

חעי מימרן

1.8

סביבת הפיתוח

PyCharm



# מה אנחנו צריכים כדי לכתוב תוכניות?

- ▶ עד עתה הכרנו את תהליך הפיתוח על המחשב שלנו, שבו אנחנו עובדים עם שתי תוכנות לסירוגין:
  - ▶ עורך טקסט
  - ▶ תוכנת המפרש להפעלת תוכניות
- ▶ לעיתים בתהליך הפיתוח התוכנה לא מתנהגת בדיוק כמו שרצינו.
- ▶ כיצד מבררים למה ואיפה הטעות?
- ▶ כמו שנראה בהמשך הקורס, כשתכניות הופכות להיות מורכבות, קשה יותר לעקוב אחרי הריצה שלהן. לכן כדאי להכיר כלים מתקדמים יותר לפיתוח.

# מה היינו רוצים כדי לפתח תוכנית מורכבת?

היינו רוצים כלי שיוכל להגיד לנו:

▶ אילו תנאי if מתבצעים ואילו לא

▶ אלו בלוקים של קוד מופעלים שוב ושוב וכיצד

▶ מה הערכים של משתנים בזמן ריצת התכנית?

▶ אילו פונקציות מופעלות בפועל בזמן התכנית, ולפעמים מה קורה בהן

כיוון ששגיאה בתכנית נקראת לעיתים באג (Bug), תהליך של איתור וזיהוי שגיאות נקרא דיבאג (Debug).

תוכנה שמסייעת בתהליך הזה נקראת מנפה שגיאות (Debugger).

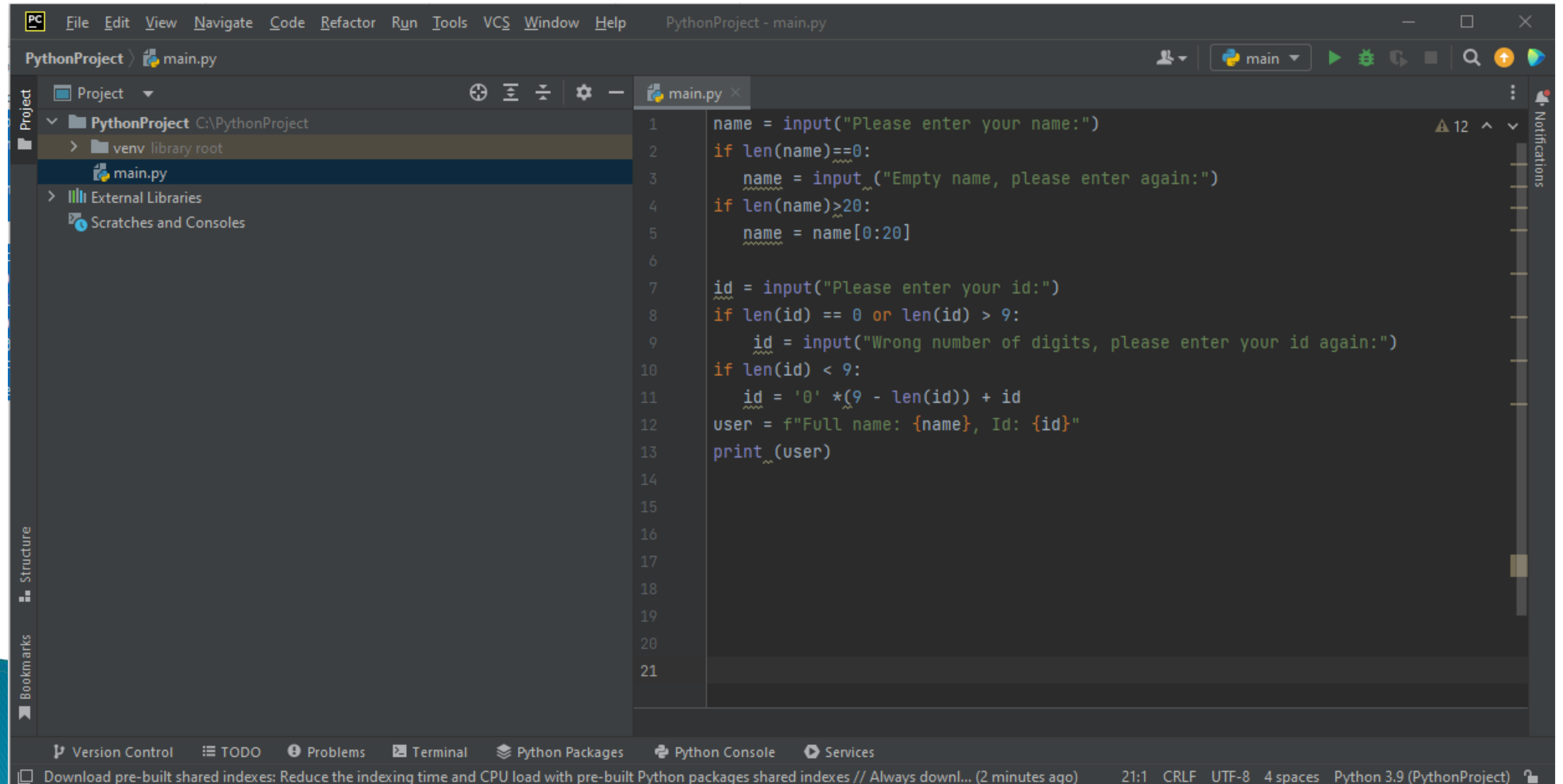
# תוכנת PyCharm

- ▶ זו סביבת פיתוח מהסוג שנקרא IDE – Integrated Development Environment.
- ▶ בסביבה זו אפשר גם לכתוב תכניות בפייתון, גם להריץ אותן עם המפרש בשלמותן, וגם להריץ אותן במצב מבוקר (מצב debug) שבו עוצרים בשלבים מסוימים או אפילו שורה-שורה, בודקים מה התכנית מקבלת וכותבת, מצב התוכנית וערכי משתנים.
- ▶ כמו כן הסביבה מכירה את תחביר שפת פייתון ושפות נוספות, ומסייעת בזמן הכתיבה.
- ▶ הסביבה פותחה ע"י חברה צ'כית בשם JetBrains, גרסה ראשונה יצאה ב-2010.
- ▶ קיימת מהדורה שזמינה בחינם בשם PyCharm Community Edition, ומהדורות מתקדמות יותר עם כלי פיתוח נוספים בתשלום (Professional Edition).
- ▶ הגרסה החינמית מספיקה לצרכים שלנו. כמו כן הסביבה עובדת במערכות הפעלה שונות: Windows, MacOS, Linux.

# איך מפעילים את זה?

- ▶ מורידים את התוכנה, בגרסה החינמית, מהאתר של JetBrains:
- ▶ <https://www.jetbrains.com/pycharm/download/>
- ▶ לצורך עבודה צריך להגדיר בתוכנה "פרויקט" ולפתוח לו תיקיית פרוייקט.
- ▶ לאחר מכן ניתן לכתוב קבצי תוכנית ולשייך אותם לפרוייקט.
- ▶ התוכנה כוללת בתוכה גם מפרש פייתון, אולם ניתן גם לקשר אותה להתקנה אחרת או גרסה אחרת של המפרש.
- ▶ נדגים כיצד עושים זאת עם תכנית התרגיל שבנינו בשעור שעבר.

# PyCharm 70m



The screenshot shows the PyCharm IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, and Help. The title bar indicates the current file is 'PythonProject - main.py'. The left sidebar shows the Project view with the following structure:

- Project
- PythonProject C:\PythonProject
  - venv library root
  - main.py
- External Libraries
- Scratches and Consoles

The main editor window displays the following Python code:

```
1 name = input("Please enter your name:")
2 if len(name)==0:
3     name = input("Empty name, please enter again:")
4 if len(name)>20:
5     name = name[0:20]
6
7 id = input("Please enter your id:")
8 if len(id) == 0 or len(id) > 9:
9     id = input("Wrong number of digits, please enter your id again:")
10 if len(id) < 9:
11     id = '0' *(9 - len(id)) + id
12 user = f"Full name: {name}, Id: {id}"
13 print(user)
14
15
16
17
18
19
20
21
```

The bottom status bar shows various tool windows: Version Control, TODO, Problems, Terminal, Python Packages, Python Console, and Services. The status bar also displays the following information: Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always downl... (2 minutes ago) 21:1 CRLF UTF-8 4 spaces Python 3.9 (PythonProject)