Begin Microsoft Small Basic: Hoofdstuk 2: Basisbeginselen van het kleine basisprogramma

Inhoudsopgave

Bekijken en bekijken

In de eerste klas besteedden we al onze tijd aan het voorbereiden van onze computer voor het maken en uitvoeren van kleine basisprogramma's. In deze tweede klas zullen we dieper ingaan op enkele van de taken die we hebben gedaan. We zullen het welkomstprogramma uit klasse 1 opnieuw bekijken. We zullen enkele van de basisregels leren voor het schrijven van kleine basisprogramma's. We zullen een programma maken en opslaan met Small Basic. Dit geeft ons de vaardigheden die nodig zijn om ons eerste Small Basic-programma in klasse 3 te maken.

Het welkomstprogramma (Revisited)

Start **SmallBasic** en open het **welkomstprogramma** dat we in klasse 1 hebben bekeken Hier is de code die u in de editor ziet:

Dit hoofdstuk is een bewerking van het boek *BEGINNING Microsoft Small Basic* van Philip Conrod en Lou Tylee.

Om dit boek in zijn geheel te kopen, zie de

```
'
' Welcome Program
' Beginning Small Basic
'
TextWindow.Title = "Welcome Program"
TextWindow.WriteLine("Welcome to Beginning Small Basic!")
```

Computer Science For Kids website .

A program is made up of many **statements**. Every line is a statement and every statement instructs the computer to do something.. Let's go through this code line by line to explain its structure and see what each line does.

De eerste paar regels van het programma zijn:

```
'
' Welcome Program
' Beginning Small Basic
'
```

Deze regels zijn **opmerkingen**. Ze geven gewoon wat informatie over wat het programma is en bieden wat contactgegevens. De opmerking begint met een enkele apostrof (**'**). Deze lijnen worden ook wel **programheader genoemd**. Het is een goed idee om altijd een koptekst op uw Small Basic-programma's te plaatsen om iemand een idee te geven van wat uw programma doet en wie het heeft geschreven. Bij het uitvoeren van een programma negeert Small Basic eventuele opmerkingen - hun enige gebruik is uitleg geven.

De eerste non-comment verklaring is:

TextWindow.Title = "Welkomstprogramma"

Herinner je je het welkomstprogramma nog in klas 1? Toen u het programma uitvoerde, zag u dit venster

Let op de woorden **WelcomeProgram** in de titelbalk van het venster. In de bovenstaande coderegel wordt die titel weergegeven. In deze regel is **TextWindow** een **object** dat is ingebouwd in Small Basic - het is het venster dat de uitvoer van het programma weergeeft. Small Basic heeft een aantal van dergelijke objecten beschikbaar voor ons gebruik. We zullen het **TextWindow-object** uitgebreid gebruiken in onze eerste paar programma's. Objecten hebben zowel **eigenschappen** als **methoden**. Eigenschappen beschrijven objecten, terwijl methoden dingen doen met objecten. In deze enkele regel code stellen we de eigenschap **Title** van het **TextWindow-object** in op de tekstreeks "**Welcome Program**". De punt (**.**) en de toewijzingsoperator (**=**) zijn interpunctie die op de juiste manier moet worden geplaatst zodat de computer uw intentie kan begrijpen. Deze coderegel zegt letterlijk 'stel de eigenschap **Title** van het **textwindow-object** in op **Welkomstprogramma'**.

De andere stelling in dit korte programma is:

TextWindow.WriteLine("Welkom bij Beginning Small Basic!")

Merk op dat er in het tekstvenster van het actieve programma een bericht is met de tekst **Welcome to Beginning Small Basic!** . De bovenstaande coderegel drukte dat bericht af. Deze coderegel gebruikt de methode TextWindow **WriteLine** om de taak uit te voeren. We zeggen dat de tekst "Welcome to Beginning Small Basic!" wordt doorgegeven aan de **WriteLine-methode** - de invoer wordt tussen haakjes geplaatst - wat vervolgens resulteert in de invoertekst die in het tekstvenster wordt geschreven.

Hoewel dit een zeer kort, zeer eenvoudig programma is, illustreert het enkele belangrijke componenten in een Small Basic-programma. We willen een programmakoptekst en de juiste opmerkingen. Met behulp van eigenschappen en methoden kunnen we informatie weergeven met behulp van het ingebouwde Small Basic-object, het **TextWindow**.

Enkele regels voor kleine basisprogrammering

Laten we nog een keer naar de **welkomstcode** kijken om te wijzen op enkele basisregels van Small Basic-programmering. Hier is die code:

```
1. '
2. ' Welcome Program
3. ' Beginning Small Basic
4. '
5. TextWindow.Title = "Welcome Program"
6. TextWindow.WriteLine("Welcome to Beginning Small Basic!")
```

And, here's the rules:

- Small Basic code requires perfection. All keywords must be spelled correctly. If you type **WriteLne** instead of **WriteLine**, a human may know what you mean, but a computer won't.
- Small Basic is not case-sensitive, meaning upper and lower case letters are considered to be the same characters. That means **writeline** and **WriteLine** are the same. But, even though Small Basic is not case-sensitive, it is good practice to use accepted case conventions in programming.
- Small Basic ignores any "**whitespace**" such as blanks. We will often use white space to make our code more readable to humans.
- To set an object property, we use this 'dot' convention:

  ObjectName.PropertyName = PropertyValue

  where **ObjectName** is the object, **PropertyName** the property and **PropertyValue** the value you want to establish.

- To invoke an object method, use this convention:

  ObjectName.MethodName(MethodInputs)

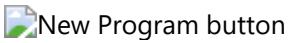
  where **ObjectName** is the object, **MethodName** the method and **MethodInputs** the inputs needed by the method.

We'll learn a lot more Small Basic programming rules as we progress.

Creating Small Basic Programs

In Class 3, we will begin learning the Small Basic language and start writing our own Small Basic programs. In preparation for this, you'll need to know how to create a new program with Small Basic. Let's do that now. What we'll do is re-create the **Welcome** program.

If it's not already running, start **Small Basic**. Click the **NewProgram** button in the toolbar:



An empty editor will appear:

Click in this window and start typing in the code for the **Welcome** program.

Type one line at a time, paying close attention that you type everything as shown (pay attention to the rules seen earlier). After each line, press the <Enter> key. Here, again, is the code.

```
'
' Welcome Program
' Beginning Small Basic
'
TextWindow.Title = "Welcome Program"
TextWindow.WriteLine("Welcome to Beginning Small Basic!")
```

After typing the four comment lines and starting to type the first line of code you will notice this popup appears:

Small Basic has a feature called "intellisense" that helps you type your programs faster. When this list appears, you move through the list using the up/down arrow keys and make a selection by pressing <Enter>. It will appear for object names, properties and methods. Give it a try!

Also notice as soon as you type **TextWindow**, this appears in the help area of the Small Basic environment:

Small Basic provides “context-sensitive” help. The help area will always display information it deems is important to the user at the appropriate time. In this case, information concerning the properties (marked by painter’s palette icon) and methods (marked by gear icon) for the TextWindow object are displayed. And, once you select a property or method, a help description for that selection appears. For example, once you type **Title**, you will see this help screen describing the property and how its used:

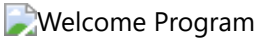
With intellisense and context-sensitive help, you always have on-line information to help you with your programming tasks.

Notice these editing buttons in the toolbar:

If you’ve ever used a word processor, these tasks are familiar to you. When typing code, you can **Cut**, **Copy** and **Paste**. And you can **Undo** and **Redo** tasks. These tasks make typing code (especially long programs) much easier in the Small Basic environment. Another thing to notice is that the editor uses different colors for different things in the code. Comments, objects, method names and data used by objects are all colored differently. This coloring sometimes helps you identify mistakes you may have made in typing.

When done typing, you should see:

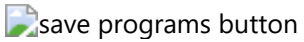
Try running your program. Use the toolbar run program button (or press <**F5**>). You should once again see the **Welcome to Beginning Small Basic!** Message:



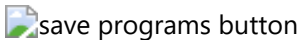
You should also see that it’s really kind of easy to get a Small Basic program up and running.

Saving Small Basic Programs

Before leaving Small Basic, we need to discuss how to save programs we create. Each program should be saved in its own folder. You decide where you want to store this folder. There are two buttons on the toolbar used to save programs. To save a new program, click the **SaveProgram** button:

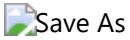


A dialog box will appear:



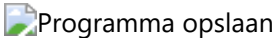
Create the folder you want to save the program in. In this example, I have created a folder named **Welcome** in a **MySmallBasic** folder.

Select a name for your program. Here I have selected **Welcome**:


Save As

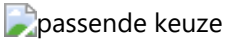
Click **Save** and the program is saved. From this point on, whenever you reopen this program and make modifications, if you click the **Save** toolbar button, the program will be automatically saved with the same name in the same folder. We suggest you do this occasionally while modifying a program.

Als u een andere naam wilt toewijzen aan of een andere map wilt maken voor het gewijzigde programma, gebruikt u de werkbalkknop **Opslaan als**:


Programma opslaan

Gebruik de resulterende dialoogvensters om uw programma een naam te geven en te zoeken.

Als u Small Basic probeert af te sluiten en geen programma's hebt opgeslagen, verschijnt Small Basic een dialoogvenster om u hiervan op de hoogte te stellen en u de mogelijkheid te geven bestanden op te slaan voordat u afsluit. Een voorbeelddialoogvenster is:

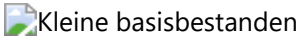

passende keuze

Maak de juiste keuze. Klik op "Ja" om uw programma op te slaan als u de wijzigingen die u erin hebt aangebracht wilt behouden.

Kleine basisbestanden

Wanneer u een Small Basic-programma in een bepaalde map opslaat, worden andere bestanden dan het bestand met uw code opgeslagen. Deze bestanden zijn nodig voor de Small Basic-omgeving om dingen bij te houden.

Ga met Deze computer of Windows Verkenner in Windows naar de map met het **welkomstprogramma dat** u zojuist hebt gemaakt en uitgevoerd. U zou de volgende bestanden moeten zien:


Kleine basisbestanden

Het bestand met de naam **Welcome** met het type **Small Basic Program** is de broncode die wordt weergegeven in de editor van Small Basic. Het **welkomstbestand** met **de vermelding Application** is een 'gecompileerde' versie van de code en is de 'uitvoerbare' code. Als u dubbelklikt op dit bestand, wordt het welkomstprogramma onafhankelijk van de small basic-omgeving uitgevoerd. Probeer het als je wilt. Later zullen we leren hoe u dit bestand kunt gebruiken om uw vrienden uw programma's op hun computers of zelfs op internet te laten uitvoeren! Het **Welcome.pdb-bestand** is een databasebestand met informatie die uw

programma nodig heeft en ten slotte wordt **SmallBasicLibrary.dll** een runtime-bibliotheek genoemd. Het bevat bestanden die uw programma helpen draaien.

We beschrijven deze bestanden zodat u op de hoogte bent van hun aanwezigheid. Wijzig of verwijder geen van deze bestanden buiten de Small Basic-omgeving.

Samenvatting

Na al het downloaden en installeren in de eerste klas, moet deze tweede klas een fluitje van een cent hebben geleken. In deze les hebben we gekeken naar verschillende belangrijke concepten waarmee we onze eigen Small Basic-programma's kunnen gaan bouwen.

In deze klas bestudeerden we de structuur van een programma, wetende dat het is gebouwd met behulp van objecten, eigenschappen en methoden. We hebben geleerd hoe we Small Basic kunnen gebruiken om een nieuw programma te maken en uit te voeren. We hebben kort gekeken naar enkele van de regels die worden gebruikt bij het schrijven van Small Basic-code en we zagen hoe we een programma konden opslaan. In de volgende les gaan we eindelijk aan de slag met het leren van de Kleine Basistaal. En we zullen ons eerste Small Basic-programma schrijven en uitvoeren.

Volgende hoofdstuk > >