

Ons eerste programma begrijpen

Wat is nu eigenlijk een computerprogramma?

Een programma is een reeks instructies voor de computer. Deze instructies vertellen de computer precies wat te doen en deze instructies worden altijd opgevolgd. Net zoals mensen, kunnen computers alleen instructies volgen als deze zijn verwoord in een taal die ze kunnen begrijpen. Dit zijn de zogenaamde programmeertalen. Er zijn veel talen die de computer kan begrijpen en **Small Basic** is hier een van.

Stel je eens een gesprek voor tussen jou en je vriend. Jij en je vrienden gebruiken woorden, georganiseerd in zinnen, waarmee je informatie heen en weer overdraagt. Op gelijke wijze kunnen programmeertalen verzamelingen woorden bevatten die in zinnen kunnen worden georganiseerd waarmee informatie aan de computer kan worden overgedragen. En programma's zijn in feite een reeks zinnen (soms maar een paar en soms vele duizenden) die zinvol is voor zowel de programmeur de computer.

Small Basic-programma's

Een typisch Small Basic-programma bestaat uit een aantal *instructies*. Elke regel van het programma vertegenwoordigt een instructie en elke instructie is een instructie voor de computer.

Als we de computer vragen een Small Basic-programma uit te voeren, wordt het programma geladen en de eerste instructie gelezen. De computer begrijpt wat we proberen te zeggen en vervolgens wordt onze instructie uitgevoerd. Als de eerste instructie is uitgevoerd, keert de computer terug naar het programma en wordt de tweede regel gelezen en uitgevoerd. De computer gaat door totdat het einde van het programma wordt bereikt. En dan is ons programma voltooid.

De computer kan vele talen begrijpen. Java, C++, Python, VB, enz. zijn allemaal krachtige computertalen die worden gebruikt voor het ontwikkelen van eenvoudige tot complexe softwareprogramma's.

Terug naar ons eerste programma

Hier is het eerste programma dat we hebben geschreven:

```
TextWindow.WriteLine("Hallo wereld")
```

Dit is een eenvoudig programma dat bestaat uit één *instructie*. Deze instructie vertelt de computer een regel tekst, **Hallo wereld**, naar het tekstvenster te schrijven.

In het geheugen van de computer wordt dit letterlijk vertaald naar:

```
Write Hallo wereld
```

Je hebt misschien al opgemerkt dat de instructie kan worden gesplitst in kleinere segmenten, zoals zinnen kunnen worden opgesplitst in woorden. In de eerste instructie hebben we 3 afzonderlijke segmenten:

- a) TextWindow
- b) WriteLine
- c) "Hallo wereld"

De punt, haakjes en aanhalingstekens zijn allemaal leestekens die op de juiste posities in de instructie moeten worden geplaatst zodat de computer onze bedoeling kan begrijpen.

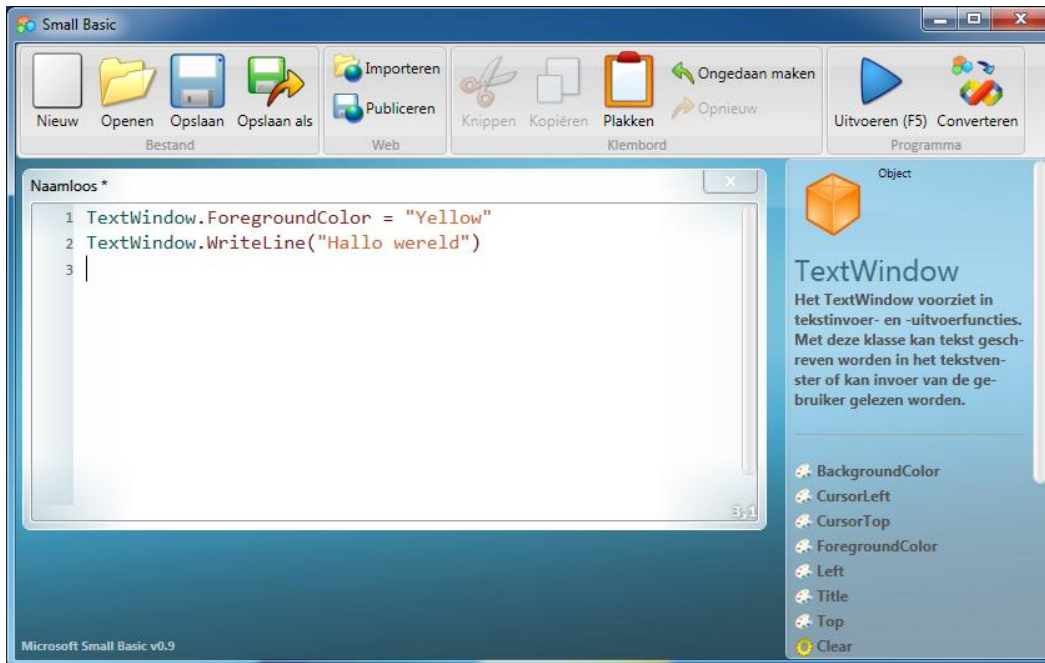
Misschien kun je je nog het zwarte venster herinneren dat verscheen toen we ons eerste programma uitvoerden. Dat zwarte venster wordt het TextWindow genoemd of soms ook console. Dit is de plaatst waar het resultaat van dit programma wordt weergegeven. **TextWindow** wordt in ons programma een *object* genoemd. Een aantal van dergelijke objecten zijn beschikbaar voor gebruik in onze programma's. We kunnen verschillende *bewerkingen* uitvoeren op deze objecten. We hebben de WriteLine -bewerking al in ons programma gebruikt. Je hebt misschien ook al opgemerkt dat de bewerking WriteLine wordt gevolgd door **Hallo wereld** binnen aanhalingstekens. Deze tekst wordt doorgegeven als invoer bij de WriteLine-bewerking en wordt vervolgens naar het tekstvenster geschreven. Dit wordt een *invoer* bij de bewerking genoemd. Sommige bewerkingen kunnen meer dan een invoer hebben terwijl andere er geen enkele hebben.

Leestekens zoals aanhalingstekens, spaties en haakjes zijn heel belangrijk in een computerprogramma. Gebaseerd op hun positie en aantal, kunnen ze de betekenis wijzigen van wat wordt uitgedrukt.

Ons tweede programma

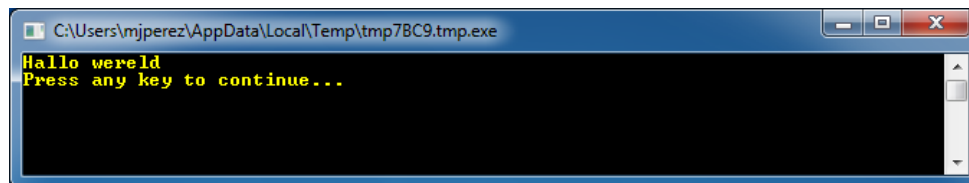
Nu we ons eerste programma begrijpen, kunnen we het ingewikkelder maken door een aantal kleuren toe te voegen.

```
TextWindow.ForegroundColor = "Yellow"  
TextWindow.WriteLine("Hallo wereld")
```



Afbeelding 5 – Kleuren toevoegen

Wanneer je het bovenstaande programma uitvoert, zie je dat dezelfde "Hallo wereld"-zin naar het TextWindow wordt geschreven, maar deze zin verschijnt nu in geel in plaats van in grijs zoals in het vorige voorbeeld.



Afbeelding 6 – Hallo wereld in geel

Let op de nieuwe instructies die we aan ons oorspronkelijke programma hebben toegevoegd. Er wordt een nieuw woord gebruikt: `ForegroundColor`. Dit woord hebben we gelijkgesteld aan de waarde `"Yellow"`. Dit betekent *dat we "Yellow"* hebben toegewezen aan `ForegroundColor`. Het verschil nu tussen `ForegroundColor` en de bewerking `WriteLine` is dat `ForegroundColor` geen invoer of haakjes nodig heeft. In plaats hiervan wordt het gevolgd door een *gelijk aan*-symbool en een woord. We definiëren `ForegroundColor` als een *eigenschap* van `TextWindow`. Hieronder volgt een lijst met waarden die geldig zijn voor de eigenschap `ForegroundColor`. Probeer `"Yellow"` met een van deze waarden te vervangen en bekijk de resultaten. Vergeet de aanhalingstekens niet, dit zijn verplichte leestekens.

```
Black  
Blue  
Cyan  
Gray  
Green  
Magenta  
Red  
White  
Yellow  
DarkBlue  
DarkCyan  
DarkGray  
DarkGreen  
DarkMagenta  
DarkRed  
DarkYellow
```