## Lesbrief 2: Variabelen en datatypes

| * Probeer de voorbeelden uit. * Voer de opdrachten uit. |
| --- |

In deze lesbrief ga je leren over **variabelen** en **datatypen**.

In de basis worden variabelen gebruikt om waarden te onthouden. Stel je wilt in plaats van “World” in de tekst “Hello World” de naam van een gebruiker op het scherm tonen, bijvoorbeeld “Hello Tom” dan zul je eerst de naam van een gebruiker moeten achterhalen en deze ook moeten onthouden. Dat doe je in een **variabele**.

De volgende code vraagt de gebruiker om zijn of haar naam en print deze op het scherm. De naam van de gebruiker wordt onthouden in een variabele genaamd “naam”.

| using System;  namespace HelloWorld {  class Program  {  static void Main(string[] args)  {  string naam;  Console.WriteLine("Wat is jouw naam?");  naam = Console.ReadLine();  Console.WriteLine("Hello " + naam);  }  } } |
| --- |

### 

### Datatypes

In bovenstaand programma wordt een variabele “naam” gecreëerd: *string naam*. Echter moet je C# iets meer vertellen, namelijk wat voor type gegeven, data, de variabele “naam” bevat.

C# kent verschillende **datatypes**. Datatypes representeren o.a. gehele getallen, kommagetallen, letters en woorden. Enkele datatypes zijn:

| **Type** | **Omschrijving** |
| --- | --- |
| int | Geheel getal (positief en negatief) |
| float | Komma getal (32 bit) |
| double | Komma getal (64 bit) |
| char | Letter of character |
| string | Woord of zinnen |
| bool | true of false (waar of niet waar) |

Meer over datatypes kun je vinden op: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/types-and-variables>

In bovenstaand voorbeeld heeft de variabele “naam” als datatype **string**.

Een variabele kun je ook meteen een waarde geven. Enkele voorbeelden:

*string naam;* of *string naam = “Piet”*;

*int nummer;* of *int nummer = 2;*

*float btw;* of *float btw = 0.21f; //****let op*** *de f aan het eind van het getal.*

*double hoek;* of *double hoek = 5.123;*

*char c;* of *char c = ‘c’;*

*bool is\_leerling;* of *bool is\_leerling = true;*

Variabelen kun je ook op een andere manier een waarde geven, bijvoorbeeld:

| naam = Console.ReadLine(); |
| --- |

In dit voorbeeld krijgt “naam” als waarde het resultaat van de “Console.ReadLine” methode. Als je jouw naam intypt dan heeft de variabele “naam” jouw naam als waarde.

Vervolgens printen we “Hello” gevolgd door de naam. Ok hier gebruik je de variabele naam.

| Console.WriteLine("Hello " + naam); |
| --- |

Met de “+” operator plak je 2 strings achter elkaar. Dit wordt ook wel **concateneren** genoemd.

| **Opdracht**   * Schrijf een programma die de gebruiker vraagt om zijn leukste computerspel en dit vervolgens op het scherm print. |
| --- |

### Namen van variabelen

Een variabele moet een duidelijke naam zijn voor de data die zo’n variabele bevat. Een variabele naam begint met een klein letter. Indien een variabele naam uit meerdere woorden bestaat dan schrijf je de woorden aan elkaar vast, waarbij elk volgend woord met een hoofdletter begint. Enkele voorbeelden:

*message*

*name*

*line*

*phrase*

*speedOfLight*

*localCustomer*

*scoreQuery*

### Oefenvoorbeeld: Boodschappen programma

In onderstaand voorbeeld worden meerdere variabelen gebruikt. Het programma vraagt de gebruiker om het aantal pakken koffie en rekent vervolgens uit wat deze kosten.

| using System;  namespace Boodschappen {  class Program  {  static void Main(string[] args)  {  string product = "Koffie";  int aantal;  float prijsExclBtw = 3.95f;  float btwPercentage = 0.21f;    Console.WriteLine("Een pakje " + product + " kost " + prijsExclBtw + " ex btw.");  Console.WriteLine("Voer een aantal in: ");   aantal = Convert.ToInt32(Console.ReadLine());  float totaalPrijs = aantal \* prijsExclBtw + (aantal \* prijsExclBtw) \* btwPercentage;   Console.WriteLine(aantal + " pakken " + product + " kost in totaal " + totaalPrijs);   }  } } |
| --- |

In bovenstaand programma worden een aantal variabelen met hun datatype gedeclareerd:

| string product = "Koffie";  int aantal;  float prijsExclBtw = 3.95f;  float btwPercentage = 0.21f; |
| --- |

De variabelen “product”, “prijsExclBtw” en “btwPercentage” bevatten reeds een waarde. De variabele “aantal” krijgt later in het programma een waarde.

### Getallen: Rekenen met variabelen

In C# kun je met variabelen rekenen mits deze van het type byte, short, int, float, double, long of decimal zijn. In de boodschappen applicatie wordt de totaalprijs berekend inclusief de BTW.

| float totaalPrijs = aantal \* prijsExclBtw + (aantal \* prijsExclBtw) \* btwPercentage; |
| --- |

Hier wordt de variabele “aantal” van het type “int” vermenigvuldigd met een variabele van het type “float” (prijsExclBtw).

Als je variabelen van hetzelfde datatype van elkaar aftrekt, of met elkaar optelt, vermenigvuldigd of deelt, dan krijg je dezelfde datatype als resultaat, dus:

*float \* float === geeft een ===> float*

*int - int === geeft een ===> int*

*double / double === geeft een ===> double*

Echter in het geval van 2 verschillende datatypes dan zal C# kiezen voor het “meest precieze” datatype, dus:

*float \* int === geeft een ===> float*

*int - double === geeft een ===> double*

*int % float === geeft een ===> float*

| **Opdracht** - boodschappen  Pas het boodschappen programma aan zodanig dat:   * Het product een computerspel omvat. * De prijs is de prijs van het computerspel. * De gebruiker wordt gevraagd hoeveel computerspellen die wilt kopen. * De applicatie de totale prijs inclusief BTW uitrekent en toont aan de gebruiker. |
| --- |

### Constanten

Variabelen kunnen ook constanten zijn. Constanten hebben als eigenschap dat ze niet van waarde veranderen. Een constante variabele wordt aangeduid met de **const** keyword en begint met een hoofdletter.

De structuur of syntax ziet er als volgt uit:

*const <datatype> <variabele naam> = <waarde>;*

Voorbeelden van constanten zijn:

| Het aantal dagen in een jaar | const int Dagen = 365; |
| --- | --- |
| Het aantal maanden in een jaar | const int Maanden = 12; |
| Hoog btw tarief | const float BtwHoog = 0.21f; |
| Het aantal seconden in een minuut | const int Seconden = 60; |
| Een constante titel | const string Titel = “Harry Potter”; |

| **Opdracht**:  Verander je boodschappen programma.   * Maak van de btw percentage een constante. * Maak de naam van het product constant |
| --- |

### 

### Convert

In het voorbeeld van het Boodschappen programma krijgt de variabele “aantal” de waarde die de gebruiker heeft ingevoerd.

| aantal = Convert.ToInt32(Console.ReadLine()); |
| --- |

Echter retourneert de methode “Console.ReadLine()” een string als waarde. Daarom converteren we deze waarde eerst naar een “int” met behulp van de methode “Convert.ToInt32”:

Met de Convert klasse kun je een datatype converteren naar een andere datatype. De convert klasse kent vele methoden voor het converteren van datatypes naar andere datatypes. Meer hierover kun je lezen op:

<https://docs.microsoft.com/en-us/dotnet/api/system.convert?view=netframework-4.7.2>