

C# Essentials Dictionaries

Koen Bloemen



**DE HOGESCHOOL
MET HET NETWERK**

Elfde-Liniestraat 24, 3500 Hasselt, www.pxl.be





- Dictionaries

Dictionaries

- Dictionary<K,V>
- Slaat elementen op als sleutelpaar (key-value pair)
 - Sleutel (key) van type K
 - Waarde (value) van type V
- Met de sleutel kan je waarden opzoeken, aanpassen en verwijderen
 - Is gelijkaardig aan de index van een array
 - Kan je vergelijken met een soort index die eender welke waarde of type kan hebben
- Iedere waarde heeft een sleutel toegekend
 - Waarde kan eender welk type hebben
- Sneller als je vaak gegevens moet opzoeken (sneller dan (binair) zoeken in List)
- Trager als je door gegevens moet lopen

Dictionaries

- Declaratie en initialisatie

```
Dictionary<int, int> eenDictionaryMetGetallen;
```

```
//In dict2 zijn de keys van type string en de values van type int array  
Dictionary<string, int[,]> dict2 = new Dictionary<string, int[,]>();
```

```
// Je kan een dictionary ook onmiddellijk initialiseren  
Dictionary<int, string> dict= new Dictionary<int, string>()  
{  
    { 1, "hello"},  
    { 3, "world"},  
    { 9001, "over 9000"}  
};
```

Dictionaries

- Toevoegen en wijzigen van values

```
Dictionary<int, string> dict= new Dictionary<int, string>()
{
    { 1, "hello"},
    { 3, "world"},
    { 9001, "over 9000"}
};
Dictionary<string, int> dictWithIntValues= new Dictionary<string, int>();

// Toevoegen
dict.Add(2, "hello");
dict.Add(2, "gaat niet werken"); // werkt niet => key bestaat al
dict[3] = "vier waarde"; // werkt ook met index-notatie
dictWithIntValues["a"] = 1;

// Aanpassen
dict[1] = "overschreven";
```

Dictionaries

- Verzameling opvragen van alle keys en values

```
Dictionary<int, string> dict= new Dictionary<int, string>()
{
    { 1, "hello"},
    { 3, "world"},
    { 9001, "over 9000"}
};
// OPGELET: door een dictionary lopen is trager dan lopen door een list!!!
foreach(var key in dict.Keys)
{
    Console.WriteLine(key);
}
foreach(var value in dict.Values)
{
    Console.WriteLine(value);
}
```

Dictionaries

- Element opzoeken

```
Dictionary<string, int> punten = new Dictionary<string, int>()
{
    { "Tom", 12},
    { "Ozgun", 17},
    { "Elise", 16}
};
// Element opzoeken in dictionary
int puntenTom = punten["Tom"];

// Element PROBEREN opzoeken in dictionary => zelfde werking als TryParse
int puntenAnna;
bool isGevonden = punten.TryGetValue("Anna", out puntenAnna);
if (isGevonden)
    Console.WriteLine($"Anna heeft {puntenAnna}/20");
```

Dictionaries

- Element verwijderen

```
Dictionary<string, int> punten = new Dictionary<string, int>()
{
    { "Tom", 12},
    { "Ozgun", 17},
    { "Elise", 16}
};
// Element verwijderen uit dictionary
punten.Remove("Tom");

// Resultaat van verwijdering opvragen
bool isVerwijderd = punten.Remove("Anna");
```


Dictionaries

- Controleren of element (key) bestaat

```
Dictionary<string, int> punten = new Dictionary<string, int>()
{
    { "Tom", 12},
    { "Ozgun", 17},
    { "Elise", 16}
};
bool bestaat = punten.ContainsKey("Tom");
```

Dictionaries

- Loopen over dictionary (TE VERMIJDEN = TRAAG)

```
Dictionary<string, int> punten = new Dictionary<string, int>()
{
    { "Tom", 12},
    { "Ozgun", 17},
    { "Elise", 16}
};
// Loopen over dictionary en keys en values opvragen van elk item
foreach (var item in punten)
{
    Console.WriteLine($"{item.Key}: {item.Value}/20");
}
```

Dictionaries

- Dictionary omzetten naar List (TE VERMIJDEN!)

```
Dictionary<string, int> punten = new Dictionary<string, int>()
{
    { "Tom", 12},
    { "Ozgun", 17},
    { "Elise", 16}
};
//Omzetten naar List
List<KeyValuePair<string, int>> list = punten.ToList();
// Daarna afdrukken
foreach (KeyValuePair<string, int> pair in list)
{
    Console.WriteLine($"{pair.Key}: {pair.Value}");
}
```