

## C# Les 4

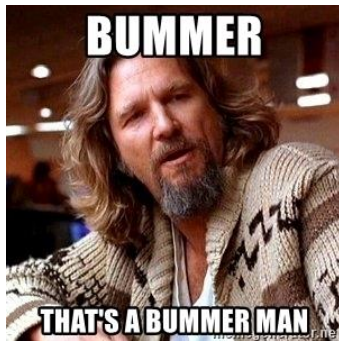
Als programmeur ga je vaak dingen moeten **opzoeken**. Zoals bijna elke andere programmeertaal wordt C# verder uitgebreid en komen er nieuwe mogelijkheden bij. Ook is het niet mogelijk om heel de taal te bespreken in 1 of 2 cursussen en zal je op je pad om een degelijke programmeur te worden vaak aan **zelfstudie** moeten doen. Daarom gaan we vanaf deze les ook wat onderzoekopdrachten doorheen de cursus verwerken.



Goede bronnen zijn de Microsoft documentatie, w3schools en community sites zoals stackoverflow. Als je meer googlet zal je diverse blogs tegenkomen die over C# schrijven, deze zijn vaak ook interessant.

- <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/>
- <https://www.w3schools.com/cs/>

### Lists



In de vorige les gebruikten we arrays om gegevens in een lijst bij te houden. Het vervelende is dat arrays een vaste lengte hebben eens ze zijn aangemaakt. Je kan wel een nieuwe grotere array gaan maken als je meer plaats nodig hebt, maar dat is omslachtig. En een element uit een array verwijderen werkt ook al niet (tenzij je weer een nieuwe array maakt).

Gelukkig bestaat er een handigere manier die we vanaf nu gaan gebruiken: de list. De list is een van de C# *collections*. **Om de collections te kunnen gebruiken moet je nog wel een extra import-regel toevoegen bovenaan je document:**

```
using System.Collections.Generic;
```

Een list van strings maken en tonen doe je zo:

```
List<string> names = new List<string> {  
    "Bergamont Grandurance RD7",  
    "Tokyobike Bisou",  
    "Koga E-Worldtraveller",  
    "Achielle Sam",  
    "Santos Travelmaster 2.8"  
};
```

```
for (int index = 0; index < names.Count; index++)
{
    Console.WriteLine(names[index]);
}
```

### Oefening 1: elementen toevoegen aan een list

In een list kan je ook gemakkelijk elementen toevoegen. Zoek op hoe je dit kan doen. Werk verder op bovenstaande lijst en zorg ervoor dat de gebruiker elementen kan toevoegen aan de list.



```
C:\Users\so19079\source\repos\ConsoleAppLes4\bin\Debug\netcoreapp...
Bergamont Grandurance RD7
Tokyobike Bisou
Koga E-Worldtraveller
Achielle Sam
Santos Travelmaster 2.8

Geef een nieuw fietsmodel in (q om te stoppen): Gazelle Tour Populair_
```



### Elementen verwijderen

Met `RemoveAt(...)` kan je elementen verwijderen uit een lijst. Op deze pagina zie je er een voorbeeld van: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/collections>

### Oefening 2: Toevoegen en verwijderen

- A. Vertrek van onderstaande code. Codeer de methodes `ToonList`, `AddTo` en `RemoveFrom` zodat je fietsen kan toevoegen en verwijderen.

```
static void Main(string[] args)
{
    List<string> namen = new List<string> {
        "Bergamont Grandurance RD7",
        "Tokyobike Bisou",
        "Koga E-Worldtraveller",
        "Achielle Sam",
        "Santos Travelmaster 2.8"
    };

    string input = "";

    while(input != "q")
    {
        Console.Clear();
        ToonList(namen);
    }
}
```

```

    Console.WriteLine("\nAdd, remove or quit? (type a, r or q): ");
    input = Console.ReadLine();

    if(input == "a")
    {
        AddTo(namen);
    }
    else if(input == "r")
    {
        RemoveFrom(namen);
    }
}

Console.WriteLine("Tot de volgende keer!");
Console.ReadLine();
}

```

B. Voeg een extra optie toe zodat je een fietsnaam kan wijzigen. Wijzigen gaat op dezelfde manier als bij arrays:

```
list[index] = "nieuwe waarde";
```

C. Maak 2 extra "parallele" lijsten bij (zoals de auto arrays): één voor de prijs en één voor het aantal versnellingen (je mag de waarden in de lijst verzinnen). Zorg dat het tonen, aanmaken, verwijderen en aanpassen werkt voor de 3 parallele lijsten.

