

Inmiddels weet je al goed hoe je variabelen kunt gebruiken. Je hebt al veel geleerd en hopelijk vind je het nog steeds leuk.

Laten we het eerste programma met variabelen dat we hebben geschreven nog eens opnieuw bekijken:

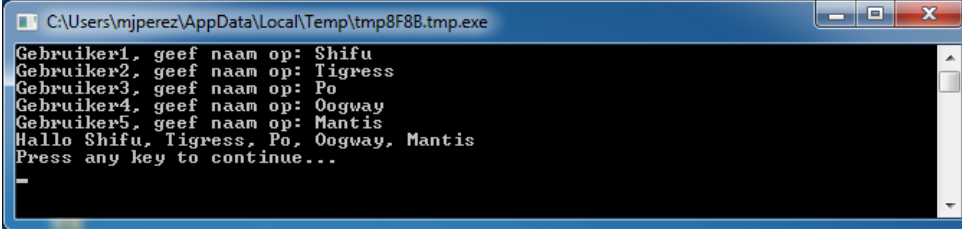
```
TextWindow.Write("Geef je naam op: ")
naam = TextWindow.Read()
TextWindow.WriteLine("Hallo " + naam)
```

In dit programma hebben we de naam van de gebruiker ontvangen en opgeslagen in de variabele **naam**. Later hebben we de gebruiker begroet met “Hallo”. Laten we in dit voorbeeld aannemen dat er meerdere gebruikers zijn, zeg er zijn 5. Hoe kunnen we al hun namen opslaan? Hier volgt één manier hoe we dit kunnen doen:

```
TextWindow.Write("Gebruiker1, geef naam op: ")
naam1 = TextWindow.Read()
TextWindow.Write("Gebruiker2, geef naam op: ")
naam2 = TextWindow.Read()
TextWindow.Write("Gebruiker3, geef naam op: ")
naam3 = TextWindow.Read()
TextWindow.Write("Gebruiker4, geef naam op: ")
naam4 = TextWindow.Read()
TextWindow.Write("Gebruiker5, geef naam op: ")
naam5 = TextWindow.Read()
```

```
TextWindow.Write("Hallo ")
TextWindow.Write(naam1 + ", ")
TextWindow.Write(naam2 + ", ")
TextWindow.Write(naam3 + ", ")
TextWindow.Write(naam4 + ", ")
TextWindow.WriteLine(naam5)
```

Als je dit programma uitvoert, zie je het volgende resultaat:



```
C:\Users\mjiperez\AppData\Local\Temp\tmp8F8B.tmp.exe
Gebruiker1, geef naam op: Shifu
Gebruiker2, geef naam op: Tigress
Gebruiker3, geef naam op: Po
Gebruiker4, geef naam op: Oogway
Gebruiker5, geef naam op: Mantis
Hallo Shifu, Tigress, Po, Oogway, Mantis
Press any key to continue...
```

Afbeelding 49 – Zonder het gebruik van matrices

Het is duidelijk dat er een betere manier moet zijn om een dergelijk eenvoudig programma te schrijven. Vooral omdat de computer heel goed is in het uitvoeren van herhalende taken. Het is niet nodig om dezelfde code voor elke nieuwe gebruiker te schrijven. De truc hier is om met dezelfde variabele de naam van meerdere gebruikers op te slaan en op te halen. Als we dit kunnen doen, kunnen we een **For**-lus gebruiken die we in vorige lessen hebben geleerd. En hier kunnen we gebruikmaken van matrices.

Wat is een matrix?

Een matrix is een speciale variabele die meer dan een waarde tegelijkertijd kan opslaan. Dit betekent dat we **naam1**, **naam2**, **naam3**, **naam4** en **naam5** niet hoeven te maken om deze vijf gebruikersnamen op te slaan, maar dat we met **naam** alle vijf gebruikersnamen kunnen opslaan. We kunnen meerdere waarden opslaan met gebruik van een 'index'. Bijvoorbeeld, **naam[1]**, **naam[2]**, **naam[3]**, **naam[4]** en **naam[5]** kunnen elk een waarde opslaan. De getallen 1, 2, 3, 4 en 5 worden *indexen* bij de matrix genoemd.

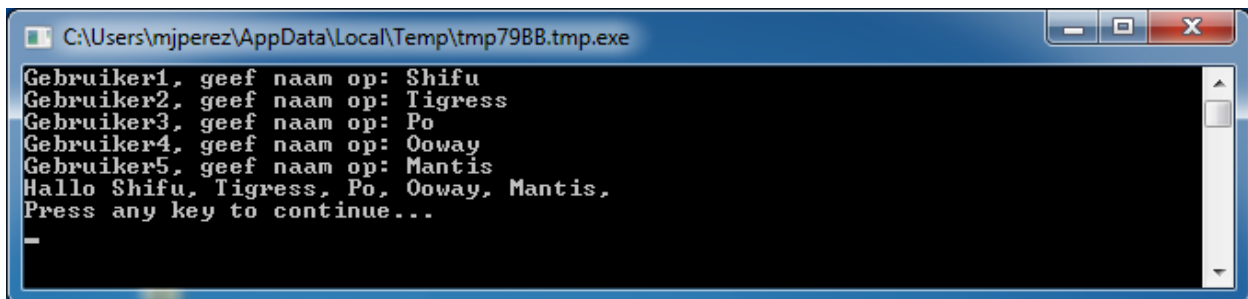
Het lijkt alsof **naam[1]**, **naam[2]**, **naam[3]**, **naam[4]** en **naam[5]** verschillende variabelen zijn, maar in werkelijkheid vormen ze samen één variabele. En je kunt je afvragen wat het voordeel hier van is. Het grootste voordeel van het opslaan van waarden in een matrix is dat je de index kunt specificeren met een andere variabele, waarmee je gemakkelijk toegang krijgt tot matrices binnen lussen.

Laten we nu eens kijken hoe we met deze nieuwe kennis ons vorig programma met matrices kunnen herschrijven.

```
For i = 1 To 5
    TextWindow.Write("Gebruiker" + i + ", geef naam op: ")
    naam[i] = TextWindow.Read()
EndFor

TextWindow.Write("Hallo ")
For i = 1 To 5
    TextWindow.Write(naam[i] + ", ")
EndFor
TextWindow.WriteLine("")
```

Dit is veel gemakkelijker te lezen of niet dan? Let op de twee vetgedrukte regels. Met de eerste regel wordt er een waarde in de matrix opgeslagen en met de tweede wordt de waarde van de matrix gelezen. De waarde die je opslaat in **naam[1]** zal de waarde die je hebt opgeslagen in **naam[2]** niet veranderen. Voor de meeste doeleinden kunnen je daarom **naam[1]** en **naam[2]** beschouwen als twee verschillende variabelen met dezelfde identiteit.



```
C:\Users\mjperrez\AppData\Local\Temp\tmp798B.tmp.exe
Gebruiker1, geef naam op: Shifu
Gebruiker2, geef naam op: Tigress
Gebruiker3, geef naam op: Po
Gebruiker4, geef naam op: Ooway
Gebruiker5, geef naam op: Mantis
Hallo Shifu, Tigress, Po, Ooway, Mantis,
Press any key to continue...
```

Afbeelding 50 – Matrices gebruiken

In het bovenstaande programma is het resultaat precies hetzelfde als in het programma zonder matrices. Het enige verschil is de komma aan het einde van *Mantis*. We kunnen dit oplossen door de printloop als volgt te herschrijven:

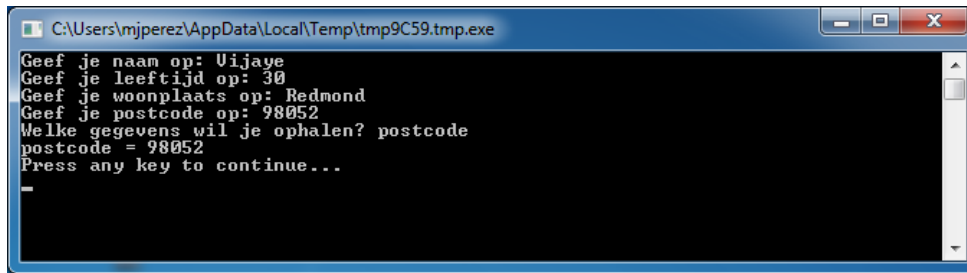
```
TextWindow.Write("Hallo ")
For i = 1 To 5
    TextWindow.Write(naam[i])
    If i < 5 Then
        TextWindow.Write(", ")
    EndIf
EndFor
TextWindow.WriteLine("")
```

Een matrix indexeren

In ons vorig programma heb je gezien dat we getallen hebben gebruikt als indexen om waarden van de matrix op te slaan en op te halen. Indexen worden echter niet alleen gebruikt voor getallen. In de praktijk is het erg nuttig om ook tekstuele indexen te gebruiken. In het volgende programma gaan we bijvoorbeeld informatie over een gebruiker opslaan en vervolgens laten we de computer alleen de gegevens naar het tekstvenster te schrijven waar de gebruiker om vraagt.

```
TextWindow.Write("Geef je naam op: ")
gebruiker["naam"] = TextWindow.Read()
TextWindow.Write("Geef je leeftijd op: ")
gebruiker["leeftijd"] = TextWindow.Read()
TextWindow.Write("Geef je woonplaats op: ")
gebruiker["woonplaats"] = TextWindow.Read()
TextWindow.Write("Geef je postcode op: ")
gebruiker["postcode"] = TextWindow.Read()

TextWindow.Write("Welke gegevens wil je ophalen? ")
index = TextWindow.Read()
TextWindow.WriteLine(index + " = " + gebruiker[index])
```



Afbeelding 51 – Niet-numerieke indexen gebruiken

Meer dan een dimensie

Zeg dat je de naam en het telefoonnummer van al je vrienden wilt opslaan om hun telefoonnummers op te kunnen zoeken wanneer je die nodig hebt. Net als in een telefoonboek. Hoe gaan we een dergelijk programma schrijven?

In dit geval zijn er twee series indexen (ook bekend als de dimensie van de matrix). Laten we aannemen dat we elke vriend kunnen herkennen aan zijn of haar bijnaam. Dit wordt onze eerste index in de matrix. Nadat we onze eerste index hebben gebruikt voor het ophalen van onze vriendvariabele, gebruiken we de tweede index, **naam** en **telefoonnummer** om de daadwerkelijke naam en telefoonnummer van die vriend op te halen.

Matrixindexen zijn niet hoofdlettergevoelig. Net zoals bij gewone variabelen hoeven de hoofdletters van de overeenkomsten van matrixindexen niet precies overeen te komen.

Deze gegevens worden op de volgende manier opgeslagen:

```
vrienden["Rob"]["Naam"] = "Robert"
vrienden["Rob"]["Telefoon"] = "06-34345658"

vrienden["VJ"]["Naam"] = "Vijaye"
vrienden["VJ"]["Telefoon"] = "06-41732679"

vrienden["Ash"]["Naam"] = "Ashley"
vrienden["Ash"]["Telefoon"] = "06-21345678"
```

Aangezien we twee indexen hebben in dezelfde matrix, **vrienden**, wordt dit een tweedimensionale matrix genoemd.

Als we dit programma eenmaal hebben ingesteld, kunnen we de bijnaam van onze vrienden als invoer gebruiken en vervolgens de informatie die we over ze hebben opgeslagen door de computer naar het tekstvenster laten schrijven. Hier is het volledige programma dat dit doet:

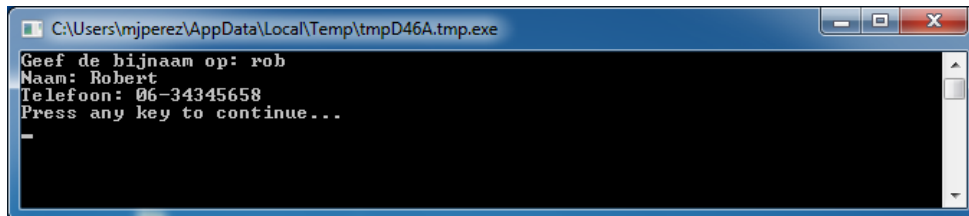
```
vrienden["Rob"]["Naam"] = "Robert"
vrienden["Rob"]["Telefoon"] = "06-34345658"

vrienden["VJ"]["Naam"] = "Vijaye"
vrienden["VJ"]["Telefoon"] = "06-41732679"

vrienden["Ash"]["Naam"] = "Ashley"
vrienden["Ash"]["Telefoon"] = "06-21345678"

TextWindow.Write("Geef de bijnaam op: ")
bijnaam = TextWindow.Read()

TextWindow.WriteLine("Naam: " + vrienden[bijnaam]["Naam"])
TextWindow.WriteLine("Telefoon: " + vrienden[bijnaam]["Telefoon"])
```



Afbeelding 52 – Een eenvoudig telefoonboek

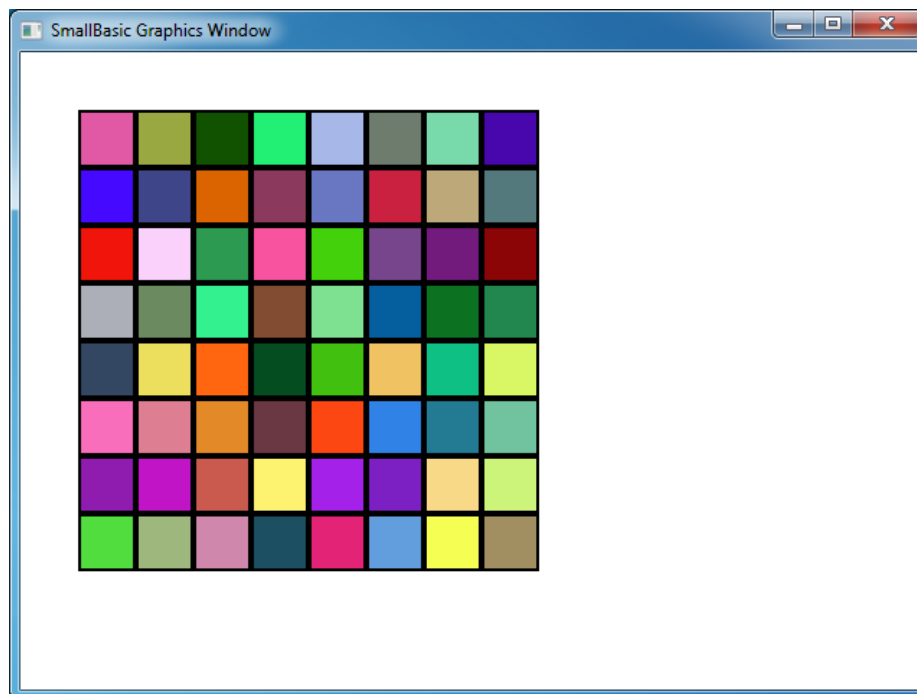
Matrices gebruiken voor rasters

Multidimensionale matrices worden veel gebruikt voor het maken van rasters/tabellen. Rasters hebben rijen en kolommen die goed passen in een tweedimensionale matrix. Hieronder wordt een eenvoudig programma weergegeven waarin vakken in een raster worden gerangschikt:

```
rijen = 8
kolommen = 8
afmeting = 40

For r = 1 To rijen
  For k = 1 To kolommen
    GraphicsWindow.BrushColor = GraphicsWindow.GetRandomColor()
    vakken[r][k] = Shapes.AddRectangle(afmeting, afmeting)
    Shapes.Move(vakken[r][k], k * afmeting, r * afmeting)
  EndFor
EndFor
```

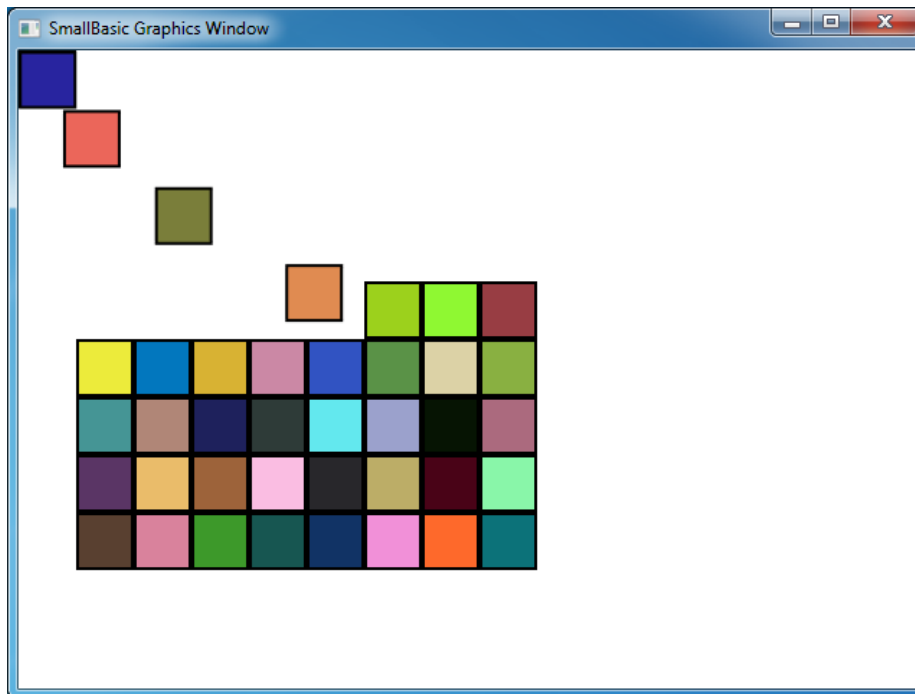
Met dit programma worden rechthoeken in een raster van 8x8 geplaatst. Naast dat deze vakken worden gerangschikt, worden ze ook opgeslagen in een matrix. Hiermee kunnen we de vakken gemakkelijk opnieuw ophalen als we de ze later opnieuw nodig hebben.



Afbeelding 53 – Vakken in een raster rangschikken

Als je bijvoorbeeld de volgende code toevoegt aan het einde van het vorige programma, worden de vakken naar de linkerbovenhoek verplaatst.

```
For r = 1 To rijen
  For k = 1 To kolommen
    Shapes.Animate(vakken[r][k], 0, 0, 1000)
    Program.Delay(300)
  EndFor
EndFor
```



Afbeelding 54 – De vakken in het raster bijhouden