



Lektion 4: Schleifen	Übung 3: Die Repeat...End Schleife
----------------------	------------------------------------

In der dritten Übung von Lektion 4 geht es um die **Repeat...End** - Schleife und den Vergleich mit der **While...End** – Schleife.

- Lernziele:**
- Verstehen der **Repeat...End** - Schleife.
  - Unterschiede zur **While...End** - Schleife.
  - Verwendung von **Repeat...End** bei der Programmierung der Fibonacci - Zahlen.

**Hinweis:** Obwohl die *Bedingung* für **Repeat** am Anfang der Schleife steht, wird sie erst am Ende der Schleife überprüft. Die Befehle in der **Repeat** – Schleife werden also auf jeden Fall **einmal ausgeführt!**

Es gibt die beiden Schleifen **While** und **Repeat** in der Programmiersprache, weil sie sich ein wenig anders verhalten.

**Die Repeat... End - Schleife**

Der Befehlsblock innerhalb der **Repeat...End** – Schleife wird solange ausgeführt, wie die *Bedingung falsch* ist. Damit verhält sich **Repeat** genau entgegengesetzt zu **While**, aber das ist nicht der einzige Unterschied. Die Struktur der **Repeat** – Schleife sieht zunächst einmal genauso aus wie die der **While** – Schleife:

```
Repeat <Bedingung>
  <Befehlsblock>
End
```

Die beiden Beispielprogramme erzeugen dieselbe Ausgabe. Worin liegen die Unterschiede?

```
NORMAL FLS AUTO REELL BOGENM MP
PROGRAM: REPEAT
:0→A
:Repeat A>10
:Disp A
:A+1→A
:End
:█
```

Die *<Bedingung>* ist ein logischer Ausdruck wie z.B. **X>0**.  
 Der *<Befehlsblock>* besteht aus einer beliebigen Gruppe von Anweisungen, die auch weitere Schleifen und **If** – Befehle enthalten können. Er wird einmal ausgeführt und dann immer wieder, bis die *Bedingung wahr* ist. Die Struktur sieht also eher so aus:

```
Repeat
  <Befehlsblock>
  Bis (until) <Bedingung> wahr
End
```

Das Schlüsselwort „until“ gibt es in TI-Basic nicht, denn es ist in das **Repeat...End** implementiert. Selbst wenn die *Bedingung* beim Starten der Schleife wahr ist, wird der Befehlsblock einmal durchlaufen, da die *Bedingung* erst am Ende der Schleife überprüft wird.

Bei einer **While** - Schleife ist es notwendig, dass die Schleifenvariable *vor* der Schleife einen Wert zugewiesen bekommt. Bei einer **Repeat** – Schleife kann dies auch innerhalb des Befehlsblockes passieren, da die *Bedingung* ja erst am Ende überprüft wird. Aber genauso wie bei **While** sollte innerhalb des Befehlsblockes eine Anweisung stehen, die die Schleifenvariable verändert, so dass die Schleife beendet werden kann, damit die nachfolgenden Befehle ausgeführt werden können. Auch hier

```
NORMAL FLS AUTO REELL BOGENM MP
PROGRAM: WHILE
:0→K
:While K<10
:Disp K
:K+1→K
:End
:█
```



<p>steht diese Anweisung üblicherweise am Ende des Befehlsblockes.</p>	
<p><b>Ein Programm für die Fibonacci - Zahlen</b></p> <p>Es soll ein Programm geschrieben werden, dass die Fibonacci – Zahlen bis zu einer gewissen Grenze berechnet und ausgibt. Das Bild zeigt die ersten Zahlen bis <math>N = 5</math>.</p> <p>Kann man aus der Zahlenfolge das Bildungsgesetz erraten, ohne weiter unten auf der Seite die Lösung zu lesen?</p>	
<p>Mit dem Befehl <b>Prompt</b> wird ein Wert für die Obergrenze <b>N</b> durch den Benutzer eingegeben. Die ersten beiden Fibonacci – Zahlen sind 1 und 1 und werden in den Variablen <b>A</b> und <b>B</b> gespeichert. Diese Variablen werden dann auch für die weitere Berechnung verwendet.</p>	
<p>Nun schließt sich die <b>Repeat</b> – Schleife mit der Bedingung <b>A&gt;N</b> an. Der Befehlsblock soll also durchlaufen werden <i>bis A größer ist als N</i>. Die ersten beiden Zahlen werden ausgegeben.</p>	
<p>Jetzt werden die nächsten beiden Fibonacci – Zahlen berechnet und die Schleife wird mit <b>End</b> abgeschlossen.</p> <p>Es sieht dabei so aus, als ob in den beiden Variablen <b>A</b> und <b>B</b> mit <b>A+B</b> derselbe Wert abgespeichert würde. <i>Das ist aber nicht der Fall!</i> Man kann sich das ganz leicht für die ersten Zahlen überlegen. Der erste Befehl speichert <math>1+1</math> in <b>A</b>, also 2. Der zweite Befehl speichert nun <math>2+1</math> in <b>B</b>, also 3. Im nächsten Durchgang wird <math>2+3</math> in <b>A</b> abgespeichert, also 5, und <math>5+3</math> in <b>B</b>, also 8. Dann erhält man <math>5+8</math> für <b>A</b>, also 13, und <math>13+8</math> für <b>B</b>, also 21. Aber jetzt wird die <i>Bedingung</i> wahr und die Schleife endet.</p>	
<p>Man sollte nun das Programm mit verschiedenen Werten ausprobieren und sehen, ob es sich so verhält wie geplant.</p> <p>Was passiert, wenn man die <b>Repeat</b> – Bedingung in <b>B&gt;N</b> ändert?</p> <p>Wie kann man das Programm verändern, sodass auf jeden Fall immer die größte Fibonacci – Zahl angezeigt wird, die kleiner als <b>N</b> ist?</p>	

