© איילת ויץ

בס"ד

לולאות while

לולאה שחוזרת על עצמה עד שתנאי מסויים כבר לא תקף. היתרון שלה על לולאת for הוא שלא צריך לדעת מראש כמה פעמים לחזור על קוד הלולאה.

מבנה:

while(ביטוי בוליאני)

הביטוי בוליאני נבדק לפני כל איטרציה. אם הוא true, הקוד בתוך הלולאה יתבצע.
אם הוא false הלולאה תיפסק ולא תבוצע יותר.

אם ההוראות בבלוק הלולאה יכללו יותר משורה אחת, נקיף ב {}.

דגשים חשובים

1. **תנאי ראשוני ומעודכן:**
   - יש לוודא שהתנאי המתחיל נכון (true) אחרת הלולאה לא תבוצע כלל.
   - יש לעדכן את התנאי בתוך הלולאה (למשל באמצעות משתנה שסופר את האיטרציות) כדי להימנע מלולאה אינסופית.
2. **לולאה אינסופית :**אם התנאי נשאר תמיד true הלולאה תמשיך לרוץ ללא הפסקה.

דוגמאות

שימוש ב while-עם תנאי יציאה

```
int number = 0;
        while (number != 7)
        {
           Console.WriteLine("Enter a number (enter 7 to exit):");
           number = int.Parse(Console.ReadLine());
        }
        Console.WriteLine("You've exited the loop.");
```

לולאה אינסופית עם תנאי יציאה פנימי

```
    while (true)
        {
           Console.WriteLine("Enter a number (enter 0 to stop):");
           int input = int.Parse(Console.ReadLine());
```

```
    if (input == 0)
    {
        break;  // עצירה אם המשתמש הכניס 0
    }
    Console.WriteLine($"You entered: {input}");
}
```

<u>לולאת do...while</u>

אם צריך לבצע את הקוד לפחות פעם אחת לפני בדיקת התנאי, השתמש בלולאת do...while

לולאה זו תתבצע לפחות פעם אחת, גם אם התנאי בוליאני הוא false כבר מהרגע הראשון.

מבנה:

```
do
{
//הוראות
} while(   ביטוי בוליאני    )
```

לדו':

```
int count = 0;
    do
    {
        Console.WriteLine($"Count is: {count}");
        count++;
    } while (count < 3);
```

(מכיוון שהתנאי נבדק פעם ראשונה לאחר ביצוע הקוד פעם אחת)