Unit 4: Lussen | **Oefenblad 1: For-lussen**

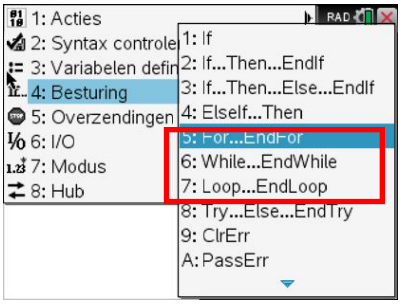
In deze les leer je over je het lus-,concept dat wordt gebruikt bij programmeren en onderzoek je de For… lus

Doelen:

- Het concept van een lus bij het programmeren beschrijven.
- Een programma en functies schrijven met de For…EndFor-structuur.

Docenten Tip: Er bestaan drie *fundamentele* lussen in TI-Nspire TI-Basic: **For**, **While** en **Loop**. Een lus-structuur geeft het programma de mogelijkheid om een groep opdrachten steeds opnieuw te verwerken, ofwel door dit te herhalen voor een serie van waarden (zoals in de **For-**lus) of totdat er aan een bepaalde voorwaarde is voldaan (of niet) zoals bij **While** en **Loop**. De lessen van unit 4 introduceren elk van deze structuren apart.

Programma's worden complex omdat het noodzakelijk is om al deze structuren (rij, **If-**opdrachten en lussen) te combineren in een programma om aan meer complexe algoritmes te kunnen voldoen. Dat maakt het programmeren juist LEUK!



Over Lussen

De TI-Basic programmeertaal maakt het mogelijk om een serie opdrachten steeds weer opnieuw te verwerken. Dit herhalen van opdrachten wordt '**lussen'** (looping) genoemd.

De drie lus-structuren die je in deze unit gaat leren zijn elk te vinden door **Menu > Besturing** te selecteren in het menu van de programma-editor. (Zie **For**…, **While**…, en **Loop**… hier rechts)

De **While…** en **Loop…** structuren zullen we in extra lessen van deze Unit verkennen. Ga voor meer informatie over de extra besturingsstructuren in dit menu naar de Handleiding op de TI website over programmeren (codes).

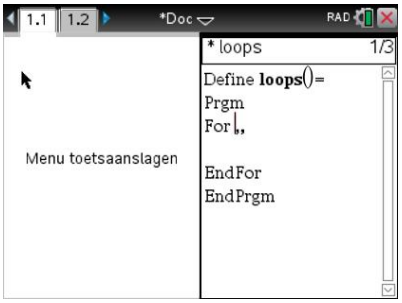
For…EndFor

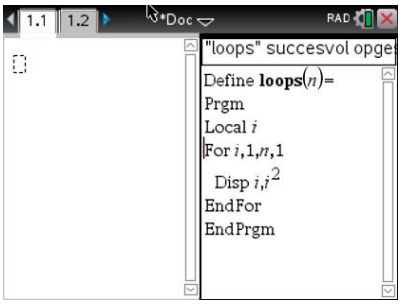
De For… lus wordt gebruikt om de waarden uit een rekenkundige rij te verwerken. Dit proces staat bekend als 'iteratie'.

Het selecteren van de For…EndFor-opdracht uit het menu Besturing geeft je de benodigde onderdelen om de rest van de structuur op te bouwen:

For , , ,

EndFor





De komma's achter het woord **For** geven aan dat je **vier** elementen moet opgeven:

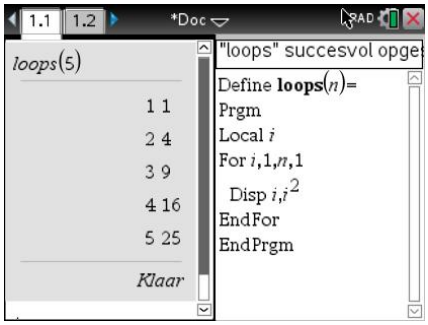
For *i*, 1, *n*, 1

1. *i* is de **besturingsvariabele van de lus**: Het eerste element moet een variabele zijn.
2. 1 is de **beginwaarde**: Elke keer dat de lus wordt verwerkt neemt deze waardes aan of hij telt het aantal lussen vanaf de gespecificeerde *startwaarde* (1).
3. *n* is de **eindwaarde**: Elke keer dat de lus wordt verwerkt neemt hij of waardes aan of telt hij het aantal lussen tot de gespecificeerde eindwaarde (n).
4. 1 is de waarde van de **stapgrootte**: Elke keer dat de lus wordt verwerkt neemt hij waardes aan of telt hij het aantal lussen van begin tot eind met de gespecificeerde waarde van de stapgrootte (1).

Elke of alle van de laatste drie elementen kunnen getallen of variabelen zijn.

Docenten Tips:
Als de stapgrootte 1 is dan kan het vierde element weggelaten worden:
For *i*, 1, n gebruikt een standaardwaarde van 1 als stapgrootte
De waarde van de stapgrootte kan elk getal zijn, zelfs een negatief getal:
For *i*, 5, -5, -1 begint bij 5 en loopt af tot -5
De berekening hoeft niet exact op de eindwaarde uit te komen. Als de waarde van *i* de over de eindwaarde heen gaat zal de lus eindigen:
For *i*, 1, 10, 6 zal de waardes 1 en 7 verwerken en dan eindigen.
Ken geen waardes toe aan de *besturingsvariabele van de lus* binnen de lus. Dit zal het verwerken van de lus verstoren en ongewenste resultaten opleveren. Normaal gesproken is de besturingsvariabele van de lus een lokale (Local) variabele

Het programma uitvoeren



Het uitvoeren van het programma dat in het plaatje hierboven is weergegeven geeft het resultaat hier rechts.

- De waarde voor *n* is een argument van het programma
- De besturingsvariabele van de lus is een lokale (local) variabele; hij heeft geen invloed op de rest van de opgave.
- De lus begint met *i*=1 en toont de waardes van *i* en i^2.
- Na de **Disp**-opdracht geeft de **EndFor** opdracht de besturing weer terug aan de **For** opdracht waarbij de stapgrootte bij *i* is opgeteld.
- Als de resulterende waarde kleiner of gelijk is aan de eindwaarde dan wordt de lus opnieuw uitgevoerd met de opgehoogde waarde voor *i*.
- Dit proces herhaalt zich totdat de eindwaarde wordt bereikt of overschreden.

Nadat de lus stopt, zal *i* groter dan de eindwaarde zijn geworden! Geloof je het niet? Voeg een extra Disp *i* opdracht toe na de EndFor-opdracht en test het maar.

Docenten Tip: Nadat de lus eindigt, is de lusvariabele groter dan de eindwaarde: