

## Lektion 2: Prozeduren

Bearbeitet von Britta Schreiber & Kristel Jenkel

Es ist umständlich, für umfangreichere Grafiken lange Befehlslisten einzutippen. Wir müssten z. B. jedes Mal, wenn wir ein Rechteck der Länge 200 und der Breite 150 zeichnen wollten, die Befehlsfolge

```
fd 150 rt 90 fd 200 rt 90 fd 150 rt 90 fd 200
```

eintippen. Wir können eine solche Befehlsfolge auch unter einem Namen abspeichern: Befehlsfolgen, die mit einem Befehlswort aufgerufen werden, nennen wir **Prozeduren**. Den Namen für die Prozedur können wir selbst wählen. Wir werden solche Namen wählen, die uns daran erinnern, was das neu definierte Befehlswort bewirkt. Prozedurnamen dürfen nicht aus mehreren Wörtern bestehen. Wenn man dennoch mehrere Wörter zur Charakterisierung benötigt, muss man zwischen die Wörter ein verbindendes Zeichen setzen, z. B. einen Punkt oder einen Unterstrich, also z. B. RECHTECK.ZEICHNEN oder RECHTECK\_ZEICHNEN. Es spielt keine Rolle, ob wir große oder kleine Buchstaben verwenden.

Neue Prozeduren werden durch

```
TO { Prozedurname }
```

aufgerufen, also z. B. *to rechteck* weist den Rechner an, eine Befehlsfolge unter dem Namen RECHTECK anzunehmen. Es erscheint im oberen Bildschirmteil (im Grafikenfenster) ein neues Fenster, in das wir dann zeilenweise die Befehlsfolge eingeben können. Abgeschlossen wird die Eingabe mit dem Befehl *end*; hiermit wird automatisch die Eingabe der Prozedur beendet. Der Rechner meldet im Commander, dass er einen neuen Befehl gelernt hat (z. B. *rechteck defined*).

Das neue Befehlswort ist nun im Speicher, allerdings nur solange, wie er Rechner eingeschaltet ist! Immer wenn wir nun den Prozedurnamen eintippen, wird die zugehörige Befehlsfolge ausgeführt. Tippt man ein nicht-definiertes Befehlswort ein, z. B. *RECHT*, dann meldet der Rechner

```
I don't know how to RECHT
```

Beispiele:

```
to quadrat  
fd 200 rt 90 fd 200 rt 90 fd 200 rt 90 fd 200  
end
```

```
to dreieck  
fd 200 rt 120 fd 200 rt 120 fd 200  
end
```

Einen eingegebenen Prozedurtext kann man bearbeiten, indem man auf der oberen Bildschirmleiste unter „file“ auf „edit“ klickt.

Hier kann man die gewünschte Prozedur oder alle Prozeduren, die sich im Arbeitsspeicher befinden, zum Bearbeiten aufrufen.

Wer nicht gerne einen neuen Prozedurtext zeilenweise eingibt, kann direkt nach dem Prozedurnamen als zweite Eingabe *end* tippen, dann schließt sich der Zeileneditor. Dann kann man den Editor aufrufen und dort den Text der neuen Prozedur(en) eingeben. Den Editor verlässt man wieder, indem man unter „file“ den Befehl „save and exit“ anklickt.

## Problem 2

Schreibe eine Prozedur, mit deren Hilfe wir das Wort LOGO groß auf den Bildschirm schreiben können. Die Breite der Buchstaben soll 60, die Höhe 100 Turtleschritte betragen.

### Lösung

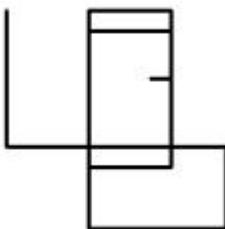
Das Wort LOGO setzt sich aus den drei Buchstaben L, O und G zusammen. Wir schreiben für jeden Buchstaben eine eigene Prozedur. Die Buchstaben müssen eckig geschrieben werden, da wir noch keine Kurven zeichnen können:

```
to L
fd 100 bk 100 rt 90 fd 60
end
```

```
to O
fd 100 rt 90 fd 60 rt 90 fd 100 rt 90 fd 60
end
```

```
to G
rt 90 fd 60 lt 90 fd 50 rt 90 bk 15 fd 15 rt 90 fd 50
rt 90 fd 60 rt 90 fd 100 rt 90 fd 60 rt 90 fd 15
end
```

Wenn wir allerdings jetzt nacheinander die Befehle *L O G O* (mit einem Blank zwischen den einzelnen Prozedurnamen) eingeben, dann sieht die Schrift gar nicht gut aus :



Wir müssen nach jedem Buchstaben wieder auf die Grundlinie gehen und dabei jedes Mal ein wenig Platz bis zum nächsten Buchstaben lassen. Außerdem sollten wir möglichst weit links auf dem Bildschirm anfangen. Dazu ergänzen wir die einzelnen Prozeduren um die notwendigen Befehle und schreiben eine Prozedur *nachlinks* und eine Prozedur *platz*:

```
to nachlinks
pu lt 90 fd 400 rt 90 pd
end
```

```
to platz
pu rt 90 fd 30 lt 90 pd
end
```

Es erscheint außerdem schöner, wenn die Schrift fließt, die Turtle also nicht hin und her springt. Wir verändern die Prozeduren L, O und G entsprechend:

```
to L
pu fd 100 pd
bk 100 rt 90 fd 60 lt 90
platz
end
```

```
to O
fd 100 rt 90 fd 60 rt 90 fd 100
rt 90 fd 60 bk 60 rt 90
platz
end
```

```
to G
pu fd 85 rt 90 fd 60 lt 90 pd
fd 15 lt 90 fd 60 lt 90 fd 100 lt 90 fd 60 lt 90 fd 50 lt 90 fd 15 bk 15 rt 90 bk 50
platz
end
```

Die gewünschte Prozedur setzt sich dann nur noch aus fünf Befehlen zusammen. Dadurch, dass wir die Aufgabe in kleine Teilaufgaben zerlegt haben, wird das Ganze leichter durchschaubar:

```
to LOGO
ht nachlinks L O G O
end
```



## Speichern von Grafiken

Grafiken kann man als Bitmap speichern; dies geschieht, indem man auf der oberen Bildschirmleiste „bitmap“ anklickt und die Grafik mithilfe von [save](#) oder [saveas](#) im gewünschten Ordner abspeichert.

Es empfiehlt sich, vor dem Speichern auszuprobieren, ob die Dicke der Turtle-Spur günstig ist. Dies kann dadurch geschehen, dass man in der oberen Bildschirmleiste die Option „set“ und „pensize“ anklickt und dann die gewünschte Strichstärke wählt.

## Probiere aus ...

- Was geschieht, wenn man bei der Eingabe der Befehlsfolge einer Prozedur den Befehl `end` vergisst? Überprüfe auch, was geschieht, wenn man mehrere Prozeduren auf einmal eingibt, ohne dass die einzelnen Befehlsfolgen durch `end` abgeschlossen werden.
- Welche Fehlermeldung erscheint, wenn man z. B. zwischen `fd` und `100` keine Leerstelle (blank) lässt? Warum erscheint diese Fehlermeldung, d. h. wie interpretiert der Rechner den Befehl?
- Was geschieht, wenn man als Prozedurnamen einen Namen einer bereits definierten Prozedur wählt, z.B. `cs` ?

## Überlege ...

Was geschieht, wenn man die Prozedur `RECHTECK` mehrfach hintereinander aufruft? Woran liegt es, dass nicht jedes Mal dasselbe Rechteck gezeichnet wird? Ergänze die Prozedur durch einen geeigneten Befehl.

## Übungen

### Ü2.1

Schreibe die Befehlsfolgen aus Problem 1, Ü1.2 und Ü1.3 als Prozeduren.

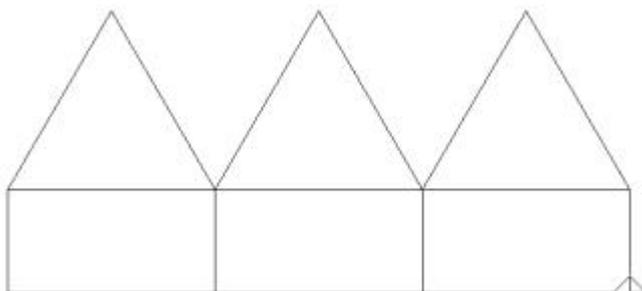
a) HAUS b) H c) T d) E e) TREPPE

### Ü2.2

Schreibe eine Prozedur `SPRUNG`, mit deren Hilfe man an den linken Rand springen kann (ohne dass die Turtle einen Strich dorthin zieht).

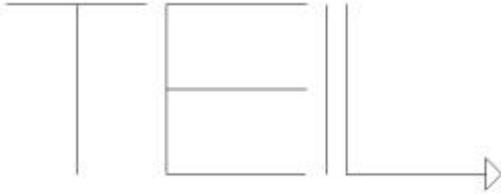
### Ü2.3

Schreibe eine Prozedur `HAUSREIHE`, durch die mehrere nebeneinander stehende Häuser gezeichnet werden sollen. Benutze dabei die Prozedur `HAUS` aus Ü1.2. Überlege, wo der Turtle am Ende der Prozedur `HAUS` stehen muss.



### Ü2.4

Schreibe eine Prozedur TEIL, die das Wort ‚TEIL‘ groß auf den Bildschirm schreibt. Benutze dabei Prozeduren für einzelne Buchstaben.



### Ü2.5

Schreibe eine Prozedur QUADER (vergleiche Ü1.4).

### Ü2.6

- a) Man kann den Quader in Ü2.6 mithilfe der Prozedur RECHTECK zeichnen. Dabei wird diese Prozedur zweimal in der Prozedur QUADER benötigt.
- b) Man kann den Würfel in Ü1.4 mithilfe der Prozeduren Raute und Quadrat zeichnen.