Lists

Created time	August 29, 2026 8:31 AM
Category	Loops_Sets
Last edited by	R.A R.A
Last updated time	September 1, 2026 4:24 AM

Lists

יצירת List

בדומה ל־Array:

```
string[] firstNames = new string[5];
```

אפשר ליצור `List` כך:

```
List<string> firstNames = new List<string>();
```

עכשיו אפשר להוסיף אליו שמות:

```
firstNames.Add("Tim");
firstNames.Add("Sue");
firstNames.Add("Bob");
firstNames.Add("Jane");
```

גישה לאיבר האחרון ב־List

```
Console.WriteLine(firstNames[firstNames.Count - 1]);
```

ב־Array השתמשנו ב־`Length`.

ב־List משתמשים ב־`Count`.

מכיוון שהספירה מתחילה מ־`0`, כדי להגיע לאיבר האחרון משתמשים ב:

```
Count - 1
```

List של מספרים

List רק לא חייב להכיל `string`.

אפשר ליצור גם List מסוג `int`:

```
List<int> ages = new List<int>();

ages.Add(1);
ages.Add(2);
ages.Add(3);
ages.Add(4);
```

List — Generic

קוראים לזה:

```
List<T>
```

ה־`T` מייצג את ה־Type שאנחנו רוצים שה־List יכיל.

כשאנחנו יוצרים את ה־List, אנחנו בוחרים מה יהיה ה־Type שלו.

לדוגמה:

```
List<string> firstNames = new List<string>();
```

במקרה הזה ה־List יכול להכיל `string`.

אבל אפשר ליצור גם:

```
List<DateTime>
List<bool>
```

```
List<string>
List<int>
List<double>
List<char>
```

ההבדל בין List ל־Array

`List` דומה מאוד ל־Array, אבל יש לו כמה תוספות ושיפורים.

עם Array אנחנו בדרך כלל צריכים להגדיר מראש את הגודל שלו.

לדוגמה:

```
string[] firstNames = new string[5];
```

כאן אמרנו מראש שיהיו מקום ל־5 פריטים.

לעומת זאת, ב־List:

```
List<string> firstNames = new List<string>();
```

אנחנו **לא מגדירים מראש כמה פריטים יהיו בו**.

אנחנו פשוט יכולים להמשיך להוסיף:

```
firstNames.Add("Tim");
firstNames.Add("Sue");
firstNames.Add("Bob");
firstNames.Add("Jane");
```

ה־List מתאים את עצמו לצרכים שלנו.

הוא יכול לגדול עם הזמן, ולכן הרבה יותר קל וגמיש לעבוד איתו.

אם יש לנו אוסף של נתונים ויש לנו אפשרות לבחור בין Array לבין List, הרבה פעמים יהיה קל יותר לעבוד עם `List`.

יצירת List מתוך String

יש לנו את הנתונים הבאים:

```
string data = "Corey, Smith, Jones";
```

ואנחנו רוצים לפצל אותם.

הדרך עם Array

ב־Array עשינו:

```
string[] lastNames = data.Split(',');
```

הפעולה:

```
Split(',')
```

מחזירה לנו Array.

המרה ל־List

במקום לקבל Array, אנחנו יכולים להפוך את התוצאה ל־List:

```
List<string> lastNames = data.Split(',').ToList();
```

כלומר:

```
Split(',')
```

מפצל את הטקסט ונותן לנו Array.

ואז:

```
.ToList()
```

הופך את ה־Array ל־List.

עכשיו יש לנו List של שמות משפחה.

היתרון של List

מכיוון שזה עכשיו List, אנחנו יכולים להוסיף אליו עוד פריטים:

```
lastNames.Add("Franklin");
```

אתחול List

כשאנחנו כותבים:

```
List<string> firstNames = new List<string>();
```

אנחנו יוצרים List של `string` וקוראים לו:

```
firstNames
```

החלק:

```
new List<string>();
```

אומר למעשה:

צור List חדש מסוג string.

ואז ה־List החדש נשמר בתוך:

```
firstNames
```

הוספת פריטים ל־List

אחרי שיצרנו את ה־List:

```
List<string> firstNames = new List<string>();
```

אפשר להוסיף אליו פריטים באמצעות:

```
.Add()
```

לדוגמה:

```
firstNames.Add("Tim");
```

אנחנו לא צריכים להגיד באיזה מיקום להכניס את `"Tim"`.

אנחנו פשוט מוסיפים אותו ל־List.

לדוגמה:

```
firstNames.Add("Tim");
firstNames.Add("Sue");
firstNames.Add("Bob");
firstNames.Add("Jane");
```

עכשיו יש לנו 4 פריטים בתוך ה־List.

מיקום של פריטים ב־List

כמו Array, גם List מתחיל לספור מ־`0`.

אם יש לנו:

```
firstNames.Add("Tim");
firstNames.Add("Sue");
firstNames.Add("Bob");
firstNames.Add("Jane");
```

המיקומים יהיו:

```
0 = Tim
1 = Sue
2 = Bob
3 = Jane
```

ולכן אפשר לגשת לפריטים לפי המיקום שלהם:

```
Console.WriteLine(firstNames[0]);
```

ידפיס:

```
Tim
```

ואילו:

```
Console.WriteLine(firstNames[3]);
```

ידפיס:

```
Jane
```

Length במקום Count

ב־Array השתמשנו ב:

```
Length
```

אבל ב־List משתמשים ב:

```
Count
```

כדי להגיע לפריט האחרון:

```
Console.WriteLine(firstNames[firstNames.Count - 1]);
```

אנחנו עושים `- 1` בגלל שגם List מתחיל לספור מ־`0`.

הוספת פריטים נוספים

אחד היתרונות החשובים של List הוא שאנחנו יכולים פשוט להמשיך להוסיף אליו פריטים.

לדוגמה:

```
firstNames.Add("Tim");
firstNames.Add("Sue");
firstNames.Add("Bob");
```

```
firstNames.Add("Jane");
firstNames.Add("Frank");
```

לא היינו צריכים לשנות את הגודל של ה־List.

עכשיו:

```
Console.WriteLine(firstNames[firstNames.Count - 1]);
```

ידפיס:

```
Frank
```

כי `"Frank"` הוא עכשיו הפריט האחרון ב־List.