

Prompts Engineering



גלעד מרקמן
פרויקט ב- Github

מה זה Prompt

- Prompt - הוא ההנחיה שאנחנו נותנים למודל השפה (שאלה, הוראה, או בקשה).
- התשובה של המודל תלויה בניסוח ה Prompt. לעיתים קרובות, ההבדל בין תשובה טובה, שאנו יכולים לעשות בה שימוש בקוד, לתשובה לא טובה, תלויה בניסוח ה Prompt.
- דוגמה:
- ש: "מהי בירת צרפת"; ת: "פריז".
- ש: "תן לי סיכום תמציתי על בירת צרפת והייחודיות שלה"; ת: תשובה מפורטת ומלאה.

סוגי תפקידים בשיחה

- System Message – מגדיר את הכללים והתפקוד של המודל. מוגדר פעם אחת בתחילת השיחה.
- דוגמה: אתה מורה למתמטיקה מסביר פנים שתמיד מפרט שלב אחר שלב.
- User Message (משתמש) – שאלת המשתמש או התוכן שמתבקש.
- Assistance Message – התשובה של מודל ה AI
- חשוב: אם יש סתירה בין System ל User – המערכת גוברת.

עקרונות להנדסת פרומפטים

- **בהירות** – תנסח את השאלה בצורה ברורה, ישירה ומפורטת.
- **הקשר (Context)** – תספק מידע רקע או מטרה.
 - תכתוב מכתב <-> אתה יועץ משפטי של החברה. תכתוב מכתב רשמי בלשון משפטית ותקיפה.
- **הודעות מערכת (System Prompt)** – קובעות את הסגנון, התפקיד וכללי הבטיחות.
- **דוגמאות** – הוסף דוגמה אחת (one shot) או יותר (few shot) לתשובה המצופה לשם בהירות ודיוק.
- **Reasoning** – הנחה את המודל לענות בשלבים כדי לשפר את דיוק התשובה. למשל: "פטור שלב אחר שלב".
- **פירוק למשימות (Decomposition)** – פרק את משימה גדולה לשלבים לפתרון, ותן למודל את שלבי העבודה המצופים.

דוגמה

- נשנה את הקוד שלנו בשניים:
- משתמש יוכל להכניס גם System Prompt
- המערכת ללא זיכרון – ההיסטוריה נמחקת לפני כל שאלה.

```
ChatHistory history = new ChatHistory();

while (true)
{
    Console.Write("System>> ");
    string systemMessage= Console.ReadLine();
    if (!string.IsNullOrEmpty(systemMessage))
    {
        history.AddSystemMessage(systemMessage);
    }
    Console.Write("User>> ");
    string userMessage = Console.ReadLine();
    if (string.IsNullOrEmpty(userMessage)) {break;}

    history.AddUserMessage(userMessage);
    var result = await chatService.GetChatMessageContentAsync(history);
    Console.WriteLine(result.Content);
    history.Clear();
}
```