

CMD בסיסי

הערות צבועות בכחול ופקודות באדום

שימו לב – מומלץ לנסות במחשב שלכם פקודות כמו הדוגמאות שכתובות במסמך. בהמשך יהיה תרגול מעשי בו תתנסו בכל מה שלמדתם ועוד.

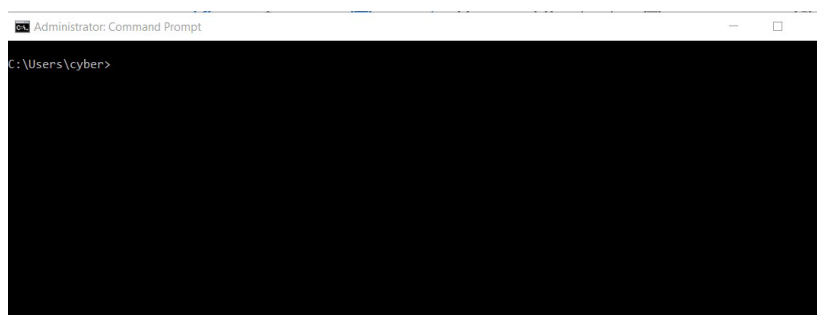
במחשב שלנו, אנחנו יכולים לבצע כל מיני פעולות בעזרת העכבר והמקלדת, **לדוגמה**: יצירת תיקייה חדשה (קליק ימני בdesktop -> חדש -> תיקייה) או מחיקת קבצים (לחיצה על קובץ ולחיצה על delete במקלדת). הצורה הזו נקראת **הצורה הגרפית**.

ממשק גרפי (חלונאי) – ממשק עבודה המשלב בתוכו גרפיקה.

העבודה עם מערכת ההפעלה מלווה בשימוש בעכבר, אייקונים וצבעים.

קיימת צורת עבודה נוספת בה אנו יכולים לבצע פעולות במחשב, והיא נקראת **הקריקטריאלית**. בצורה זו אנו ניתן פקודות למערכת, כמו פתיחת תיקייה או מחיקת קובץ, דרך הקלדתן בטקסט. ההגדרה:

ממשק קריקטריאלי (Console) - ממשק המאופיין בטקסט בלבד. כל פעולות המשתמש מתבצעות על ידי הקלדת טקסט, פקודות בלבד. נשתמש בצורת עבודה זו בתוך ממשק הנקרא **CMD (נקרא גם Command Prompt)**. הוא נראה כך:



נגיע אליו דרך התחל (start) -> CMD.exe

CMD - ממשק קריקטריאלי בו ניתן להקיש פקודות על מנת לבצע פעולות במחשב.

אנו נקליד את הפקודות שברצוננו לבצע במערכת לפי תחביר מסוים (syntax) – כלומר, לכל פקודה יש דרך מסוימת שבה היא צריכה להיכתב על מנת ש-CMD יבין מה עליו לבצע. אם נכתוב בעברית לדוגמה, הפעולה לא תיקלט ולכן לא תתבצע, וגם אם נכתוב בצורה חופשית לפי תחביר השפה האנגלית – גם כאן CMD לא יבין מה הפעולה, כי יש תחביר מסוים לכל פעולה, ולא נוכל לכתוב ב-CMD ללא חוקים מסוימים.

לדוגמה:

אנו רוצים לפתוח תיקייה חדשה על שולחן העבודה בשם LearnCMD.

נפתח את חלון ה-CMD ונקליד: **mkdir Desktop\LearnCMD**



התחביר של הפקודה ליצירת תיקייה חדשה הוא קודם כל שם הפקודה (במקרה שלנו mkdir). אחר כך יגיע הנתיב בו אנו רוצים למקם את התיקייה – Desktop, ולבסוף שם התיקייה החדשה שניצור - LearnCMD.

*התו '\ ' מפריד בין תיקיות. בהמשך נראה עוד דוגמאות של פקודות ב-CMD
*mkdir – קיצור של make directory. הרבה פקודות CMD הן קיצורים מסוג דומה.

מאפייני CMD:

יתרונות: פשטות, יעילות, גמישות בביצוע תהליכים, אוטומציה של תהליכים.

חסרונות: אין גרפיקה, אין שימוש בעכבר, אין עברית, מקום לטעויות בעקבות ריבוי

הקלדה, זכירה של פקודות.

CMD הוא לא sensitive case - כלומר לא רגיש לאותיות קטנות וגדולות, אז אין צורך להקפיד על כך כאשר מקלידים פקודות או שמות של תיקיות וקבצים.

אם מתעסקים ב-CMD, יש להכיר את המושגים הבאים העוסקים בנתיבים:

נתיב מלא:

הנתיב המלא של קובץ מסוים אליו נרצה לפנות. לדוגמה, הנתיב של התיקייה Temp במערכת הוא:

C:\Windows\Temp

נתיב נוכחי:

בכל רגע נתון בממשק ה-CMD אנו נימצא בתוך תיקיה כלשהי.

כל פקודה שנבצע ולא נציין ספציפית על מה נבצע אותה, תתבצע בנתיב הנוכחי.

כפי שניתן לראות בתמונה, הנתיב הנוכחי מופיע בשורת הפקודה



לדוגמה, אם נבצע את הפקודה **mkdir LearnCMD** מבלי להגדיר איפה אנחנו רוצים לפתוח את התיקייה (כמו שהגדרנו desktop בדוגמה הקודמת), התיקייה תפתח בנתיב הנוכחי – במקרה שלנו C:\Users\cyber.

בדומה, אם נמחק קובץ בלי לפרט את הנתיב שהוא נמצא בו, ה-CMD יניח שהקובץ נמצא בנתיב הנוכחי ושם יחפש אותו.

נתיב יחסי:

הנתיב היחסי של קובץ מסוים הוא בעצם המיקום של קובץ מסוים אשר לא מתחיל בכונן. **לדוגמה,** אם **הנתיב הנוכחי** שלי הוא התיקייה **Temp** עליה דיברנו לפני כן, ובתוכה יש לי קובץ טקסט הנקרא a.txt, אז **הנתיב היחסי** שלו הוא פשוט **a.txt**. כאשר נרצה להריץ עליו פקודה כלשהי, למשל מחיקה, אז נכתוב פשוט **del a.txt**.

לעומת זאת אם היינו רוצים לעבוד מול **הנתיב המלא**, היינו כותבים את הפקודה **del**

C:\Windows\Temp\a.txt

שני הנתיבים מתייחסים לאותו אובייקט – לקובץ טקסט שלנו.

פקודות בסיסיות:

- **Dir - הצגת התוכן הקיים תחת הנתיב הנוכחי.**
שימוש: **dir** לבד מציג את תוכן התיקייה בה אנחנו נמצאים.
Dir path – יציג את תוכן התיקייה **path**.
דוגמה: **dir c:\windows\myFolder** יציג את תוכן **myFolder**
*פלט הפקודה **dir** מציג גם תיקיות וגם קבצים שנמצאים תחת הנתיב המבוקש. נבדיל בין קבצים לתיקיות כך: התיקיות מסומנות על ידי **<DIR>** ליד שמן, וליד הקבצים מופיע מספר – גודל הקובץ בבייטים.
echo - הצגת פלט למסך.
שימוש: **echo text** ידפיס את המילה **text** למסך.
Echo text > File.txt ידפיס את המילה **text** לתוך הקובץ **file.txt**. אם **file** לא קיים, הוא ייוצר.
נכיר בהמשך את הניתוב בעזרת הסימן **>**
דוגמה: **echo today is a good day > myFile.txt** ידפיס **today is a good day** לתוך קובץ טקסט בשם **myFile**
type - קריאת טקסט מתוך קובץ (הדפסתו למסך).
שימוש: **type path** ידפיס למסך את התוכן של הקובץ **path**.
דוגמה: **type c:\users\guest\desktop\story.docx** ידפיס למסך את מה שכתוב בתוך הקובץ **story.docx** (קובץ word).
cd (change directory) - שינוי נתיב.
שימוש: **cd path** משנה את הנתיב בו אנחנו נמצאים לנתיב **path**.
cd .. חוזר לתיקיית האב של התיקייה בה אנו נמצאים
דוגמה: אם אנחנו נמצאים ב **c:\windows\temp** ונקליד **cd ..** אנו נחזור לתיקיית האב **c:\windows**.
mkdir - יצירת תיקייה
שימוש: **mkdir path** יוצר את התיקייה **path**.
דוגמה: **mkdir path\hi** יוצר תיקייה בשם **hi** תחת התיקייה **hello**. אם תיקיית **hello** לא קיימת, היא גם תיווצר.
del - מחיקת קובץ
שימוש: **del path** מחוק את הקובץ **path**.

דוגמה: `del file 1 file 2 file 3` ימחק את שלושת הקבצים

- **Rmdir (remove directory) – מחיקת תיקייה (רק אם היא ריקה)**

שימוש: `rm directory` ימחק את התיקייה directory אם היא ריקה.

דוגמה: `rm folder1\folder2` מוחק את folder2 אם היא ריקה

- **Copy – העתקת קובץ**

שימוש: `copy file1 file2` מעתיק את תוכן קובץ file1 לתוך קובץ file2

- **Find - מציאת מחרוזת כלשהי בקובץ**

שימוש: `find string file` מוצא את המחרוזת string בקובץ file.

- **Start - התחל, בדור"כ סקריפט או תכנית.**

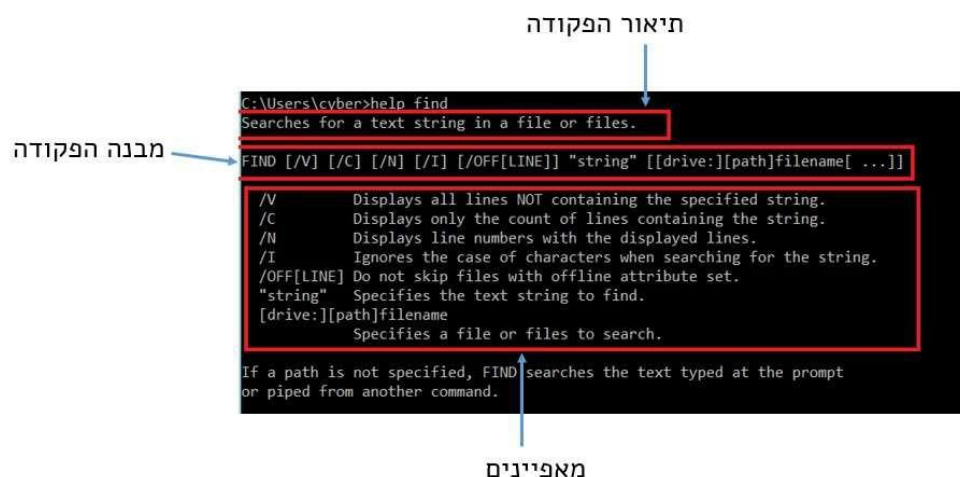
שימוש: `start path` מריץ את path.

דוגמה: `start c:\windows\myfile` יריץ/יפתח את הקובץ myfile

- **Shutdown – כיבוי המערכת**

- **Help – עזרה:**

נשתמש בפקודה help כאשר נרצה לקבל מידע על פקודה מסוימת, איך כותבים אותה (סינטקס) ואילו אפשרויות יש לה. לדוגמה אם נבצע `help find` נקבל עזרה על הפקודה `:find`:



ניתן לראות בתמונה הסבר על תפקיד הפקודה (שהוא למצוא טקסט מסוים בתוך קובץ/ים), את המבנה שלה (איך היא כתובה, באיזה סדר לכתוב אותה) והמאפיינים שלה – לדוגמה /V שגורם להראות בפלט את כל השורות **שלא** מכילות את הטקסט שברצוננו למצוא.

במקום `help` ניתן להשתמש בקיצור `/?` בסוף פקודה כדי לקבל עזרה עליה. `Help` לא נתמך בכל הפקודות ולכן נשתמש לפעמים במקבילה אליו `/?`. למשל כדי למצוא מידע על הפקודה `del` נוכל להשתמש ב `del /?`

מבנה פקודה:

Command Parameters

כל פקודה מורכבת מ-2 חלקים: פקודה ופרמטרים

Command – הפקודה עצמה, מה לבצע. פקודה תופיע פעם אחת בלבד בשורת הפקודה (חוץ

ממקרים שפקודה אחת קוראת לפקודה נוספת)

Parameters – על מה/עם מה/איך לבצע את הפקודה. פרמטר יכול להופיע פעם אחת או יותר, או

בכלל לא. ישנם 2 סוגי פרמטרים:

1. מאפיין - אופי ביצוע הפקודה (למשל /L בפקודה **dir /L** שאומר לבצע את הפקודה **dir**

ולהראות את הפלט באותיות קטנות בלבד).

2. משהו לבצע עליו את הפקודה (למשל **myfolder** בפקודה **dir /L myfolder** שאומר לבצע

את הפקודה **dir** ספציפית על התיקייה **myfolder**)

דוגמה:

dir /O /Q C:

משמעות הפרמטרים:

/O - תוכן התיקייה יוצג על פי סדר מסוים – קודם תיקיות ואח"כ קבצים לפי גודלם.

/Q – מגדיר שיוצגו בפלט גם בעלי הקבצים שבתוך התיקייה.

C:\ - התיקייה שעליה מורצת הפקודה

דגשים לגבי מבנה פקודות ב-CMD:

- הפעולה תבוא תמיד בהתחלה.
- לפני פרמטר ובין פרמטרים יבוא רווח.
- תמיד יבוא "/" לפני מאפיין של הפקודה
- לא חובה לציין פרמטרים ומאפיינים
- לכל פעולה פרמטרים ומאפיינים משלה – אפשר למצוא הכל ב-help.

שילוב פקודות:

Pipeline (|) – אופרטור לשילוב פקודות. נשתמש בו כאשר נרצה להעביר פלט של פקודה אחת כקלט של פקודה שנייה. `command1 | command2`. הפלט של `command1` ייכנס כקלט של `command2`.

איך כותבים pipeline? לוחצים על shift ועל \ ביחד:



דוגמה: `dir | find "c"`

הפקודה `dir` מייצאת פלט שהוא כל הקבצים והתיקיות שנמצאים בתיקייה הנוכחית. הפלט הזה עובר לפקודה `find` שתחפש בו את האות `c`.

ניתוב פלט

Output Redirections – ניתן לנתב פלט של פקודה לתוך קובץ במקום לפקודה אחרת או למסך (ראינו דוגמה בפקודה `echo`). זה קורה ב-3 דרכים:

1. **> - כתיבת פלט פקודה לתוך קובץ מסוים.** במידה והקובץ קיים וכבר יש בו תוכן, הוא יידרס (התוכן הקודם יימחק ובמקומו ייכתב התוכן החדש).

דוגמה: `dir > somefile`

יכניס את הפלט של הפקודה `dir` לתוך הקובץ `somefile` במקום להראות אותו במסך.

2. **>> - כתיבה ועדכון.** אם אנחנו רוצים שהתוכן הקיים בקובץ לא יידרס, נכתוב >> וכך התוכן החדש יתווסף אל התוכן הקיים.

דוגמה: `dir >> somefile` לא ידרוס את הקובץ `somefile`, אלא יוסיף לו את הפלט של `dir` כתוכן בסופו.

3. **< - קבלת קלט מתוך פקודה או קובץ.** במקום להפנות פלט, כאשר נרצה להכניס קלט

לפקודה מסוימת נעשה זאת בצורה הבאה: `command < text`. התוכן של הקובץ `Text` ייכנס לפקודה `command` בתור קלט.

סיכום

כעת הכרתם את הממשק של CMD, את היתרונות בעבודה איתו, פקודות בסיסיות בשימוש וגם מבנה פקודה. בנוסף, למדנו איך עובדים עם קבצים ותיקיות ואיך מקבלים עזרה על פקודות. זה הבסיס, וודאו שאתם מבינים אותו היטב לפני שמתקדמים לתרגול המעשי ולהמשך המקצוע.