

# C# Essentials Hulpfuncties arrays

Koen Bloemen

Wim Bervoets

Kristof Palmaers



**DE HOGESCHOOL  
MET HET NETWERK**

Elfde-Liniestraat 24, 3500 Hasselt, [www.pxl.be](http://www.pxl.be)





- Hulpfuncties arrays

# Hulpfuncties arrays

- IndexOf(): Index van eerste voorkomen

```
string[] namen = new string[] { "Peter", "Tom", "Piet", "Tom", "Benny", "Anna" };  
int index = Array.IndexOf(namen, "Tom");  
Console.WriteLine(index); // 1  
// Index Piet?  
// Index Jef? ⇒ -1 (als element niet gevonden wordt)  
// Index laatste voorkomen van Tom?
```



# Hulpfuncties arrays

- LastIndexOf(): Index van laatste voorkomen

```
string[] namen = new string[] { "Peter", "Tom", "Piet", "Tom", "Benny", "Anna" };  
int index = Array.LastIndexOf(namen, "Tom");  
Console.WriteLine(index); // 3
```

# Hulpfuncties arrays

- Copy(): Array kopiëren

```
string[] namen = {"Peter", "Sven", "Piet", "Tom", "Benny", "Anna"};
string[] kopie = new string[namen.Length];
Array.Copy(namen, kopie, namen.Length); // Laatste parameter bepaald hoeveel
items er gekopieerd worden
foreach (string naam in kopie)
{
    Console.WriteLine(naam);
}
// Peter
// Sven
// Piet
// Tom
// Benny
// Anna
```

# Hulpfuncties arrays

- Reverse(): Array omkeren

```
string[] namen = {"Peter", "Sven", "Piet", "Tom", "Benny", "Anna"};
Array.Reverse(namen);
foreach (string naam in namen)
{
    Console.WriteLine(naam);
}
// Anna
// Benny
// Tom
// Piet
// Sven
// Peter
```

# Hulpfuncties arrays

- Find(): Zoek element *(met lambda)*

```
string[] consoles = {"PS2", "XBox", "Dreamcast", "N64", "Gamecube", "PS5"};

// Zoekt eerste element dat begint met N.
string console1 = Array.Find(consoles, element => element.StartsWith("N"));
Console.WriteLine(console1); // N64

// Zoekt eerste element dat 9 karakters heeft.
string console2 = Array.Find(consoles, element => (element.Length == 9));
Console.WriteLine(console2); // Dreamcast
```

- Predicates (*element => (element.Length == 9)*) met lambda expression (*lambda operator =>*)

# Hulpfuncties arrays

- FindAll(): Zoek alle elementen *(met lambda)*

```
string[] consoles = {"PS2", "XBox", "Dreamcast", "N64", "Gamecube", "PS5"};
// Zoekt alle elementen die beginnen met een P.
string[] pConsoles = Array.FindAll(consoles, element =>
element.StartsWith("P"));
foreach (string console in pConsoles)
{
    Console.WriteLine(console);
}
// PS2
// PS5
```



# Hulpfuncties arrays

- Exists(): Bestaat het element (true/false) *(met lambda)*

```
string[] consoles = {"PS2", "XBox", "Dreamcast", "N64", "Gamecube", "PS5"};
bool bestaat = Array.Exists(consoles, element => element.Equals("Phantom"));
Console.WriteLine(bestaat);
// false
// Hoe kijken of er een console is met 3 karakters?
bool bestaat = Array.Exists(consoles, element => element.Length.Equals(3));
```

# Hulpfuncties arrays

- Sort(): Sorteren van klein naar groot

```
string[] namen = {"Peter", "Sven", "Piet", "Tom", "Benny", "Anna"};
Array.Sort(namen);
foreach (string naam in namen)
{
    Console.WriteLine(naam);
}
// Anna
// Benny
// Peter
// Piet
// Sven
// Tom
```

# Hulpfuncties arrays

- Hoe sorteren van groot naar klein?

```
string[] namen = {"Peter", "Sven", "Piet", "Tom", "Benny", "Anna"};
//??
Array.Sort(namen);
Array.Reverse(namen);
foreach (string naam in namen)
{
    Console.WriteLine(naam);
}
// Tom
// Sven
// Piet
// Peter
// Benny
// Anna
```

# Hulpfuncties arrays

- Sort(): Bepaald gedeelte van array sorteren

```
string[] namen = { "Peter", "Sven", "Piet", "Tom", "Benny", "Anna" };  
int startIndex = 1;  
int length = 4;  
Array.Sort(namen, startIndex, length);  
foreach (string naam in namen)  
{  
    Console.WriteLine(naam);  
}  
// {"Sven", "Piet", "Tom", "Benny"} =>  
// Peter  
// Benny  
// Piet  
// Sven  
// Tom  
// Anna
```



# Hulpfuncties arrays

- BinarySearch(): SNEL element zoeken in een **GESORTEERDE** array

```
string[] namen = {"Peter", "Sven", "Piet", "Tom", "Benny", "Anna"};
Array.Sort(namen);
// Anna Benny Peter Piet Sven Tom
int index = Array.BinarySearch(namen, "Sven");
Console.WriteLine(index);
// 4

// Waarom op voorhand gesorteerd?
```

- Binair zoeken: telkens kijken in de juiste helft

```
// Anna Benny Peter Piet Sven Tom
// Anna Benny Peter [Piet Sven Tom]
// Anna Benny Peter [Piet Sven] Tom
// Anna Benny Peter Piet Sven Tom
```

# Hulpfuncties arrays

- Clear(): Array leegmaken

```
// 1D Array leegmaken
float[] getallen = { 1.1f, 1.2f, 1.3f, 1.4f, 1.5f, 1.6f, 1.7f };
int startIndex = 2;
int length = 4;
Array.Clear(getallen, startIndex, length);
// { 1.1f, 1.2f, 0.0f, 0.0f, 0.0f, 0.0f, 1.7f }

// Hoe volledig leegmaken?
int startIndex = 0;
int length = getallen.Length;
Array.Clear(getallen, startIndex, length);
```

# Hulpfuncties arrays

- Resize(): Grootte van array veranderen (wordt weinig gebruikt)

```
// C# allocceert nieuwe array en kopieert inhoud over

int[] getallen = { 3, 4, 5, 6 };
// Groter maken
Array.Resize(ref getallen, 10); // 3 4 5 6 0 0 0 0 0 0
// Kleiner maken
Array.Resize(ref getallen, 3); // 3 4 5

// Automatische resize?: Gebruik List<T> uit System.Collections.Generic!
// (is voor volgende les!)
```