

1. Erste Programme

Lernziele:

- Du lernst, dass der Rechner eindeutige Anweisungen in einer eigenen Sprache erwartet und ausführt.
- Du erstellst deine ersten Programme und lernst, was ein Programm, Befehle und Parameter sind.

In der Einstiegsübung hast du festgestellt, wie wichtig **eindeutige** Anweisungen sind.

Begriffe

Programmieren heisst, Texte in einer Programmiersprache zu schreiben, welche aus klar definierten Wörtern und einer eindeutigen Syntax besteht. Die Wörter der Programmiersprache nennen wir **Befehle**. Sie beschreiben einfache Aktionen.

Wir verwenden die Programmiersprache **TigerJython**, ein Dialekt der modernen Sprache **Python**.

1.1 Steuerung der Turtle

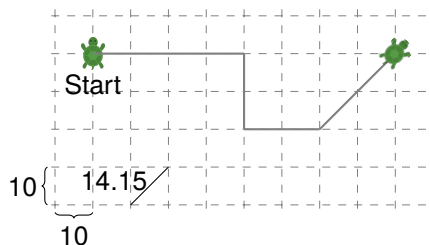
Wir beginnen mit Grafiken, die du mit der Steuerung einer „Turtle“ mit folgenden Befehlen erstellst:

Befehle	Die Turtle...	Abkürzungen
<code>forward(s)</code>	.. macht s Schritte vorwärts.	<code>fd(s)</code>
<code>back(s)</code>	.. macht s Schritte rückwärts.	<code>bk(w)</code>
<code>right(w)</code>	.. dreht sich um w Grad im Uhrzeigersinn	<code>rt(w)</code>
<code>left(w)</code>	.. dreht sich um w Grad im Gegenuhrzeigersinn	<code>lt(w)</code>

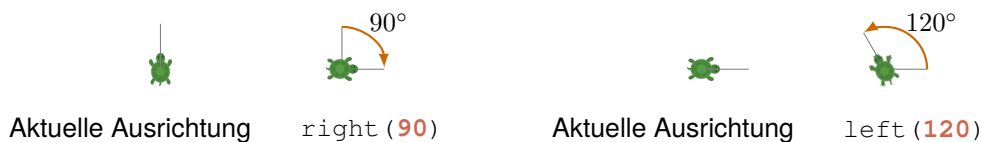
Beispiel 1.1

Ein Programm wird bei der Ausführung Zeile für Zeile ausgeführt:

```
right(90)
forward(40)
right(90)
forward(20)
left(90)
forward(20)
left(45)
forward(28.3)
```



Beachte: Die Befehle beziehen sich immer auf die aktuelle Position/Ausrichtung der Turtle:



Begriffe

Befehle bestehen aus einem **Befehlswort**, das eine Tätigkeit beschreibt und runden Klammern, in denen wir dem Befehl Werte (**Parameter**) übergeben können.

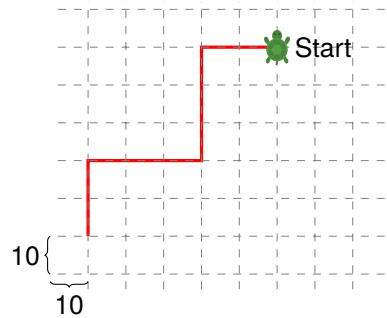
Der Befehl `forward` bewegt die Turtle vorwärts. Die Anzahl Schritte, welche die Turtle dabei machen soll, übergeben wir dem Befehl in runden Klammern (z.B. `50`).

```
forward(50)
  ↑      ↑
  Befehlswort Parameter
```

Du wirst Befehle kennenlernen, denen keine Parameter übergeben werden müssen (z.B. `ht()`).

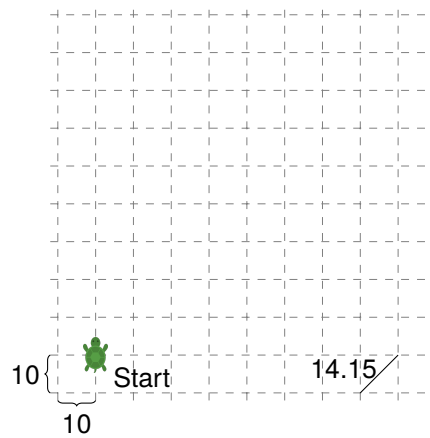
Aufgabe 1.1

- a) Schreibe links die Befehle auf, mit denen die Turtle die eingezeichnete Strecke aus der Startposition zurücklegt.



- b) Zeichne die Strecke ein, welche die Turtle bei der Ausführung des Programms aus der Startposition zurücklegt.

```
forward(60)
right(45)
forward(28.3)
right(45)
forward(50)
right(90)
forward(40)
left(45)
back(28.3)
right(90)
forward(28.3)
left(45)
forward(40)
left(90)
back(30)
```



1.2 Programme mit dem Computer

Um eigene Programme zu schreiben, verwenden wir den lokalen Editor für TigerJython, den du kostenlos von www.tjgroup.ch herunterladen kannst.

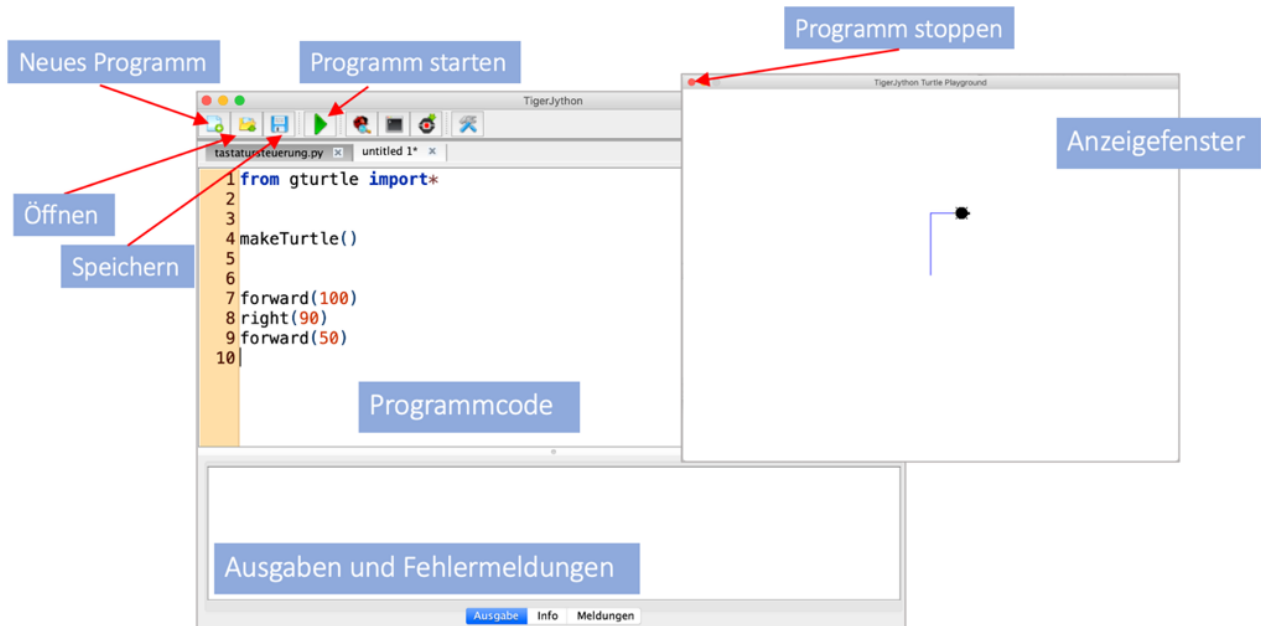


Abb. 1.1: Die Oberfläche des TigerJython Editors.

Beispiel 1.2

Öffne den **Editor** durch Klick auf den 🐻-Button und gib das folgende Programm ein:

1. Turtle Bibliothek
2. Turtle erstellen
3. Steuerbefehle

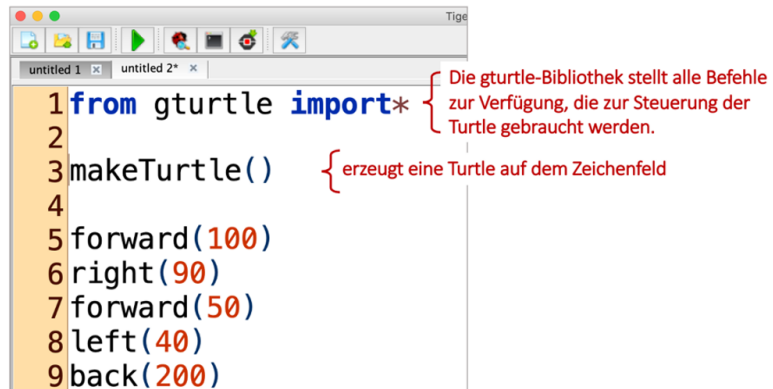


Abb. 1.2: Aufbau eines Turtleprogramms.

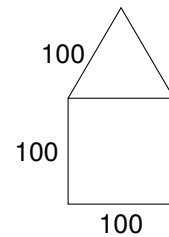
Beachte:

- Die Zeilen 1 und 3 müssen in jedem Programm vorhanden sein!
- Die Leerzeilen haben keine Auswirkung auf den Programmverlauf. Sie machen den Code aber etwas übersichtlicher.

Aufgabe 1.2

Schreibe ein Programm, mit dem die Turtle das abgebildete Haus zeichnet. Alle Strecken sind gleich lang.

Wie muss das Programm aussehen, wenn die Turtle keine der Strecken zweimal zeichnen/abfahren darf?

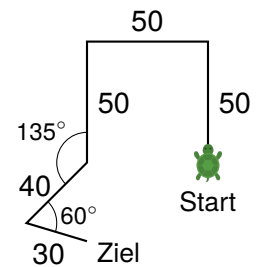


Aufgabe 1.3

Schreibe ein Programm, mit dem die Turtle vom Start ins Ziel gelangt.

Einschränkung: Verwende im Programm ausschliesslich die Befehle

`back()` und `right()`



Tipp:

Mit einem Hashtag `#` kannst du **Zeilen deaktivieren**. Diese werden beim Programmablauf nicht ausgeführt. Damit lässt sich Code testen, ohne Zeilen löschen zu müssen (Zeile 4).

```
1 from turtle import*
2 makeTurtle()
3
4 #forward(100)
5 forward(50)
6 #Kommentar
```

In der Praxis verwenden Programmierer und Programmiererinnen dies sehr oft, um **Kommentare** zu Codeabschnitten einzufügen. (Zeile 6).

Markiere eine/mehrere Zeilen und re-/ und deaktiviere sie mit der folgenden Tastenkombination:

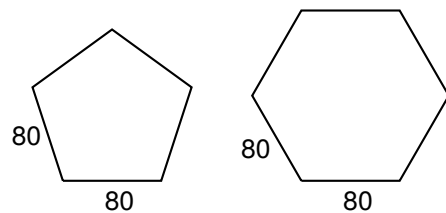
Windows: **Ctrl + Q**

Apple: **Control + Q**

Aufgabe 1.4

Ein Vieleck ist regelmässig, wenn alle Seiten gleich lang und alle Winkel gleich gross sind.

Schreibe je ein Programm, das ein regelmässiges Fünfeck und Sechseck in der dargestellten Lage zeichnet.



Aufgabe 1.5

Schreibe ein Programm, das den Stern in der dargestellten Lage zeichnet:

Tipp: Den Startpunkt der Turtle kannst du beliebig wählen.

