

In dieser Lektion wirst du das Konzept von Schleifen in Programmen kennen lernen.
Wir beginnen mit der For-Schleife.

Lernziele:

- Das Konzept von Schleifen in Programmen beschreiben
- Programme und Funktionen mit einer For...EndFor-Schleife schreiben

Hinweis: In TI-Nspire-Basic gibt es drei *wesentliche* Schleifen: **For**, **While** und **Loop**. Eine Schleife gibt dem Programm die Möglichkeit, eine Gruppe von Anweisungen immer wieder abzarbeiten, Dies geschieht entweder über eine Folge von Werten (in der **For**-Schleife), oder so lange bis eine besondere Bedingung erfüllt (oder nicht erfüllt) ist (in **While** und **Loop**-Schleifen). Die Übungen dieser Lektion behandeln jede dieser Schleifen einzeln.

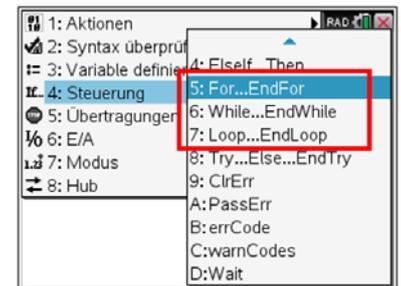
Programme werden komplex, wenn es notwendig wird, alle Kontrollstrukturen (**Schleifen** und **If**-Anweisungen) gemeinsam zu verwenden, um etwas kompliziertere Algorithmen zu behandeln. Das ist es, was Spaß und Spannung ins Programmieren bringt!

Über Schleifen

Die TI-Basic Programmiersprache bietet die Möglichkeit eine Gruppe von Anweisungen wieder und wieder abzarbeiten. Diese Wiederholung von Anweisungen nennt man **Schleifen**.

Die drei Schleifenkonstruktionen, die du jetzt lernen wirst erreicht man im Programmeditor über *menu> Steuerung* (siehe rechts: **For...**, **While...** und **Loop...**).

Die **While...** und **Loop...**-Schleifen werden in den nächsten Übungen in dieser Lektion behandelt. Für mehr Information über zusätzliche Kontrollstrukturen informiere dich bitte im Reference Guide auf der TI-Codes Webseite.



For...EndFor

Die For-Schleife dient zur Abarbeitung einer arithmetischen Folge von Werten. Man nennt einen derartigen Vorgang ‚Iteration‘.

Der Bezug der For...EndFor-Anweisung vom Steuerungsmenu gibt dir die notwendigen Bausteine, um die Anweisung richtig zu vervollständigen:

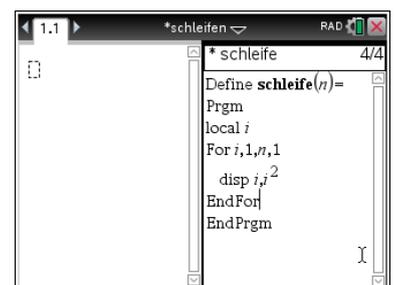
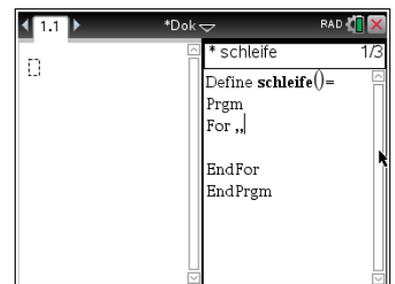
For , ,

EndFor

Die Kommata nach dem Wort **For** zeigen an, dass noch (mindestens) **drei** Angaben fehlen:

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

1. *i* ist die Lauf- oder Schleifenvariable. Hier muß eine Variable stehen.
2. 1 ist der **Anfangswert** für *i*: Nach jedem Durchlauf der Schleife wird dieser Wert erhöht oder die Zahl der durchlaufenen Schleifen (ab dem Anfangswert) gezählt (hier ab 1).





- n ist der **Endwert**: Nach jedem Durchlauf der Schleife wird der Wert der Laufvariablen mit n verglichen und so lange wiederholt, bis sie größer als n ist.
- 1 ist die **Schrittweite** (Inkrement): Nach jedem Schleifendurchlauf wird i um diesen Wert erhöht (hier ist dieser Wert 1). 1 muss als Schrittweite nicht angegeben werden, jede andere Schrittweite schon.

Jede einzelne oder alle von den letzten drei Komponenten können Zahlen oder Variable sein.

Hinweise:

Für die Schrittweite 1 kann das 4. Argument in der For-Anweisung weggelassen werden:

For $i, 1, n$ nimmt 1 als voreingestellte Schrittweite an.

Die Schrittweite kann jede Zahl, auch eine negative sein:

For $i, 5, -1$ beginnt bei 5 und geht in Schritten von 1 runter bis -5.

Die Schrittweite muss nicht genau den Endwert erreichen. Wenn der Wert von i den Endwert überschreitet, dann wird die Schleife verlassen:

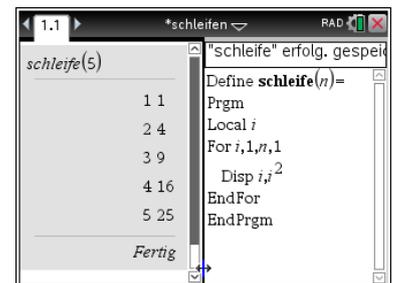
For $i, 1, 10, 6$ verarbeitet nur die Werte 1 und 7 für i .

Weise der Laufvariablen innerhalb der Schleife keine Werte zu. Dies stört den Ablauf und erzeugt unerwünschte Ergebnisse. Üblicherweise ist die Laufvariable eine lokale Variable.

Programmausführung

Das im Screenshot gezeigte Programm erzeugt das rechts abgebildete Ergebnis.

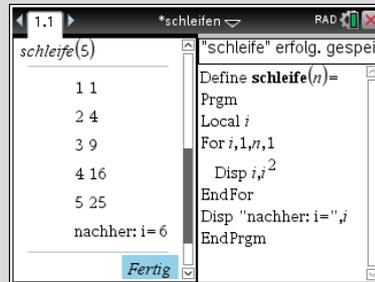
- Der Wert für n (Endwert) ist ein Programmargument.
- Die Laufvariable i ist eine lokale Variable. Sie beeinflusst nicht das übrige Problem.
- Die Schleife beginnt mit $i=1$ und erzeugt und zeigt die Werte von i und i^2 .
- Nach der **Disp**-Anweisung übergibt **EndFor** die Kontrolle zurück an die **For**-Anweisung, die i um die Schrittweite erhöht.
- Solange dieser Wert $\leq n$ ist, wird die Schleife erneut mit dem nun um die Schrittweite erhöhten Wert für i durchlaufen.
- Das wird solange wiederholt, bis der Endwert erreicht oder übertroffen wird.



Wenn die Schleife abgearbeitet ist, dann ist der Wert von i größer als der Endwert n ! Du glaubst das nicht? Dann füge ein weiteres **Disp i** nach der **EndFor**-Anweisung ein und überzeuge dich selbst.



Hinweis: Nach Verlassen der Schleife hat die Laufvariable einen Wert größer als den Endwert:



```
schleife(5)
1 1
2 4
3 9
4 16
5 25
nachher: i=6
Fertig
```

```
"schleife" erfolg. gespeichert
Define schleife(n)=
Prgm
Local i
For i,1,n,1
Disp i,i^2
EndFor
Disp "nachher: i=",i
EndPrgm
```