

JOHN BRYCE

תלמדו הייטק. זה עובד!

a *matrix* company

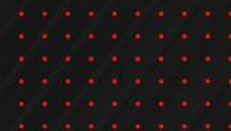
Intro to Computing



מה נלמד?

- 01 | מושגי יסוד
- 02 | איך מחשב עובד?
- 03 | קידוד תכנית - התפתחות לאורך זמן
- 04 | דור שלישי – שפות עיליות
- 05 | שלבים בפתרון בעיה
- 06 | מהו אלגוריתם





01 | מושגי יסוד

- מבנה מערכת מחשב
- חומרה
- תוכנה
- סוגי מערכות הפעלה
- בעלי תפקידים בתעשיית ההייטק





- חומרה
- תוכנה
- מערכות הפעלה

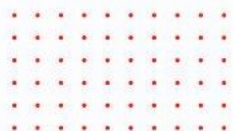


01 | מושגי יסוד | חומרה



ברזלים

- זיכרון עבודה, למשל 8 GB/16GB
- לוח אם
- דיסק קשיח – SSD/HDD נמדד ב TB בד"כ
- כרטיס מסך – הקשר בין הלוח למסך – מכיל מאיץ גרפי בד"כ
- מסך
- מקלדת
- עכבר
- מדפסת



המח

- מערכת ההפעלה

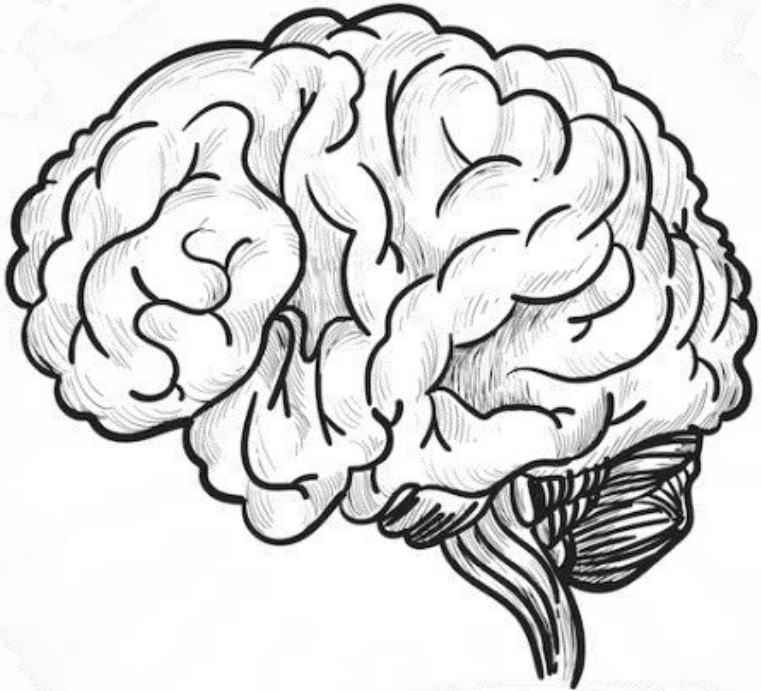
- דפדפן אינטרנט

- אנטי וירוס

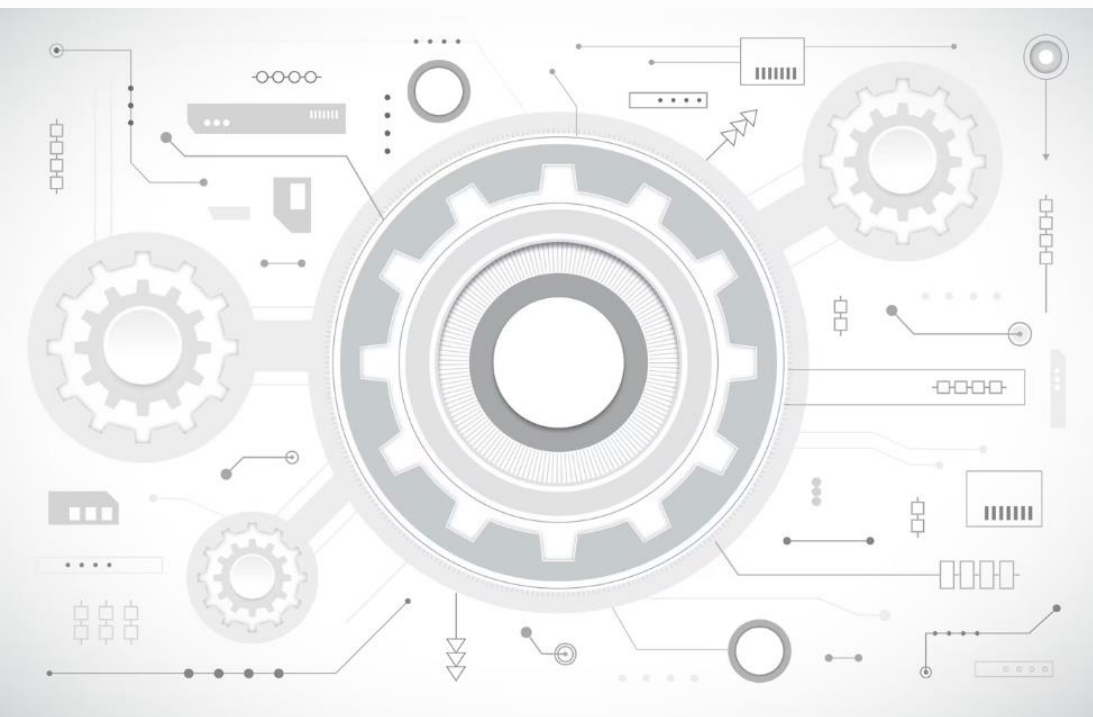
- תוכנת דוא"ל

- מעבד תמלילים

- גיליון אלקטרוני



01 | מושגי יסוד | סוגי מערכות הפעלה



Windows ■

Desktop -

Server -

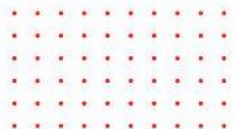
Linux ■

OS/X ■

Mobile ■

IOS -

Android -



01 | מושגי יסוד | בעלי תפקידים בתעשיית ההייטק

■ Windows מנהל רשת/תשתיות – IT Operator

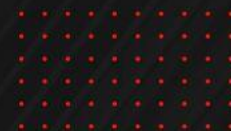
■ מפתחי תוכנה - Software Developer

■ בודקי תוכנה - QA Developer

■ מנהל תשתיות נתונים - DBA

■ DevOps - הדבר החם הנוכחי

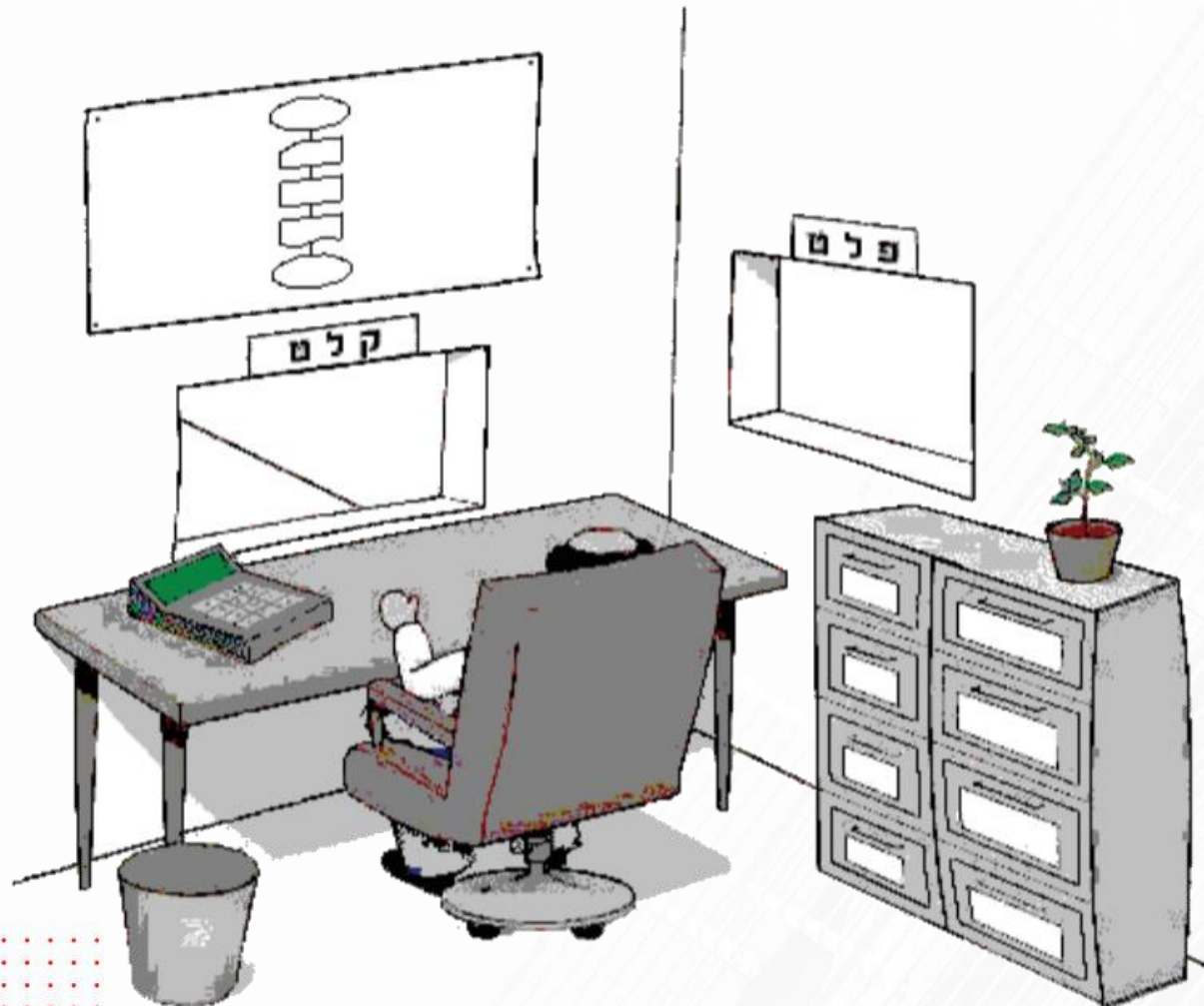




02 | איך מחשב עובד?

- רעיון
- מבנה מחשב
- ארכיטקטורה
- עיבוד נתונים





■ מודל הפקיד

- לוח הוראות

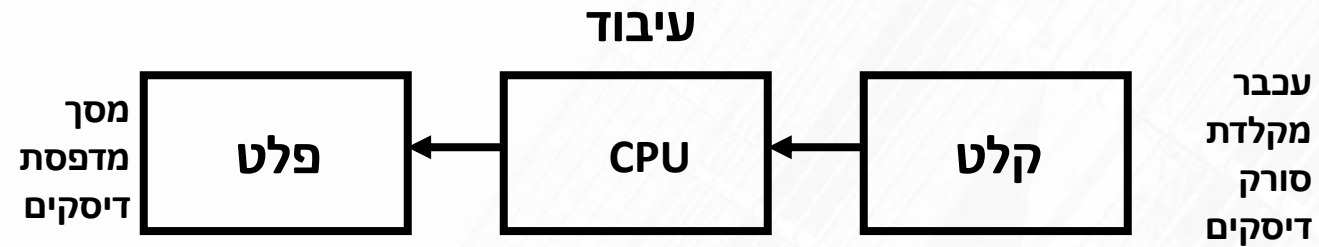
- חלונות קלט/ פלט

- מגירות

- מכונת חישוב

- מכונת כתיבה

02 | איך מחשב עובד | מבנה המחשב



- תפקיד המעבד - ביצוע פעולות.
- סוגי קלט פלט - כרטיסים מחוררים, סרטים, מסך, מקלדת וכו'.

02 | איך מחשב עובד | עיבוד נתונים

דוגמת חישוב הפונקציה: $y=ax+b$

תא עזר

קוד פקודה	כתובת בזכרון
add	16
mul	17
add	18
Store	19

clear

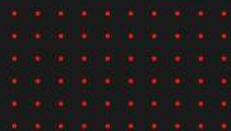
End

משתנים	כתובת
A	16
X	17
b	18
y	19



03 | קידוד תוכנית | התפתחות לאורך זמן

- דור ראשון – שפת מכונה
- דור שני – אסמבלי
- דור שלישי – שפה עילית



03 | קידוד תכנית ← התפתחות לאורך זמן

דור ראשון
שפת מכונה

דור שני
אסמבלי

דור שלישי
שפת עילית



03 | קידוד תכנית ← **דור ראשון**: שפת המכונה

קוד	כתובת
010	00000
011	10000
101	10001
011	10010
111	10011
000	00000



קוד	כתובת
2	0
3	16
5	17
3	18
7	19
0	0

חוסר יעילות, קשה לתכנות ולהבנה

03 | קידוד תכנית ← **דור שני**: אסמבלי

clr

add a

mul x

add b

str y

halt

- תרגום חד חד ערכי

- מתרגם פשוט

- יש צורך לזכור בע"פ

היינו רוצים לכתוב $y=ax+b$

03 | קידוד תכנית ← דור שלישי: שפה עילית

```
int a = 5;
```

```
int x = 6;
```

```
int b = 7;
```

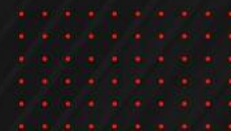
```
int y = a*x + b;
```

```
System.out.println("result : " + y);
```

- קוד קריא

- מתרגם פשוט





04 | דור שלישי | שפות עיליות

- יתרונות השפה העילית
- חסרונות השפה העילית





- את השפות העיליות ניתן להתאים אישית ע"י בניית ספריות

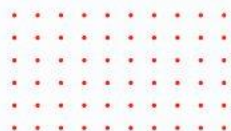
של פונקציות

- השפות משמשות לכל מיני מטרות

- יש גם שפות שמיועדות לנושאים ספציפיים

- SQL

- HTML



04 | קידוד תכנית ← יתרונות השפה העילית



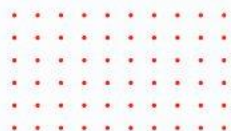
- קל לקריאה, כתיבה, תיקון

- קל לאירוח במחשבים שונים, מערכות הפעלה שונות

- כלים:

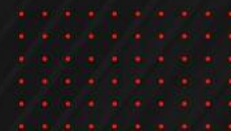
- מהדר – compiler – המרה משפה עילית לשפת מכונה

- Linker, assembler, debugger, profiler , (בהמשך..)



04 | קידוד תכנית ← חסרונות השפה העילית

- צריך קומפיילר (מהדר)
- מהירות – נחשבת איטית מאוד היות והדברים לא נעשים "קרוב לחשמל"



05 | שלבים בפתרון בעיה





■ הגדרת הבעיה באופן מדויק

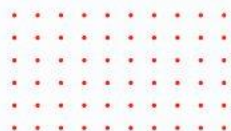
- קלט

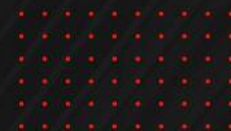
- פלט

■ פיתוח אלגוריתם לפתרון הבעיה (שיטה)

- החלק האינטליגנטי ביותר

■ קידוד (החלק הטכני)





אלגוריתם | 06





הגדרה | אוסף סופי של הוראות שביצוען צעד אחר צעד, מוביל לפתרון בעיה כלשהיא

- סופי בביצוע
- רצף הגיוני (חשיבות לסדר הפעולות)
- עונה על כל האפשרויות
- חד משמעי וברור (אין מקום לשיקול דעת)
- הוראות ברורות ביצוע למכונה

