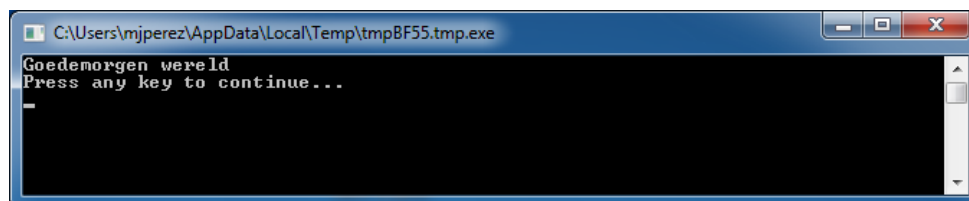


Voorwaarden en vertakkingen

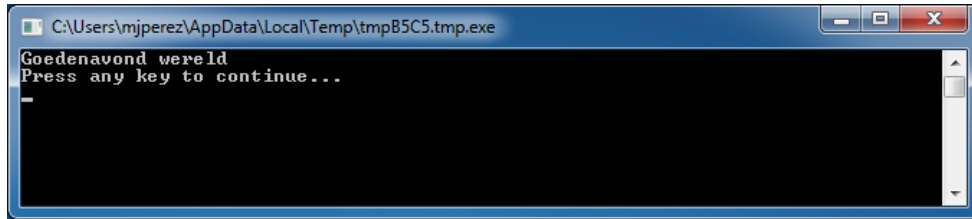
Laten we eens teruggaan naar ons eerste programma. Zou het niet leuk zijn als we in plaats van het algemene *Hallo wereld*, afhankelijk van de tijd van de dag, *Goedemorgen wereld* of *Goedenavond wereld* konden zeggen? In het volgende programma laten we de computer *Goedemorgen wereld* zeggen als het eerder is dan 12:00 uur en *Goedenavond* als het later is dan 12:00 uur.

```
If (Clock.Hour < 12) Then
    TextWindow.WriteLine("Goedemorgen wereld")
EndIf
If (Clock.Hour >= 12) Then
    TextWindow.WriteLine("Goedenavond wereld")
EndIf
```

Afhankelijk van wanneer je het programma uitvoert, zie je een van de volgende resultaten:



Afbeelding 14 – Goedemorgen wereld



Afbeelding 15 – Goedenavond wereld

Laten we de eerste drie regels van het programma eens analyseren. Je hebt al ontdekt dat je met deze regels de computer verteld dat als `Clock.Hour` minder is dan 12, de woorden "Goedemorgen wereld" naar het tekstvenster worden geschreven. De woorden **If**, **Then** en **EndIf** zijn speciale woorden die de worden geïnterpreteerd wanneer het programma wordt uitgevoerd. Het woord **If** wordt altijd gevolgd door een voorwaarde. In dit geval is dit (**`Clock.Hour < 12`**). Vergeet niet dat de haakjes noodzakelijk zijn. De computer zal anders je bedoelingen niet begrijpen. De voorwaarde wordt gevolgd door **then** en de daadwerkelijke bewerking die moet worden uitgevoerd. En na de bewerking komt **EndIf**. Hiermee wordt aangegeven dat de conditionele uitvoering is afgerond.

In Small Basic kun je het klok-object gebruiken voor toegang tot de huidige datum en tijd. Je beschikt ook over een aantal eigenschappen waarmee je de dag, de maand, het jaar, het uur, de minuut en de seconde van dit moment onafhankelijk van elkaar kunt verkrijgen.

Tussen **then** en **EndIf** kunnen meer dan een bewerking worden geplaatst en de computer zal ze allemaal uitvoeren als de voorwaarde geldig is. Je kunt bijvoorbeeld het volgende schrijven:

```
If (Clock.Hour < 12) Then
    TextWindow.Write("Goedemorgen. ")
    TextWindow.WriteLine("Hoe was het ontbijt?")
EndIf
```

Else

In het eerste programma in dit hoofdstuk, is het je misschien opgevallen dat de tweede voorwaarde min of meer overbodig is. De waarde **`Clock.Hour`** kan ofwel minder dan 12 zijn of niet. De tweede controle is niet echt nodig. Voor zulke gevallen kunnen we de twee instructies **if..then..endif** samenvoegen tot één met een nieuw woord, **else**.

Als we het programma herschrijven met **else**, ziet het er als volgt uit:

```
If (Clock.Hour < 12) Then
    TextWindow.WriteLine("Goedemorgen wereld")
Else
    TextWindow.WriteLine("Goedenavond wereld")
EndIf
```

Dit programma doet precies hetzelfde als het andere programma en dit brengt ons tot een belangrijke les in programmeren:

“ *Binnen programmeren zijn er meestal vele manieren om hetzelfde te doen. Soms is een bepaalde manier logischer dan de andere. Het is aan de programmeur om de keuze te maken. Naargelang je meer programma's schrijft en meer ervaring opdoet, zul je deze verschillende technieken en hun voor- en nadelen beginnen te ontdekken.*

Inspringen

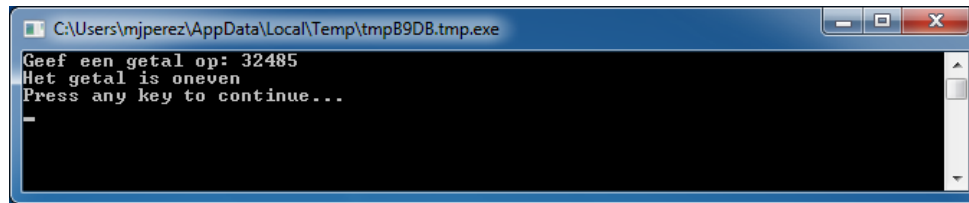
In alle voorbeelden kun je zien hoe de instructies tussen *If Else* en *EndIf* worden ingesprongen. Dit inspringen is niet noodzakelijk. Het programma wordt ook zonder deze inspringingen begrepen. De structuur van het programma is echter wel gemakkelijker te zien en te begrijpen. Vandaar dat het normaal gesproken een goede gewoonte is om instructies tussen zulke blokken in te springen.

Even of oneven

Nu dat we de instructies **If..Then..Else..EndIf** onder de knie hebben, gaan we een programma schrijven dat van een bepaald getal kan aangeven of het even of oneven is.

```
TextWindow.Write("Geef een getal op: ")
getal = TextWindow.ReadNumber()
rest = Math.Remainder(getal, 2)
If (rest = 0) Then
    TextWindow.WriteLine("Het getal is even")
Else
    TextWindow.WriteLine("Het getal is oneven")
EndIf
```

En als je dit programma uitvoert, ziet de uitvoer er als volgt uit:



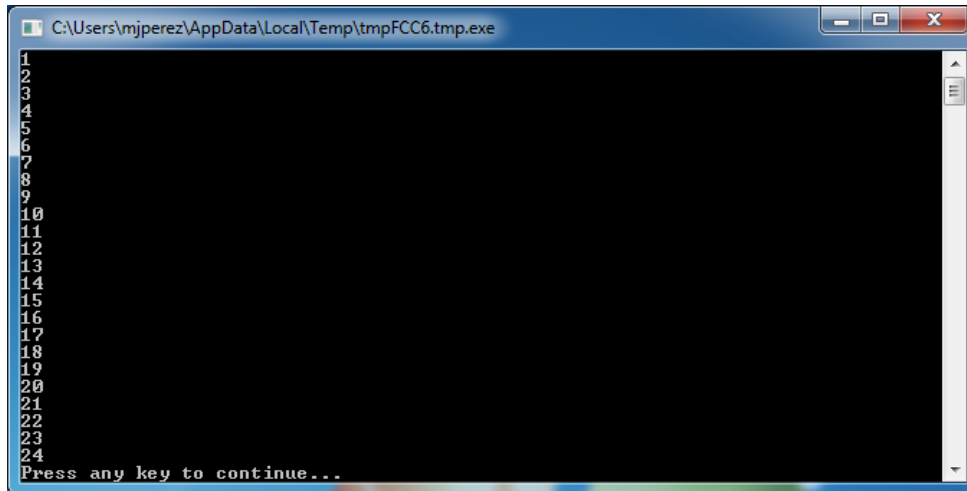
Afbeelding 16 – Even of oneven

In dit programma hebben we een andere nuttige bewerking geïntroduceerd, **Math.Remainder**. En zoals je misschien al hebt begrepen, met **Math.Remainder** wordt het eerste getal door het tweede getal gedeeld en wordt de rest vervolgens teruggegeven.

Vertakking

Misschien kun je je nog herinneren dat we in het tweede hoofdstuk hebben geleerd dat de computer tijdens de uitvoering van een programma slechts één instructie tegelijkertijd, op volgorde en van boven naar beneden, kan verwerken. Er is echter een speciale instructie waarmee de computer niet in volgorde naar een andere instructie kan springen. Laten we eens kijken naar het volgende programma.

```
i = 1
start:
TextWindow.WriteLine(i)
i = i + 1
If (i < 25) Then
    Goto start
EndIf
```



```
C:\Users\mjperrez\AppData\Local\Temp\tmpFCC6.tmp.exe
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
Press any key to continue...
```

Afbeelding 17 – Goto gebruiken

In het programma hierboven hebben we de waarde 1 toegewezen aan de variabele *i*. En vervolgens hebben we een nieuwe instructie toegevoegd die eindigt op een dubbele punt (:)

```
start:
```

Dit wordt een *label* genoemd. Labels lijken op bladwijzers die de computer kan begrijpen. Je kunt de bladwijzer elke willekeurige naam geven en je kunt zoveel labels aan een programma toevoegen als je wilt. Alle namen moeten echter wel uniek zijn.

Een andere interessante instructie hier is:

```
i = i + 1
```

Hiermee vertel je de computer om de waarde 1 bij de variabele *i* op te tellen en deze waarde weer terug te geven aan *i*. Dus als de waarde van *i* 1 was voor deze instructie, dan wordt de waarde 2 nadat deze instructie wordt uitgevoerd.

En ten slotte,

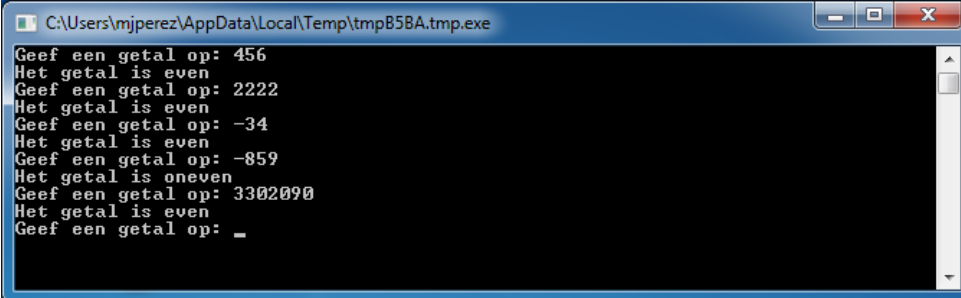
```
If (i < 25) Then
    Goto start
EndIf
```

Met dit onderdeel vertel je de computer dat er moet worden begonnen met het uitvoeren van instructies van de bladwijzer **start** als de waarde van *i* minder is dan 25.

Eindeloze uitvoering

Met de instructie **Goto** kun je de computer iets een willekeurig aantal keren laten herhalen. Je kunt bijvoorbeeld het programma Even en oneven aanpassen zoals hieronder wordt aangegeven en het programma eindeloos uitvoeren. Je kunt het programma stoppen door op de knop voor sluiten (X) te klikken in de rechterbovenhoek van het venster.

```
begin:  
TextWindow.Write("Geef een getal op: ")  
getal = TextWindow.ReadNumber()  
rest = Math.Remainder(getal, 2)  
If (rest = 0) Then  
    TextWindow.WriteLine("Het getal is even")  
Else  
    TextWindow.WriteLine("Het getal is oneven")  
EndIf  
Goto begin
```



The screenshot shows a Windows command prompt window titled "C:\Users\vmjpervez\AppData\Local\Temp\tmp858A.tmp.exe". The window contains the following text, which is the output of the program shown in the code block above:

```
Geef een getal op: 456  
Het getal is even  
Geef een getal op: 2222  
Het getal is even  
Geef een getal op: -34  
Het getal is even  
Geef een getal op: -859  
Het getal is oneven  
Geef een getal op: 3302090  
Het getal is even  
Geef een getal op: _
```

Afbeelding 18 – Even of oneven eindeloos uitvoeren