

Prozeduren – Die Schildkröte lernt dazu

Um die Lesbarkeit von Quelltexten zu verbessern, kann man eine Liste von Befehlen zu einem neuen Befehl zusammenfassen. Anstatt

```
wh 4 [vw 100 re 90]
```

schreibt man lieber:

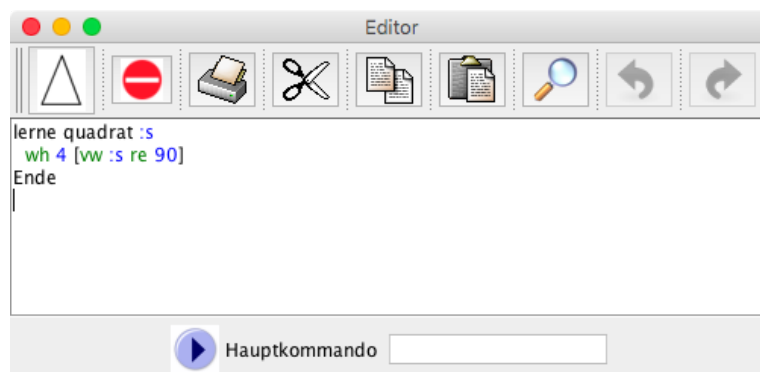
```
quadrat 100
```

Allerdings kennt die Schildkröte diesen Befehl nicht!







Aufgabe 1




Probier es einmal aus, den obigen `quadrat`-Befehl ausführen zu lassen.

Du musst diesen Befehl also erstmal definieren. Dazu muss du das Editor-Fenster öffnen, indem man auf die Schaltfläche "Editor" klickt, vgl. Abb.



Das Editor-Fenster funktioniert etwas eigenartig:

Schaltfläche	Erläuterung
	Speichere die im Editor gemachten Änderungen und schließe ihn dann. Es ist diese Schaltfläche, die Sie jedes Mal drücken müssen, um neu eingegebene Operationen anzuwenden. Wenn Sie wollen, können Sie das Tastaturabkürzel ALT + Q benutzen.
	Verlasse den Editor ohne eine der dort gemachten Änderungen zu übernehmen. Sie können auch die Abkürzung ALT + C benutzen.
	Drucke den Inhalt des Editors.
	Kopiere den ausgewählten Text in die Zwischenablage.
	Schneide den ausgewählten Text aus und kopiere ihn in die Zwischenablage.
	Füge den ausgewählten Text aus der Zwischenablage ein.

	Öffne ein Finde/Ersetzen-Dialogfeld für den Prozedur-Editor
	Mache die letzte Aktion rückgängig.
	Stelle die letzte Aktion wieder her.

Im Editor-Fenster kann man nun eigene Befehle – sogenannte Prozeduren – definieren. Prozeduren werden vom Wort `lerne` eingeführt und enden mit dem Wort `Ende`. Achten Sie darauf, dass hier `lerne` am Anfang klein und `Ende` am Ende groß geschrieben wird. Hier ist ein kleines Beispiel:

```
lerne quadrat :s
  wiederhole 4
  [
    vorwärts :s rechts 90
  ]
Ende
```

Die Prozedur wird `quadrat` genannt und erwartet einen `s` genannten Parameter als zusätzliche Information. `quadrat 100` wird deswegen ein Quadrat produzieren, das eine Seitenlänge von 100 hat. Der Parameterwert `s` gibt also genau an, wie der Befehl auszuführen ist. Ohne diese zusätzliche Information wüsste die Schildkröte nicht, um wie viele Pixel sie nach vorne gehen oder um wieviel Grad sie sich drehen soll.

Allgemein hat eine Prozedur folgende Form:

```
lerne nameVonProzedur :v1 :v2 :v3.... [:v4 ....] [:v5 ....]
  Körper der Prozedur mit Abfolge von Anweisungen
Ende
```

Dabei ist `nameVonProzedur` ein frei wählbarer Name, der der Prozedur gegeben wird.

`:v1 :v2 :v3` stehen für die Parameter (lokale Übergabe-Variablen), die innerhalb der Prozedur benutzt werden.

`[:v4 ...], [:v5 ...]` sind optionale Variablen, die man der Prozedur hinzufügen kann.

Das Konzept der optionalen Variablen zeigt folgendes Beispiel:

```
lerne quadrat [:s 100]
  wiederhole 4
  [
    vorwärts :s rechts 90
  ]
Ende
```

Hier wird also wieder eine Prozedur `quadrat` definiert, wobei diesmal die Parameterangabe beim Prozeduraufruf optional ist:

```
quadrat
```

```
--> Die Schildkröte zeichnet ein Quadrat mit der Standardbelegung  
von :s = 100
```

```
(quadrat 50)
```

```
--> Die Schildkröte zeichnet ein Quadrat mit dem Parameterwert von  
:s = 50
```

Beachte: Wenn man den Parameterwert ändern will, muss man die Prozedur `quadrat` zwischen Klammern aufrufen, um dem Interpreter zu sagen, dass man optionale Argumente benutzen möchte.

Aufgabe 2

Bringe den Befehl wie oben beschrieben der Schildkröte bei. Probier es nun erneut aus, den obigen `quadrat`-Befehl ausführen zu lassen. Nutze auch mal die Version mit optionalen Parametern.

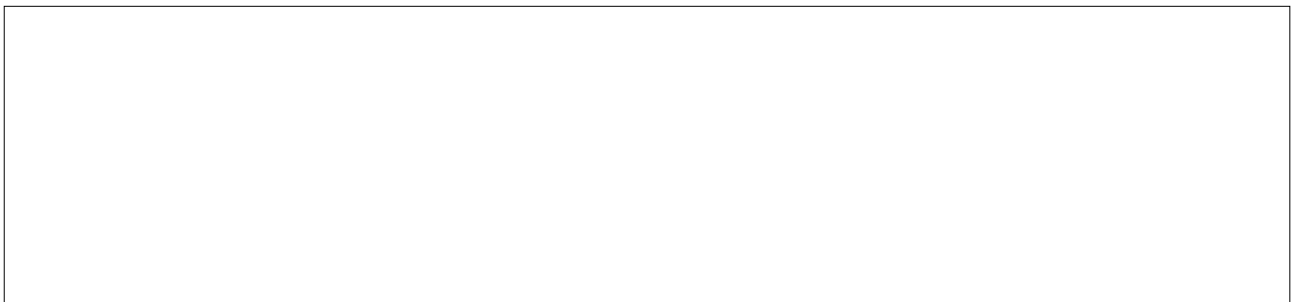
Aufgabe 3

Schreibe eigene Prozeduren für folgende Grundfiguren: Rechteck, n-Eck. Nutze jeweils passende Parameterlisten.

Lösung "Rechteck":



Lösung "n-Eck":



Aufgabe 4

Schreibe eine Prozedur für einen Stern. Dabei sollen die Parameter für die Anzahl der Zacken und die Seitenlänge der Zacken feste vorgegeben werden. Zusätzlich soll über einen optionalen Parameter die Spitze der Zacken gesteuert werden können.

Lösung "Stern":

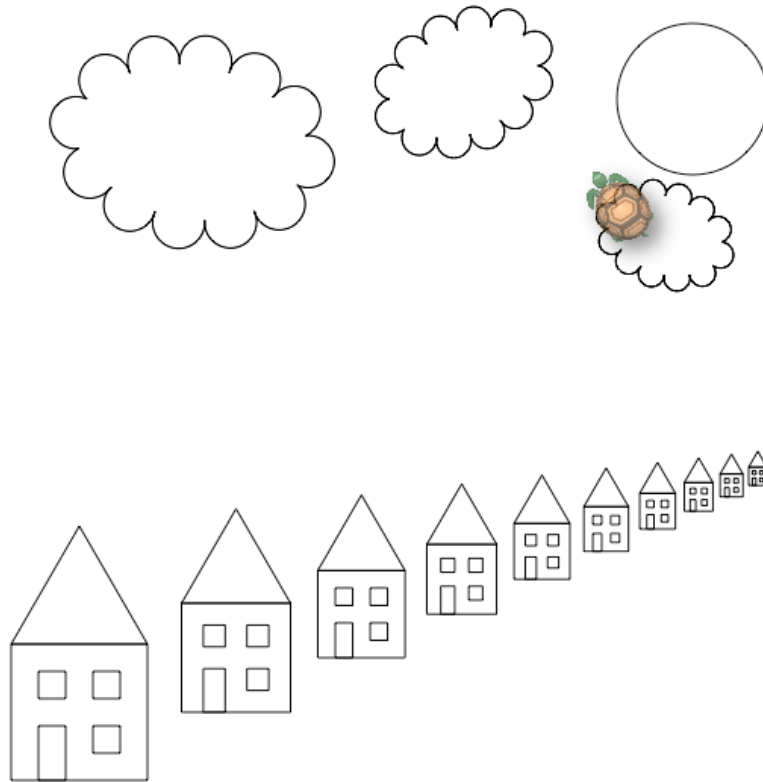
Aufgabe 5

Nutze die neuen Prozeduren, um eine Prozedur "Haus" zu implementieren. Das Haus soll dabei in der Größe veränderbar sein.

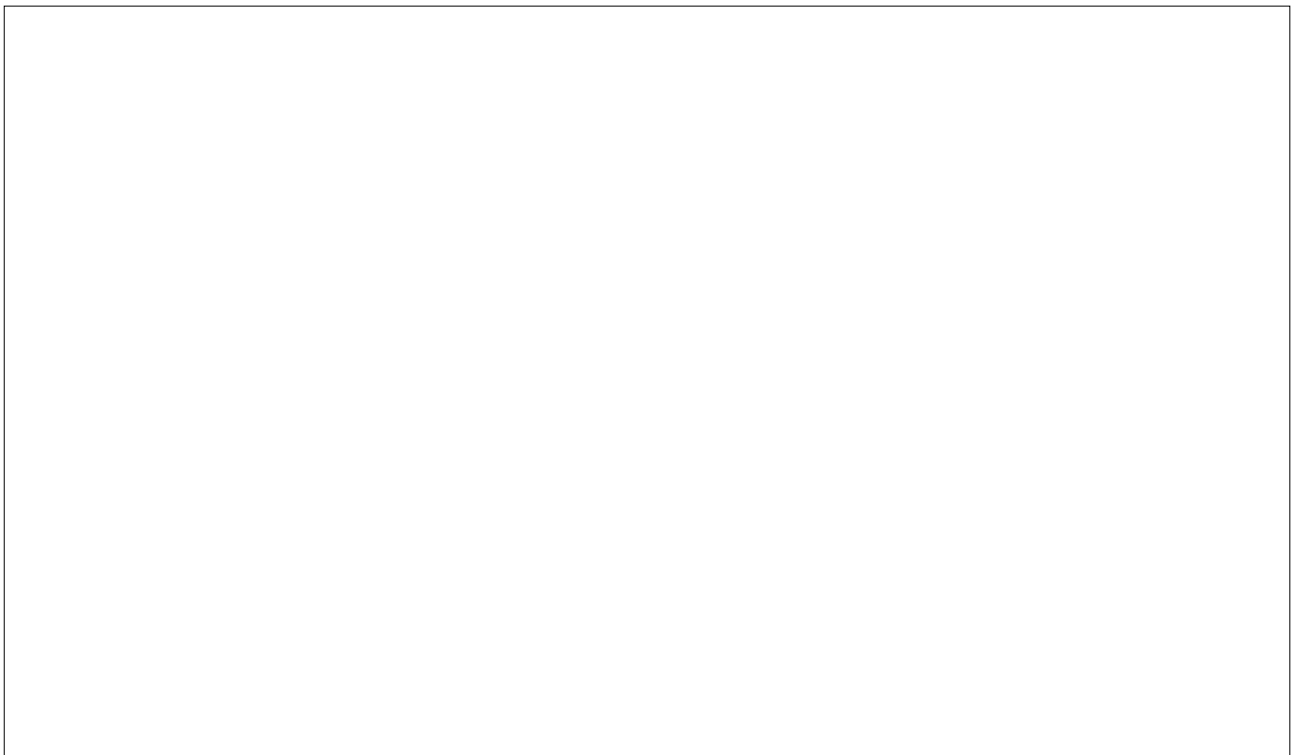
Lösung "Haus":

Aufgabe 6

Erzeuge eine Häuserreihe, die perspektivisch kleiner wird. Gerne kannst du auch eine ganze Landschaft mit Bäumen und Wolken und einer Sonne usw. draus machen. Denke aber daran, für alle Teilfiguren eigene Prozeduren zu erzeugen.



Lösung "Landschaft":



Prozeduren – Anweisungen und Anfragen

Man kann der Schildkröte nicht nur beibringen, etwas zu tun. Man kann ihr auch beibringen, auf etwas zu antworten. Allgemein unterscheidet man:

- Anweisungen: Befehle, die dazu führen, dass die Schildkröte etwas tut.
- Anfragen: Befehle, die dazu führen, dass die Schildkröte etwas antwortet.

Bisher hast du nur Anweisungen programmiert. Nun wollen wir der Schildkröte beibringen, auch Anfragen zu beantworten:

```
lerne berechneRechteckflaeche :a :b
      rückgabe :a * :b
Ende
```

Entscheidend ist hier der Befehl `rückgabe` in der letzten Zeile:

Befehl	Parameter	Beschreibung
rg, rückgabe	a : Wert	Bricht aus einer Prozedur aus mit einem zurückzugebenden Wert a.

Aufgabe 7

Schreibe eine Prozedur `berechneRechteckUmfang`, die den Umfang eines Rechtecks zurückgibt.

Lösung "berechneRechteckUmfang":

Manchmal möchte man auch, dass die Schildkröte einen Befehl wieder vergisst. Hier helfen folgende Befehle:

Befehl	Parameter	Beschreibung
vg, vergesse	a. Wort	Entfernt die Prozedur namens a.
vgv, vergessevar	a: Wort	Löscht die Variable a.
vga, vergessealles	keine	Entfernt alle aktuellen Variablen und aktuellen Prozeduren.