ל-Java

■ גרסת ה-Java שעמה נעבוד:

Java SE (Standard Edition) 8.0

■ חבילת סביבת ההרצה:

JRE (Java Runtime Environment) that includes:

- JVM (Java Virtual Machine)
- Standard Class Library

■ חבילת ערכת הפיתוח:

JDK (Java Development Kit) that includes:

- JRE
- Command line tools: compiler, debugger etc.

■ הורדה ותיעוד ב-

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

סביבת פיתוח שלובה

IDE = Integrated Development Environment ■

סביבה המשלבת רכיבי/כלי פיתוח עצמאיים: ■

עורך טקסט (editor) ■

סייר הקבצים (browser) ■

מהדר (compiler) ■

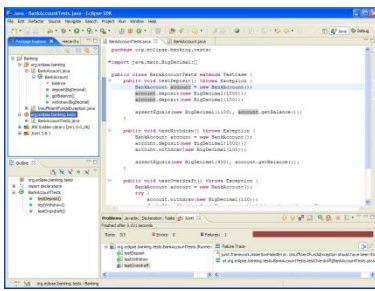
סביבת זמן ריצה (JRE) ■

מנפה השגיאות (debugger) ■

ועוד... ■

Eclipse – ה- IDE בו נשתמש בקורס. ■

Eclipse



- IDE המתאים גם לפיתוח תוכנה ב Java.
- ניתן להתקנה ב- Linux, Windows ועוד
- דורש התקנה בנפרד של JDK
- אתר הבית: www.eclipse.org
- הוראות התקנה ושימוש – באתר הקורס.
- מותקן על כל המחשבים בכיתת המחשבים בשרייבר.

Git vs. GitHub

- Git היא תוכנת ניהול גרסאות של קוד.
 - מותקנת על המחשב האישי שלכם.
- GitHub הוא שרות עבור פרויקטים פיתוח תוכנה שבהם משתמשים במערכת Git.
 - מספק אחסון למערכת ניהול הגרסאות Git.
 - משמש כרשת חברתית בה מפתחים משתפים את סטטוס ההתקדמות שלהם בכתיבת הקוד.
 - אחרים יכולים להגיב \ לערוך \ ולתקן קוד פומבי שעלה ל-GitHub.

מה מאפשר לנו Git

ניהול גרסאות של הקוד: ■

■ מעקב אחרי שינויים.

■ חזרה אחורה.

■ גיבוי

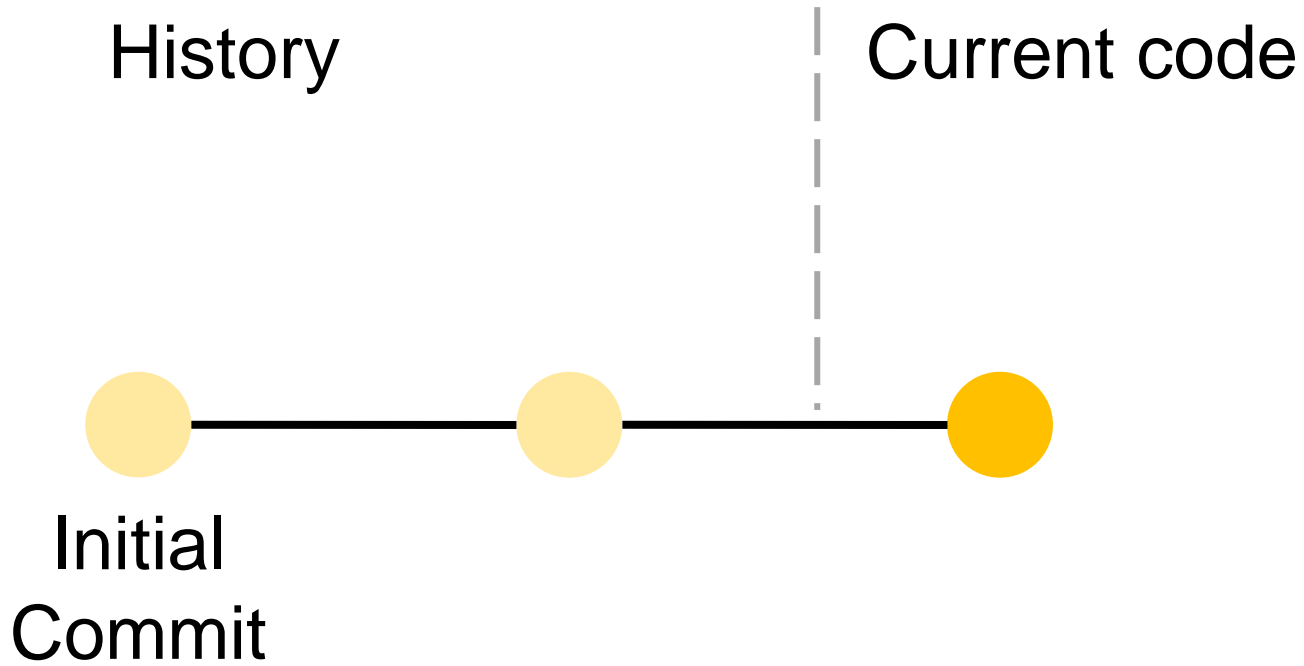
עבודה בצוות: ■

■ עדכון של שינויים שבוצעו ע"י חברי צוות אחרים.

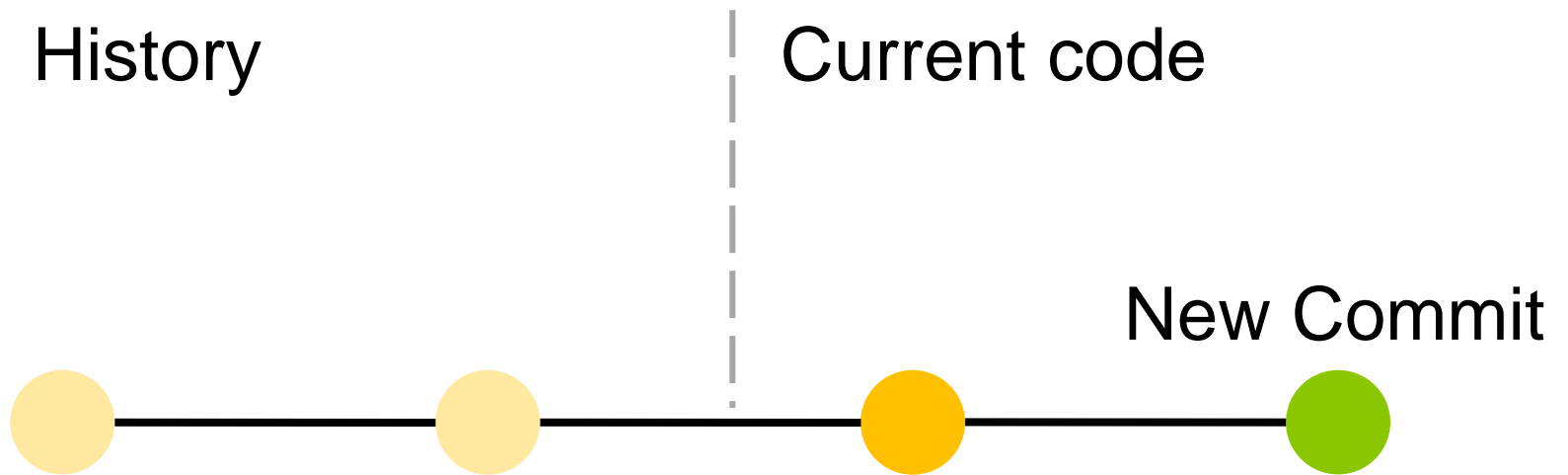
■ מיזוג (אוטומטי או ידני) במידה ושני חברי צוות משנים את אותו הקוד.

■ זו שיטת העבודה המקובלת (ואם לא Git, אז כלי בקרת תצורה אחר עם מאפיינים דומים)

Git



Git add + commit



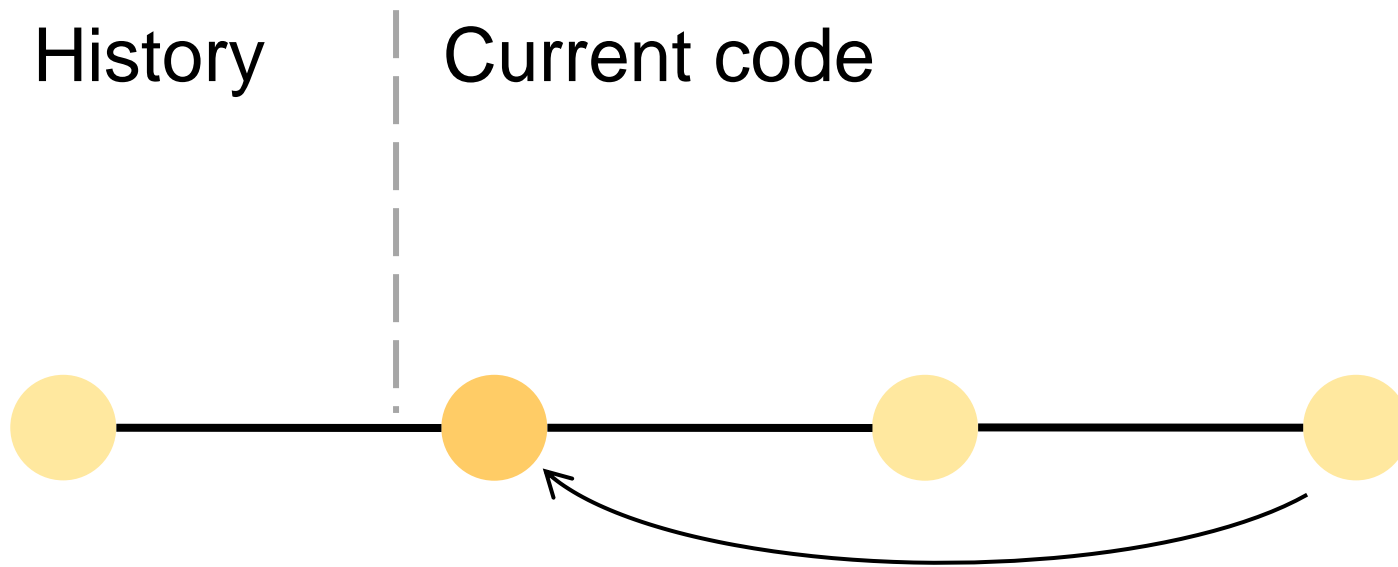
Git commit

History

Current code



Go back



Git and GitHub

My computer



HW template

Remote (Github)

Git clone

My computer



clone

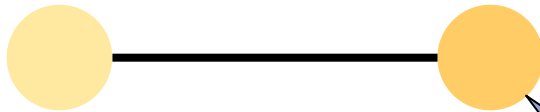


HW template

Remote (Github)

Git add + commit

My computer



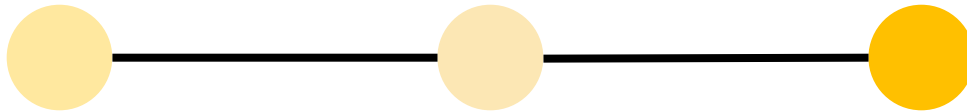
Commit:
סיימנו את
שאלה 1

HW template

Remote (Github)

Git add + commit

My computer



HW template

Remote (Github)

Git add + commit

My computer



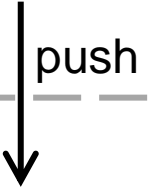
התרגיל מוכן
להגשה!

HW template

Remote (Github)

Git push

My computer



Remote (Github)



סיימנו?

Git push

Nova



נקמפל ונריץ את
הקוד על nova

clone

Remote (Github)



איך מתחילים?

- פותחים חשבון ב-GitHub (שיכיל את ה-remote repositories שלנו).
- ניתן לפתוח חשבון חינומי עם כל כתובת מייל, אבל חשבון זה מאפשר לכם לפתוח מספר מוגבל של repositories פרטיים.
- הרשמה עם כתובת המייל האוניברסיטאית תאפשר לכם לפתוח repositories נראות פומבית או פרטית.

איך מתחילים?

■ התקנת git על המחשב שלכם. ניתן להיעזר במדריך הבא:

<https://github.com/git-guides/install-git>

■ ניתן לעבוד עם git בכמה אופנים:

■ Command line (ב Windows עם Git Bash).

■ מנשק משתמש גרפי.

■ שילוב של Git בתוך כלי הפיתוח (IDE) שלכם (Eclipse \ IntelliJ).

איך מתחילים?

■ לפתוח פרויקט חדש או לשכפל פרויקט קיים?

■ כדי לשכפל פרויקט קיים נשתמש בפקודה:

```
git clone <repo> [<dir>]
```

■ כדי לפתוח פרויקט חדש וריק ב GitHub:

<https://guides.github.com/activities/hello-world/>

איך מתחילים?

■ הוספת SSH-key לחשבון ה-GitHub.

■ מאוגוסט 2021, GitHub לא מאפשרת שימוש בסיסמא על מנת לבצע אימות של פעולות Git מול GitHub*.

■ במקום זאת יש להשתמש ב-access token או SSH-key.

[Token authentication requirements for Git operations](#)*

איך מתחילים?

- יצירת SSH-key לחשבון ה-GitHub.
- מדריך ליצירת SSH-key:

Generating a new SSH key

Generating a new SSH key and adding it to the ssh-agent

After you've checked for existing SSH keys, you can generate a new SSH key to use for authentication, then add it to the ssh-agent.

Mac **Windows** Linux

About SSH key generation

If you don't already have an SSH key, you must generate a new SSH key to use for authentication. If you're unsure whether you already have an SSH key, you can check for existing keys. For more information, see ["Checking for existing SSH keys."](#)

If you want to use a hardware security key to authenticate to GitHub, you must generate a new SSH key for your hardware security key. You must connect your hardware security key to your computer when you authenticate with the key pair. For more information, see the [OpenSSH 8.2 release notes](#).

If you don't want to reenter your passphrase every time you use your SSH key, you can add your key

איך מתחילים?

- יצירת SSH-key לחשבון ה-GitHub.
- מדריך ליצירת SSH-key:

Generating a new SSH key

- 2 Add your SSH private key to the ssh-agent. If you created your key with a different name, or if you are adding an existing key that has a different name, replace *id_ed25519* in the command with the name of your private key file.

```
$ ssh-add ~/.ssh/id_ed25519
```

- 3 Add the SSH key to your account on GitHub. For more information, see "[Adding a new SSH key to your GitHub account.](#)"

איך מתחילים?

הוספת SSH-key לחשבון ה-GitHub. ■

Adding a new SSH key

Adding a new SSH key to your GitHub account

To configure your GitHub account to use your new (or existing) SSH key, you'll also need to add it to your GitHub account.

Mac Windows Linux

Before adding a new SSH key to your GitHub account, you should have:

- [Checking for existing SSH keys](#)
- [Generating a new SSH key and adding it to the ssh-agent](#)

After adding a new SSH key to your GitHub account, you can reconfigure any local repositories to use SSH. For more information, see "[Switching remote URLs from HTTPS to SSH.](#)"

Note: GitHub is improving security by dropping older, insecure key types.

DSA keys (`ssh-dss`) are no longer supported. Existing keys will continue to function through March 15, 2022. You cannot add new DSA keys to your user account on GitHub.

RSA keys (`ssh-rsa`) with a `valid_after` before November 2, 2021 may continue to use any signature algorithm. RSA keys generated after that date must use a SHA-2 signature algorithm. Some older clients may need to be upgraded in order to use SHA-2 signatures.

דוגמא – תרגיל 0

הגשת תרגילי בית

- בכל תרגיל נפרסם קישור לrepository המתאים.
- עבור תרגיל 0 (לא להגשה):
https://classroom.github.com/a/Qv8o_uv
- ההגשה מתבצעת דרך Git וגם דרך המודל.
- עליכם להתחבר ל GitHub עם המשתמש האוניברסיטאי.



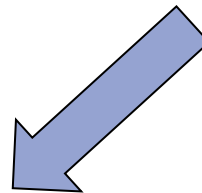
You're ready to go!

You accepted the assignment, **hw0_pilot**.

Your assignment repository has been created:

https://github.com/software1course2122a/hw0_pilot-lenadank

We've configured the repository associated with this assignment ([update](#)).



הגשת תרגילי בית

- הפרויקט יפתח בתצורה של פרויקט קיים.
- לחצו עם הכפתור code (הירוק) והעתיקו את הקישור המצורף ל `git clone`.

The screenshot shows a GitHub repository page for `software1course2122a / hw0_pilot-amirhertz`. The repository is private and has a `main` branch. The file list shows `github-classroom` (Initial commit), `src` (Initial commit), and `.gitignore` (Initial commit). The `Code` dropdown menu is open, showing options: `Clone` (selected), `Open with GitHub Desktop`, and `Download ZIP`. The `Clone` menu shows options for `HTTPS`, `SSH`, and `GitHub CLI`. The repository URL `git@github.com:software1course2122a/hw0_p:` is displayed with a copy icon. A blue arrow points to the copy icon.

הגשת תרגילי בית

Demo