**Qt**

**Qt** היא ספריית פיתוח ממשק גרפי (GUI) חוצת־פלטפורמות.

זו ספרייה שכתובה ב־**++C** ומאפשרת לכתוב תוכנה עם חלונות, כפתורים, תפריטים, טבלאות, גרפים וכו' והיא תרוץ גם על **Windows**, גם על **Linux** וגם על **macOS**, בלי לשנות את הקוד (או כמעט בלי).

**Qt** היא המקור (הספרייה עצמה ב-**++C**)

**PyQt** או **PySide** זה רק "גשר" שמאפשר להשתמש ב-**Qt** דרך **python**.

**A screen shot of a computer code

AI-generated content may be incorrect.A screenshot of a computer code

AI-generated content may be incorrect. ++C PyQt5 (python)**

**לסיכום:**

**Qt** – סיפריית **GUI** ב-**++C**  
**PyQt / PySide** עוטפים את **Qt** כדי שנוכל להשתמש בה ב-**python**.

**מושגים**

QApplication - האובייקט הראשי שמנהל את כל האפליקציה שלך ב־Qt, הוא ה"מוח המרכזי" של התוכנה הגרפית.

* הוא מתחיל את התקשורת עם מערכת ההפעלה (Windows / Linux / Mac).
* דואג לנהל **אירועים** (כמו לחיצה על עכבר, מקשים, resize לחלון וכו').
* מחזיק את ה-style הכללי (איך נראים כפתורים, טבלאות וכו').
* מוודא שכל החלונות שייכים לאותה אפליקציה.

Widget- כל רכיב שמצויר על המסך: כפתור, לייבל, חלון, ScrollArea וכו’.

Layout- מנגנון שמסדר Widgets בצורה אופקית, אנכית, גריד.

יחסי הורה ילד – **כל Widget** יכול להיות "ילד" של Widget אחר. (מי שמנהל את המיקום בפועל זה ה־Layout)  
QMainWindow מול QWidget – ההבדל בין חלון ראשי (שיש בו כבר status bar, toolbar, menu bar) לבין חלון גנרי.

QApplication - האפליקציה עצמה, זה "הלב" שמנהל אירועים (קליקים, ציור, resize)

**מושגים כללים**

קוד ברמה העליונה - קוד שכתוב ישירות בקובץ, לא בתוך פונקציה או מחלקה.

קוד ברמה נמוכה - (בתוך פונקציה / מחלקה) ← מוגדר אבל לא מבוצע עד שתקרא לו.

Qapplication