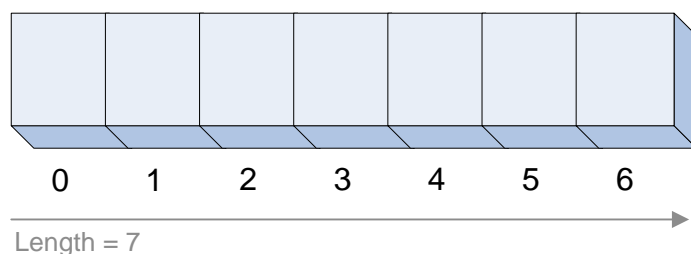


Vad är en array?

En array är ett sammanhängande antal element av samma typ som representeras med ett variabelnamn. De olika elementen koms åt genom att använda variabelnamnet och ett index mellan hakparenteser.



Figur 1. Array med sju element.

Hur många element en array har bestäms då den skapas. Antalet element är fast och kan inte ändras efter att en array har skapats.

En arrays längd är detsamma som det totala antalet element i arrayen.

Arrayindex är nollbaserade. Det första elementet i en array finns på index 0, inte på index 1. Indexvärdet 1 ger det andra elementet. Indexområdet är alltid från 0 till arrayens längd – 1.

En array är en referenstyp, men elementen i arrayen kan vara av värde- eller referenstyp. För att referera till en array krävs en arrayvariabel.

Satsen

```
int[] myArray;
```

skapar en arrayvariabel som kan referera till en array med heltal av typen `int`. Någon array skapas inte. För att skapa en array med sju heltal av typen `int` för den tidigare deklarerade variabeln `myArray` skriver du:

```
myArray = new int[7];
```

Då en array skapas initieras varje element automatiskt till typens standardvärde. Standardvärdet är 0 för heltalstyper, `0.0` för flyttal, `false` för booleska och `null` för referenstyper.

Elementen i en array kan initieras till explicita värden då arrayen skapas. Följande sats skapar en arrayvariabel, en array och initierar arrayens sju element till värdena mellan klammerparenteserna.

```
int[] myArray = new int[7] { 3, 5, -19, 8, 14, 74, -12 };
```

För att komma åt ett enskilt elements värde använder du ett index som visar vilket element du vill använda. Följande kod skriver ut det fjärde elementets värde:

```
Console.WriteLine(myArray[3]); // skriver ut 8
```

Satsen

```
int myValue = myArray[2]; // myValue tilldelas värdet -19
```

kopierar värdet som finns i det tredje elementet till variabeln `myValue`.

Ett elements värde kan ändras. Satsen

```
myArray[myArray.Length - 1] = 49; // myValue[6] tilldelas värdet 49
```

tilldelar det sista elementet värdet 49. Det sista elementet i arrayen har index 6. Arrayens längd är 7 och 7 minus 1 är 6. Egenskapen `Length` använder du då du vill ta reda på hur många element en array innehåller.

Med hjälp av en ”for”-sats och egenskapen `Length` kan du stega igenom en array:

```
for (int i = 0; i < myArray.Length; ++i)
{
    Console.WriteLine(myArray[i]);
}
```

Satserna ovan skriver ut alla element i arrayen, från det första till och med det sista, med hjälp av index. Använder du ”foreach”-satsen behöver du inte använda dig av index. ”for”-satsen kan istället skrivas med en ”foreach”-sats enligt:

```
foreach (int myValue in myArray)
{
    Console.WriteLine(myValue);
}
```

”foreach”-satsen deklarerar en iterationsvariabel (`int myValue`) som automatiskt tilldelas värdet som varje element i arrayen har.

Enklaste sättet att kopiera en array är att använda metoden `Clone`. Satsen

```
int[] copy = (int[])myArray.Clone();
```

skapar en kopia av arrayen som `myArray` refererar till. Metoden `Clone` returnerar en object-referens varför returvärdet måste typomvandlas, vilket är precis vad (`int[]`) gör.

En arrays innehåll kan sorteras. För att innehållet ska kunna sorteras krävs att typen arrayen är av implementerar interfacet `IComparable`. Typer i dotnetramverket som `int` och `string` gör detta och en array med heltal av typen `int` sorteras enkelt med:

```
Array.Sort(myArray);
```