

In deze derde les van module 4 leer je over de **Repeat...End**-lus. We zullen deze vergelijken met de **For...lus** en ook laten zien waarom deze lus krachtiger is..

Doelen:

- De structuur leren van de **Repeat...End**-lus.
- Deze vergelijken met de **While...End**-lus.
- Zien hoe **Repeat...End** wordt gebruikt bij het programmeren van de rij van Fibonacci.

Docenten Tip: Hoewel de voorwaarde voor **Repeat** (herhaal) beschreven is aan het begin van de lus, wordt deze pas getest aan het eind van de lus! Een **Repeat**-lus wordt daarom altijd minsten één keer uitgevoerd.

Er zijn zowel **While** als **Repeat** lussen in de programmeertaal om deze zich net een beetje anders gedragen.

De Repeat... End-lus

De **Repeat...End**-lus zal doorlopen zolang de <voorwaarde> ervan onwaar is. Dit is precies het *tegenovergestelde* van wat de **While**-lus doet, en dat is niet het enige verschil! De lus ziet er zo uit, wat erg lijkt op **While** structuur:

```
Repeat <voorwaarde>
  <lus kern>
End
```

De programma's rechts hebben dezelfde uitvoer. Wat zijn de verschillen?

```
NORMAL FLOAT AUTO REAL Radian MP
PROGRAM: REPEAT
: 0→A
: Repeat A>10
: Disp A
: A+1→A
: End
: █
```

De <voorwaarde> is een logische uitdrukking zoals **X>0**.

De <lus kern> is een willekeurige verzameling opdrachten, die ook lussen en **If**-structuren kan bevatten.

Deze wordt **een keer uitgevoerd** en gaat dan door **totdat** de <voorwaarde> waar is, dus de lus gedraagt zich min of meer als:

```
Repeat
  <lus kern>
  totdat <voorwaarde> waar is End
```

```
NORMAL FLOAT AUTO REAL Radian MP
PROGRAM: WHILE
: 0→K
: While K≤10
: Disp K
: K+1→K
: End
```

Maar er is geen sleutelwoord 'totdat' in TI Basic. Dit is 'impliciet', het wordt verondersteld er te zijn. Zelfs als de <voorwaarde> waar is vanaf het begin, dan wordt de kern van de lus toch één keer uitgevoerd omdat de voorwaarde oas getest wordt aan het eind van de lus.

In een **While**-lus is het een uitstekend idee om de variabele(n) waarvan de lus afhangt, vooraf te initialiseren. In een **Repeat**-lus zal dit gebeuren binnen de kern van de lus en is initialiseren dus niet nodig.

Net zoals bij **While**, zal er ergens in de <lus kern> een opdracht moeten zijn die effect heeft op de <voorwaarde> zo dat de lus uiteindelijk stopt en de opdrachten na de lus kunnen worden uitgevoerd. Gewoonlijk staat deze opdracht onderaan dicht bij het einde van de <lus kern>.

10 minuten programmeren

TI-84 PLUS SERIE

De rij van Fibonacci programmeren

Laten we een programma schrijven dat de rij van *Fibonacci* tot een bepaalde waarde, afbeeldt op het scherm. Je kunt de rij *Fibonacci* eerst onderzoeken als je er nog nooit van hebt gehoord.

De uitvoer van het programma zie je rechts. Kun je dit programma schrijven zonder hieronder af te kijken?

We beginnen met een **Prompt**-opdracht om een waarde voor de bovengrens te krijgen van de gebruiker. De eerste twee Fibonacci-getallen zijn 1 en 1 en dus zullen we deze waarde opslaan in de variabelen **A** en **B**. Deze variabelen gaan we gebruiken om de rest van de Fibonacci-getallen te berekenen (tot **N**).

Dan beginnen we de **Repeat**-lus met de voorwaarden **A>N**, wat betekent 'tot **A** groter is dan **N**'. In de lus geven we eerst de twee huidige getallen weer.

Tenslotte berekenen we de volgende twee Fibonacci-getallen en bereiken het eind (**End**) van de lus.

Deze twee laatste opdrachten van de lus laten zien dat $A+B$ wordt opgeslagen in zowel **A** als **B** en het lijkt erop dat **A** en **B** dezelfde waarde gaan krijgen. **Dit is niet het geval!** Probeer het maar met 1 en 1. De eerste opdracht slaat $1+1$ op in **A**, en geeft **A** dus de waarde 2. De tweede opdracht slaat $2+1$ op in **B** en daarmee krijgt **B** de waarde 3. Probeer het zelf maar eens – doe alsof je de computer bent!

Voer het programma uit met een aantal verschillende invoerwaarden. Werkt het zoals je had bedoeld?

Probeer nu de voorwaarde voor **Repeat** te veranderen in **B>N**. Wat is het effect? Hoe kunnen we het programma aanpassen zodat het de juiste reeks getallen laat zien en stopt wanneer exact het grootste Fibonacci-getal kleiner dan **N** is weergegeven?

MODULE 4: OEFENBLAD 3

DOCENTENHANDLEIDING

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:FIBONACC
N=?5
1
1
2
3
5
8
.....
Done.
```

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:FIBONACC
:Prompt N
:1→A
:1→B
:
```

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:FIBONACC
:Prompt N
:1→A
:1→B
:Repeat A>N
:Disp A,B
:
```

```
NORMAL FLOAT AUTO REAL RADIAN MP
PROGRAM:FIBONACC
:Prompt N
:1→A
:1→B
:Repeat A>N
:Disp A,B
:A+B→A
:A+B→B
:End
:■
```