

List comprehension

מתכנתי פייתון בעלי ניסיון אוהבים לקצר תהליכים ולנצל את כוחה של השפה לכתוב מקסימום פונקציונליות במינימום קוד.

אחת הדרכים לעשות זאת, היא ליצור רשימות עם תוכן שנרצה בדרך קיצור.

נראה זאת בדוגמה פשוטה:

נניח שנרצה רשימה שמכילה את המספרים מ 0 עד 99. הדרך הסטנדרטית תהיה:

```
lst1 = []
for i in range(100):
    lst1.append(i)
```

דרך הקיצור:

```
lst1 = [i for i in range(100)]
```

דרך זו של יצירת רשימות ניקראת: list comprehension בתירגום לעברית ניתן לומר: רשימה בהתהוות, או בהבנה

ניתן גם ליצור רשימות יותר מורכבות. למשל, נניח שנרצה רשימה עם ריבועי המספרים הזוגיים בין 2 ל 20.

```
even_squares = [x**2 for x in range(2,21) if x%2 == 0]
```

כאן גם הוספנו משפט תנאי שמגביל את המספרים המיוצרים עבורנו.

זו מעין טכניקה שנועדה 'לשטח' את הקוד ולקצר אותו - מינימליסט כבר אמרנו?

אפשר לעיתים אף להגזים וברצון לקוד מינימלי קצת לסבך את הקוד (לא מומלץ). נראה דוגמה של שילוב של if - else בתוך רשימה כזו (אל תנסו זאת בבית...)

למשל, אם נרצה מתוך המספרים מ 1 עד 20, ליצור רשימה כאשר את הזוגיים נעלה בריבוע ואת האי זוגיים נעלה בשלישית, אפשר לכתוב:

```
square_cube = [x*x if x%2 == 0 else x*x*x for x in range(1,21)]
```

כאמור, קוד פייתון צריך להיות פשוט, קריא וברור. כל סיבוך מתוחכם אינו נחשב יפה, או כמו שאמר איינשטיין: 'כל דבר צריך להיות פשוט ככל האפשר, אבל לא פשוט מידי'