



In deze eerste les van module 4 leer je over het concept lus ('loop' in het Engels) en over de structuur en het gebruik van de **For(...)** lus.

#### Doelen:

- Lussen begrijpen.
- De **For(...)**-lus gebruiken om een lijst met waarden te genereren.

**Docenten Tip:** Er zijn drie fundamentele lussen in TI-Basic: **For**, **While**, en **Repeat**. Een lus structuur geeft een programma de mogelijkheid om een serie opdrachten steeds weer uit te voeren, ofwel met iteraties (herhalingen) voor een reeks van waarden (zoals in de **For**-lus) ofwel tot aan een specifieke voorwaarde is voldaan (of niet) zoals in **While** en **Repeat**. In deze module wordt in elke les één van deze structuren behandeld. Het is ook mogelijk om de 'achterhaalde' opdrachten **Lbl** en **Goto** te gebruiken om een lust te maken, maar dat leidt tot slechte gewoontes en kan programma-fouten veroorzaken als het niet op de juiste manier wordt toegepast. We vermijden daarom verwijzingen naar deze opdrachten volledig, hoewel de opdracht **Lbl** wel wordt gebruikt en zelfs vereist is in combinatie met de opdracht **Menu** om aangepaste menu's binnen een programma te ontwerpen. Voor voorwaardelijke opdrachten en lussen is daarentegen de **Goto** opdracht helemaal niet nodig.

Programma's kunnen ingewikkeld worden omdat het nodig wordt de verschillende besturingsstructuren (If-opdrachten en lussen) te mengen in een programma om op een goede manier te kunnen werken met complexere algoritmes. Dit is wat programmeren LEUK maakt!

### Lussen (loops)

Een lus is een methode om een serie opdrachten te herhalen. Alle programmeertalen hebben minstens één lus-structuur. De lusstructuur heeft een manier om terug te gaan naar een eerdere plek in het programma. TI-Basic heeft drie verschillende soorten lussen. Een oneindige lus stopt nooit.

Om een lopend programma te onderbreken (stoppen) druk je op **[ON]**. Je ziet dan de opties 'Afsluiten' en 'Ga Naar. Met 'Afsluiten' ga je terug naar het beginscherm 'Ga Naar brengt je in de programma-editor naar de plek waar het programma is gestopt.

De drie lussen in TI-Basic zijn:

**For( ) ... End**

**While <voorwaarde is waar> ... End**

**Repeat <totdat voorwaarde waar is > ... End**

De rest van deze les gaat alleen over de **For( )** lus.

```

NORM DRIJF AUTO REZEL RAD 44N
PROGRAM: INF
:While 1>0
:Disp "ONEINDIG EN VERDER
"
:End
:

```

Een oneindige lus. Waarom?

```

NORM DRIJF AUTO REZEL RAD 44N
I/O KLEUR UITVOEREN HU
If
2:Then
3:Else
4:For(
5:While
6:Repeat
7:End
8:Pause
9\Lbl

```

**De For( ... ) lus**

Sructuur: **For**(variabele, startwaarde, stopwaarde)

Lus kern

**End**

Voorbeeld: **For**(A,1,10)

**Disp** A            ← luskern

**End**

*Opmerking:*

*De opdracht **For**( ) vereist een variabele (de variabele die de lus aanstuurt), een startwaarde en een stopwaarde gescheiden door komma's. De start- en stopwaarden kunnen variabelen zijn. De kern van de lus kan zoveel opdrachten bevatten als nodig, maar deze moeten de variabele die de lus aanstuurt niet veranderen. De lus wordt uitgevoerd van jouw startwaarde tot jouw stopwaarde met een standaard stapgrootte van 1.*

```
NORM DRIJF AUTO REEEL RAD MN
PROGRAM: TEL10
:For(A,1,10)
:Disp A
:End
```

```
NORM DRIJF AUTO REEEL RAD MN
3
4
5
6
7
8
9
10
.....Klaar.
```

**Docenten Tip:** De lus eindigt wanneer de variabele de stopwaarde *overschrijdt*. Als je de opdracht **Disp A** uitvoert nadat de lus is geëindigd dan zal je zien dat A de waarde 11 heeft en niet 10.

**For-lus met stapgroottes anders dan 1**

Er is een optioneel vierde argument voor de opdracht **For**( ): de *stapgrootte*. De stapgrootte is de waarde waarmee de variabele die de lus aanstuurt toeneemt na elke iteratie (herhaling) van de lus. De standaardwaarde is 1.

```
NORM DRIJF AUTO REEEL RAD MN
PROGRAM: TEL10
:For(A,1,10,3)
:Disp A
:End
:
```

**Docenten Tip:** Als de stapgrootte van de lus negatief is dan stopt de lus wanneer de lus-variabele *kleiner* is dan de stopwaarde.

**For(A,1,10,3)** begint met **A=1**, telt dan 3 op bij **A** elke keer dat de lus zich herhaalt. De lus stopt wanneer **A** groter is dan 10. De stapgrootte kan een negatief getal zijn.

**For(B,10,0,-1)** stelt af van 10 naar 0.

```
NORM DRIJF AUTO REEEL RAD MN
PRGM: TEL10
1
4
7
10
.....Klaar.
```



# 10 minuten programmeren

## TI-84 PLUS SERIE

### Programmeren met For(...)

Laten we een programma schrijven dat een tabel laat zien met getallen en hun kwadraten.

De gebruiker kan de onder- en bovengrens van de serie getallen invoeren.

Het lastige deel is om de paren getallen op dezelfde regel af te beelden! We kunnen dit doen door **lijsten** te gebruiken..

*Opmerking:*

**L** en **U** worden gebruikt om onder (**L**ower) en Boven (**U**pper) aan te geven. De opdracht **For( )** gebruikt de waarden van **L** en **U**. De lijsthaakjes (accoladen) staan onder de haakjestoets. Druk op ` [ ] en ` [ ] voor de haakjes.

## MODULE 4: OEFENBLAD 1

### DOCENTENHANDLEIDING

```
PROGRAM: KWADRATE
:WisHome
:Input "ONDER",0
:Input BOVEN",B
:For(A,0,B)
:Disp (A,A^2)
:End
:.
```

**Docenten Tip:** Een lijst weergeven is een handige manier om meer dan één waarde op een regel op het beginscherm weer te geven.

Voer het programma uit en voer een onder- en bovengrens in voor de tabel.

Als de lijsten te snel langskomen voeg dan eventueel een **Pause**-opdracht ins na de opdracht **Disp** en voor de opdracht **End**.

### Uitdaging:

Gebruik een **If ... Then... End**-structuur om na elke 5 paren getallen te pauzeren. Kijk even terug naar de deelbaarheidstechniek uit de vorige module.

### Voorbeeldantwoord:

```
:If (A-L+1)/5=int((A-L+1)/5)
:Pause "Druk op Enter"
```