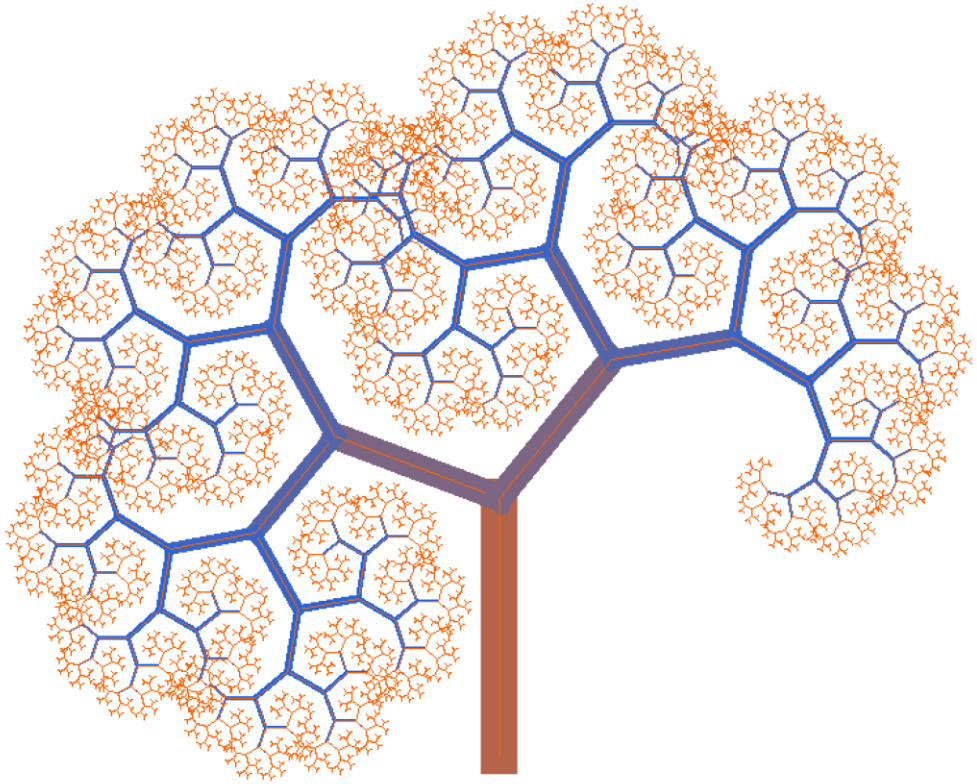
Programmieren mit TigerJython



ABZ
AUSBILDUNGS- UND BERATUNGSZENTRUM
FÜR INFORMATIKUNTERRICHT

ETH zürich

1. Erste Programme

Lernziele:

- Du lernst, dass der Computer eindeutige Anweisungen in einer eigenen Sprache erwartet und ausführt.
- Du schreibst deine erste eigenen Programme und lernst die Begriffe Programm, Befehl und Parameter kennen.

Einstiegsübung 1

Jemand aus der 2-er Gruppe baut ein LEGO-Objekt aufgrund der Beschreibung des Partners/der Partnerin möglichst genau nach.

Anschliessend werden die Rollen getauscht.

Abb. 1: LEGO-Objekt zum Nachbauen.

Im Detail:

1. Setzt euch mit dem Rücken zueinander an einen Tisch.
2. A beschreibt den Aufbau des LEGO-Objekts schrittweise.
3. B baut das Objekt nach. Wichtig: B darf keine Rückfragen stellen.
4. Sobald B das Objekt gebaut hat, tauscht ihr die Rollen:
   B beschreibt ein neues LEGO-Objekt und A baut es nach.
5. Sobald A fertig ist, vergleicht ihr die Objekte und notiert eure Eindrücke:
   (a) Was hat gut/schlecht funktioniert?
   (b) Was ist eurer Meinung nach wichtig bei der Beschreibung des Objekts?

Einstiegsübung 1

Notiert eure Erkenntnisse auf dem Blatt.

Einstiegsübung 1

Eure Erkenntnisse?

Abb. 1: LEGO-Objekt zum Nachbauen.

Einstiegsübung 1

Fazit:

- Eindeutige und klare Anweisungen (Sprache).
- Fokus auf das Wesentliche/unwichtige Details weglassen. (abstrahieren)
- Keine Interpretation der Anweisungen.

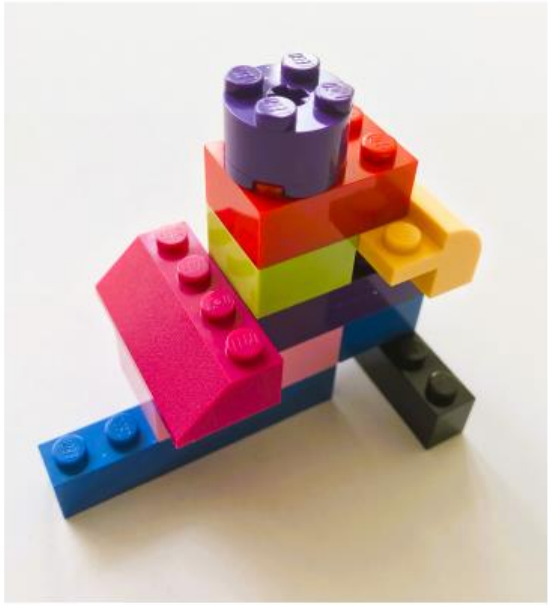


Abb. 1: LEGO-Objekt zum Nachbauen.

Einstiegsübung 1

Arbeitsweise eines Computers:

- Ein Computer interpretiert Anweisungen nicht, sondern führt sie exakt so aus wie vorgegeben.

- Ein Computer erkennt Syntaxfehler in einem Programm (z.B. Fehler in der Schreibweise/Anordnung)

  → Unter Syntax verstehen wir *Regeln* nach denen Texte strukturiert werden.

- Er erkennt keine Semantikfehler. (z.B. welche konkrete Aufgabe ein Programm lösen soll)

  → Unter Semantik verstehen wir die Zuordnung von *Bedeutung* zu Texten.

Neue Begriffe

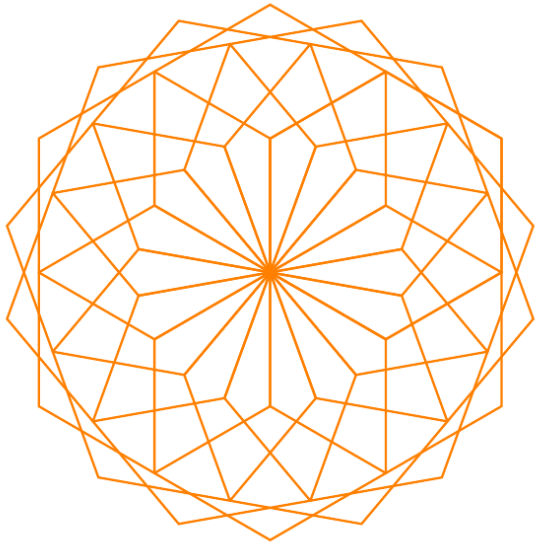
Programmieren heisst, Texte in einer Programmier-sprache zu schreiben, welche aus klar definierten Wörtern (wir nennen sie nennen Befehle) und einer eindeutigen Syntax besteht.

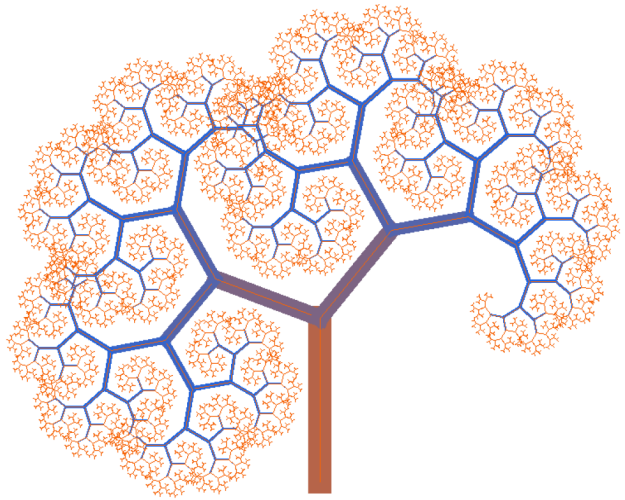
Steuerung der Turtle

Um Programmierkonzepte zu lernen, erstellst du Grafiken/Animationen mit Hilfe einer „Turtle“.

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

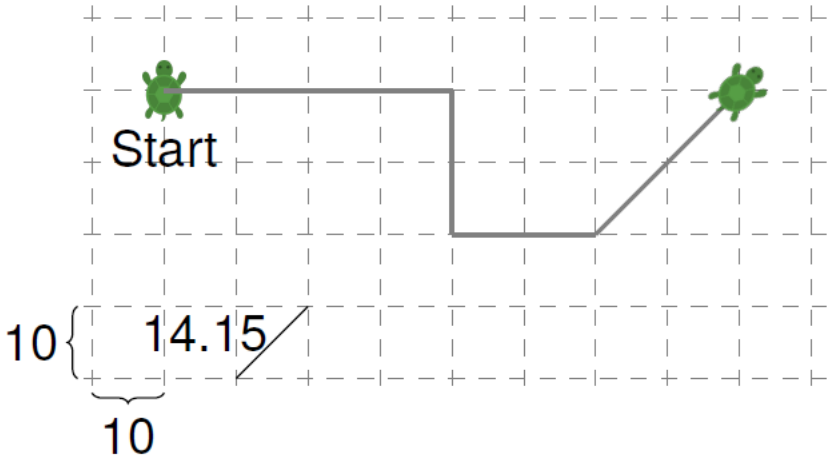
Beispielgrafiken:





Beispiel 1.1

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



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

Aktuelle Ausrichtung `right(90)`

Aktuelle Ausrichtung `left(120)`

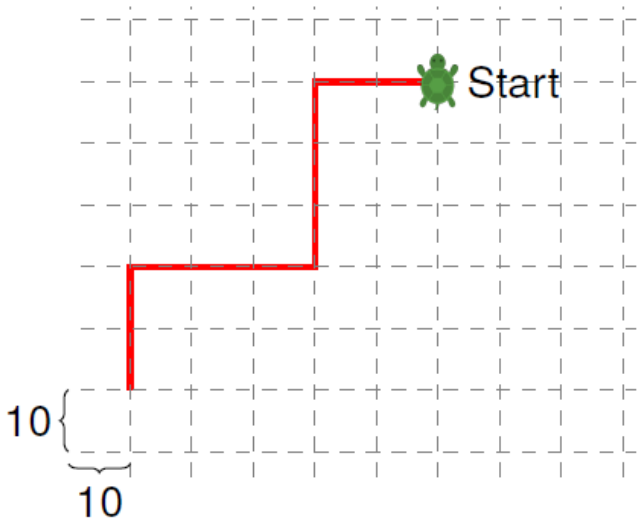
Übungsphase

Löse die Aufgabe 1.1 auf Papier.

Aufgabe 1.1 a)

Lösung:

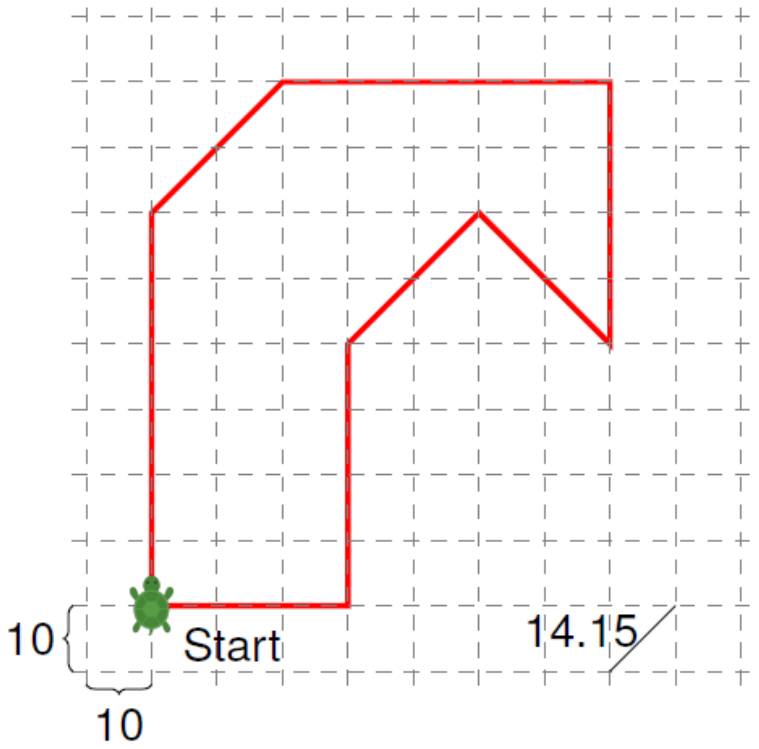
```
left(90)
forward(20)
left(90)
forward(30)
right(90)
forward(30)
left(90)
forward(20)
```



Aufgabe 1.1 b)

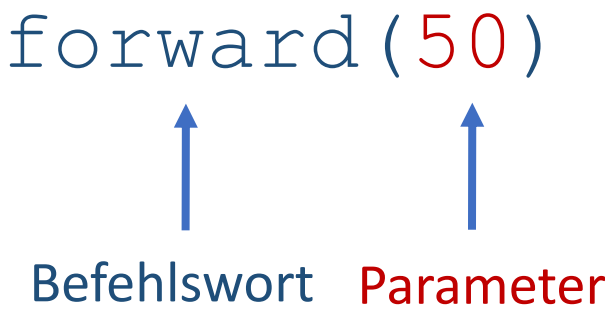
Lösung:

```
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)
```



Neue Begriffe

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



Wichtig:
Ein Befehl besitzt *immer* runde Klammern, auch wenn keine Parameterwerte übergeben werden!
(z.B. `makeTurtle()`)

Unsere Programmiersprache

Wir verwenden die Programmiersprache Tigerjython, ein Dialekt der Sprache Python.

Python …

- … wurde von *Guido van Rossum* in den Niederlanden um 1991 entwickelt.
- … ist sehr einfach, mächtig und verbreitet. (Ziel: soll so einfach zu lesen sein wie Englisch)
- … Einrückungen (anstatt Klammern) sind ihr Hauptmerkmal.
- … ist nach der britischen Komikergruppe „Monty Python“ benannt.

Quelle: https://allfamous.org/


```
eingabe = 1

while eingabe != 0:
  print("Geben Sie einen Inch-Wert ein: ")
  inchwert = input()
  eingabe = int(inchwert)

  if eingabe != 0:
    print(eingabe, " inch, sind: ", eingabe * 2.54, "cm")
```

Unsere Programmiersprache

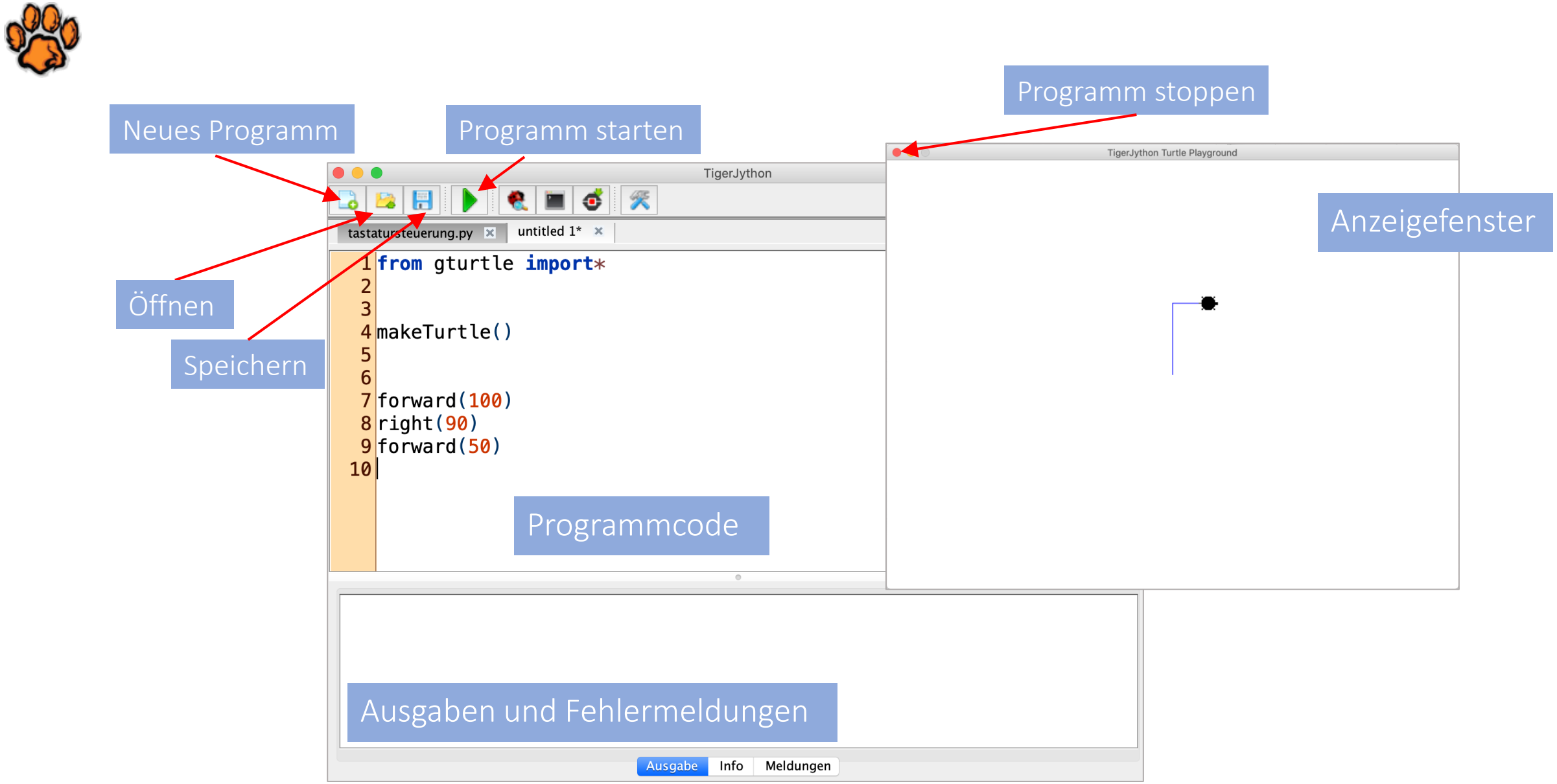
Wir verwenden die Programmiersprache Tigerjython, ein Dialekt der Sprache Python.

TigerJython …

- … wurde von *Tobias Kohn* in der ABZ-Gruppe der ETH Zürich um 2016 entwickelt.
- … simuliert die Sprache Python, basiert im Hintergrund aber auf Jython (Java).
- … besitzt Vereinfachungen für Programmieranfänger die es so in Python nicht gibt (z.B. repeat)
- … besitzt eine sehr präzisesten Fehlermeldungsengines

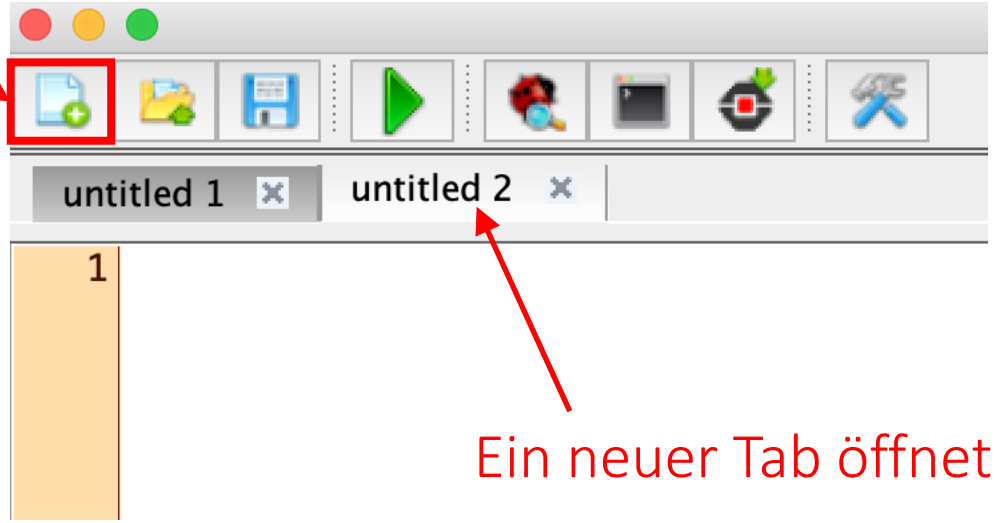
```
1  from gturtle import *
2  makeTurtle()
3
4  def quadrat(s, x, y, w):
5      setX(x)
6      setY(y)
7      setHeading(w)
8      repeat 4:
9          forward(s)
10         right(90)
11
12 #hideTurtle()
13 w=0
14 repeat 36:
15     quadrat(50, 200, 160, w)
16     w=w+10
17
```

TigerJython Editor



Neues Programm erstellen

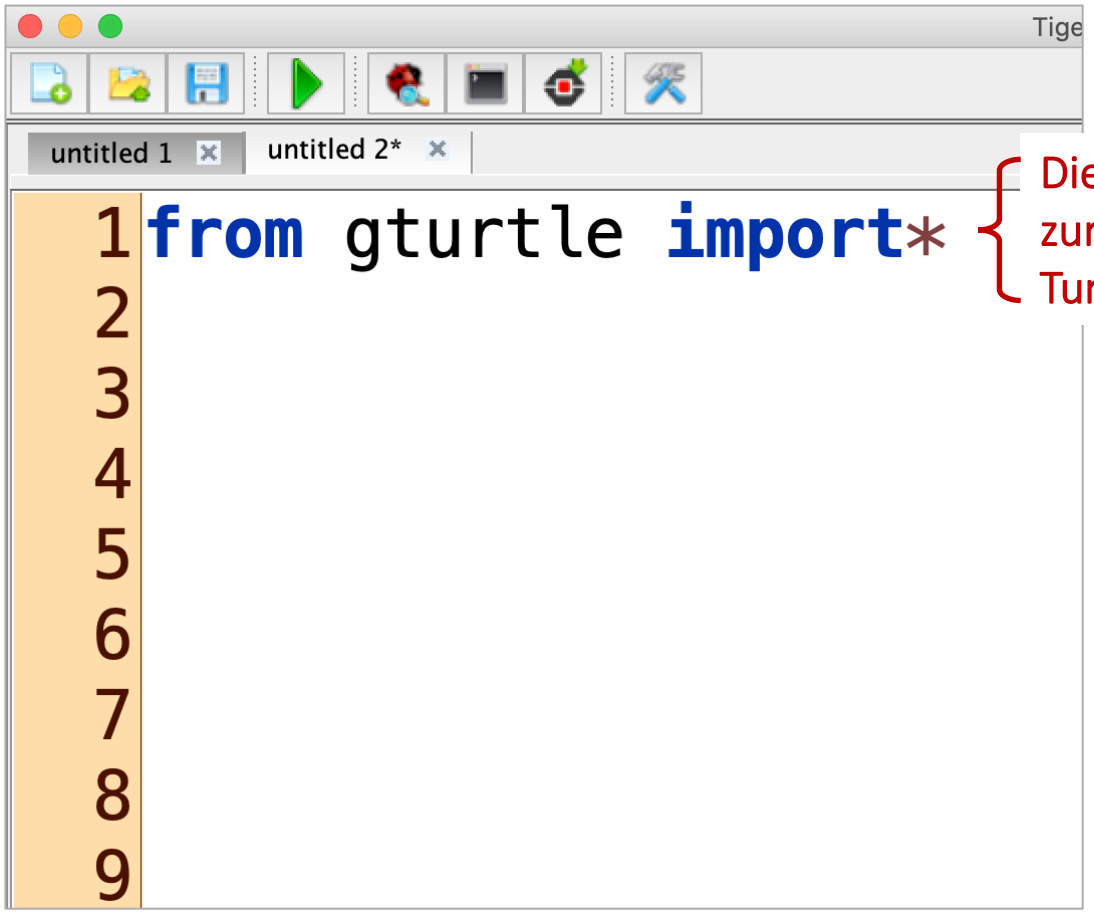
Neues Programm



Ein neuer Tab öffnet sich

Ein erstes Turtle-Programm:

1. Turtle Bibliothek

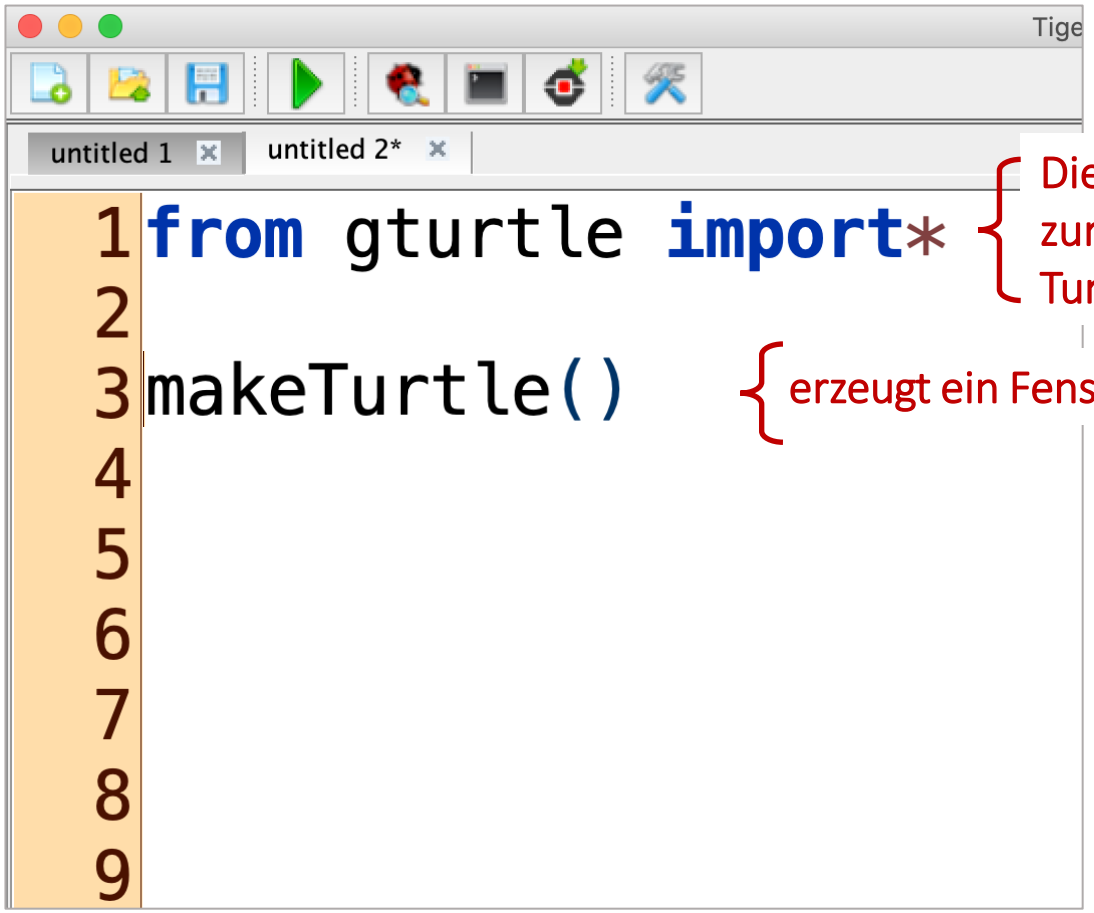


Die gturtle-Bibliothek stellt alle Befehle zur Verfügung, die zur Steuerung der Turtle gebraucht werden.

Programmaufbau

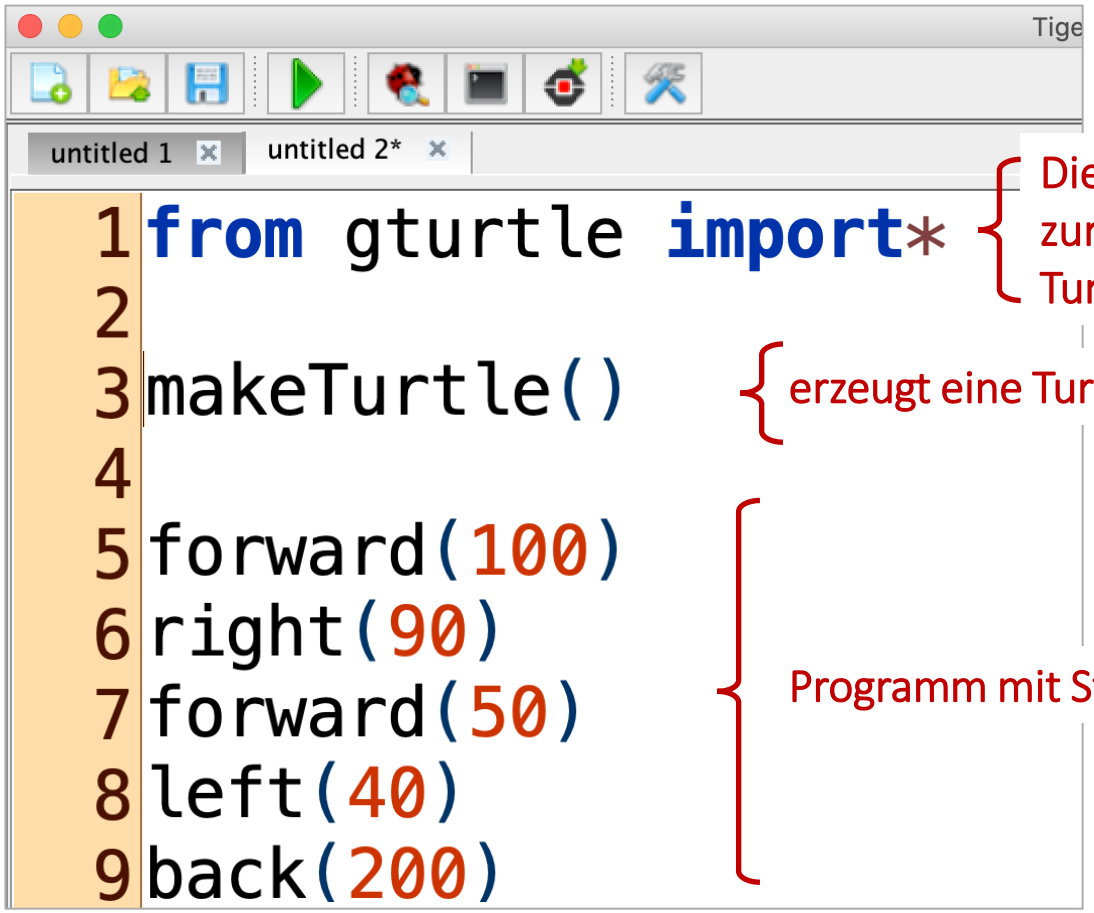
1. Turtle Bibliothek

2. Fenster/Turtle erstellen

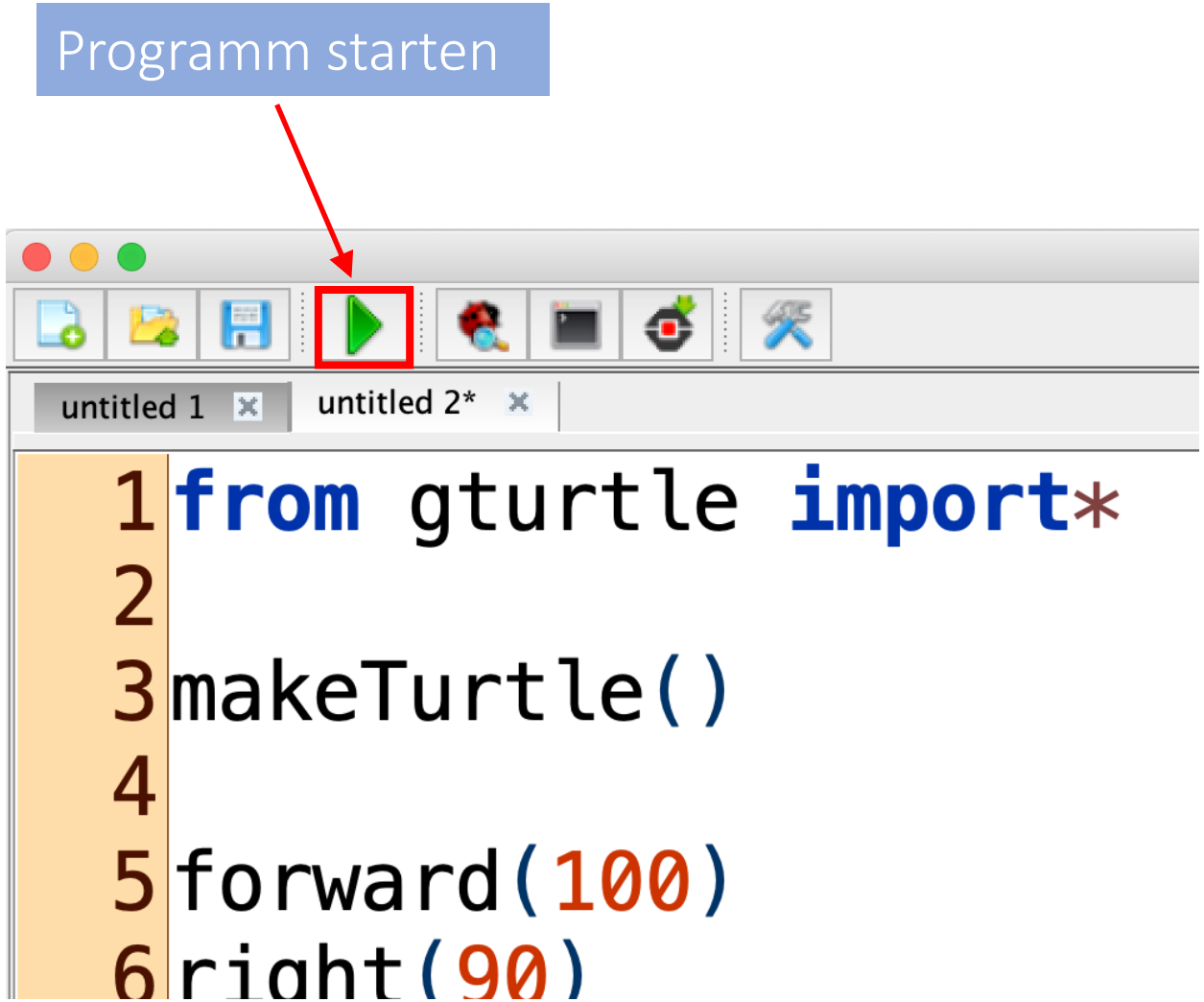


Programmaufbau

1. Turtle Bibliothek
2. Fenster/Turtle erstellen
3. Steuerbefehle

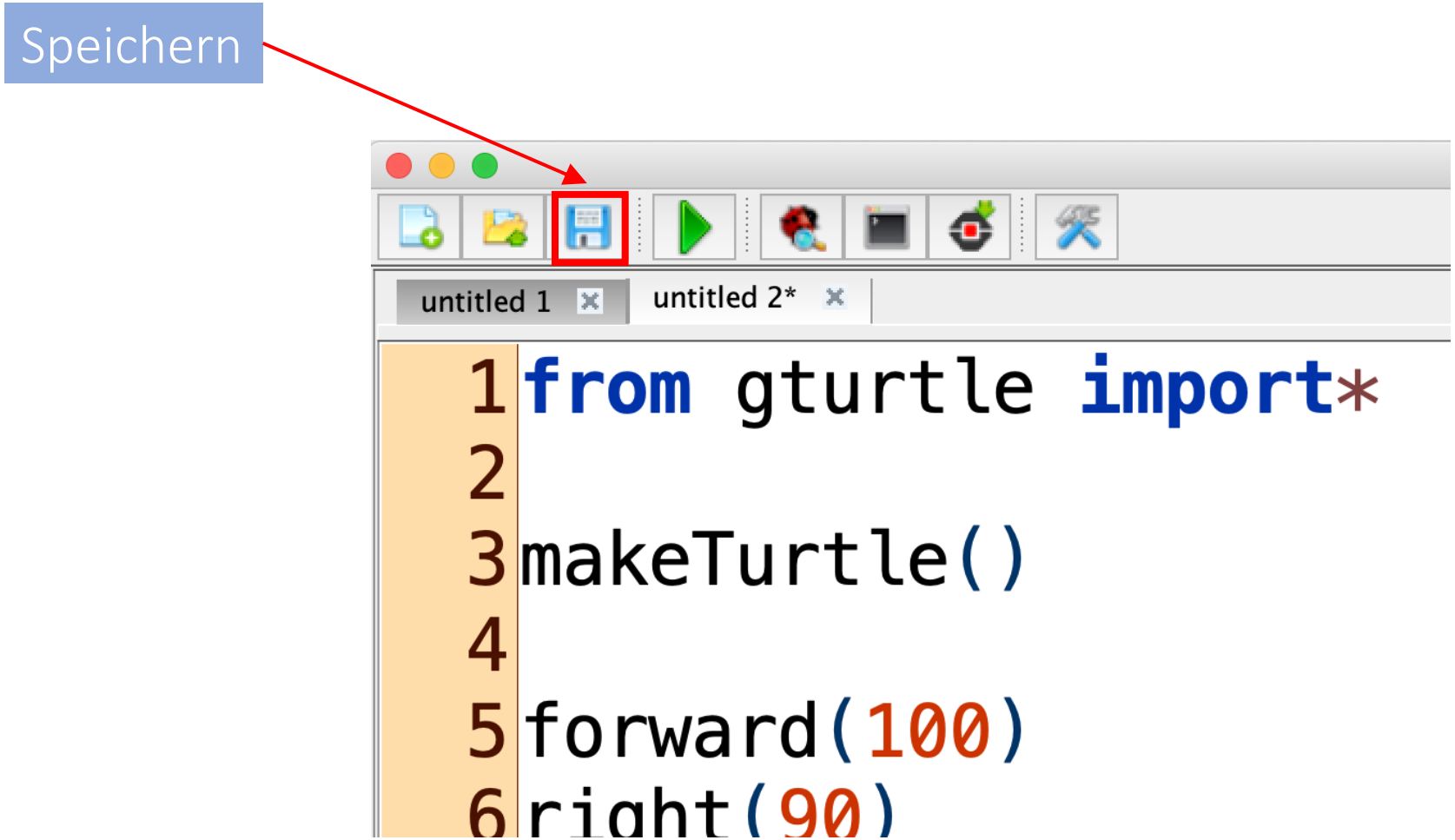


Programm starten



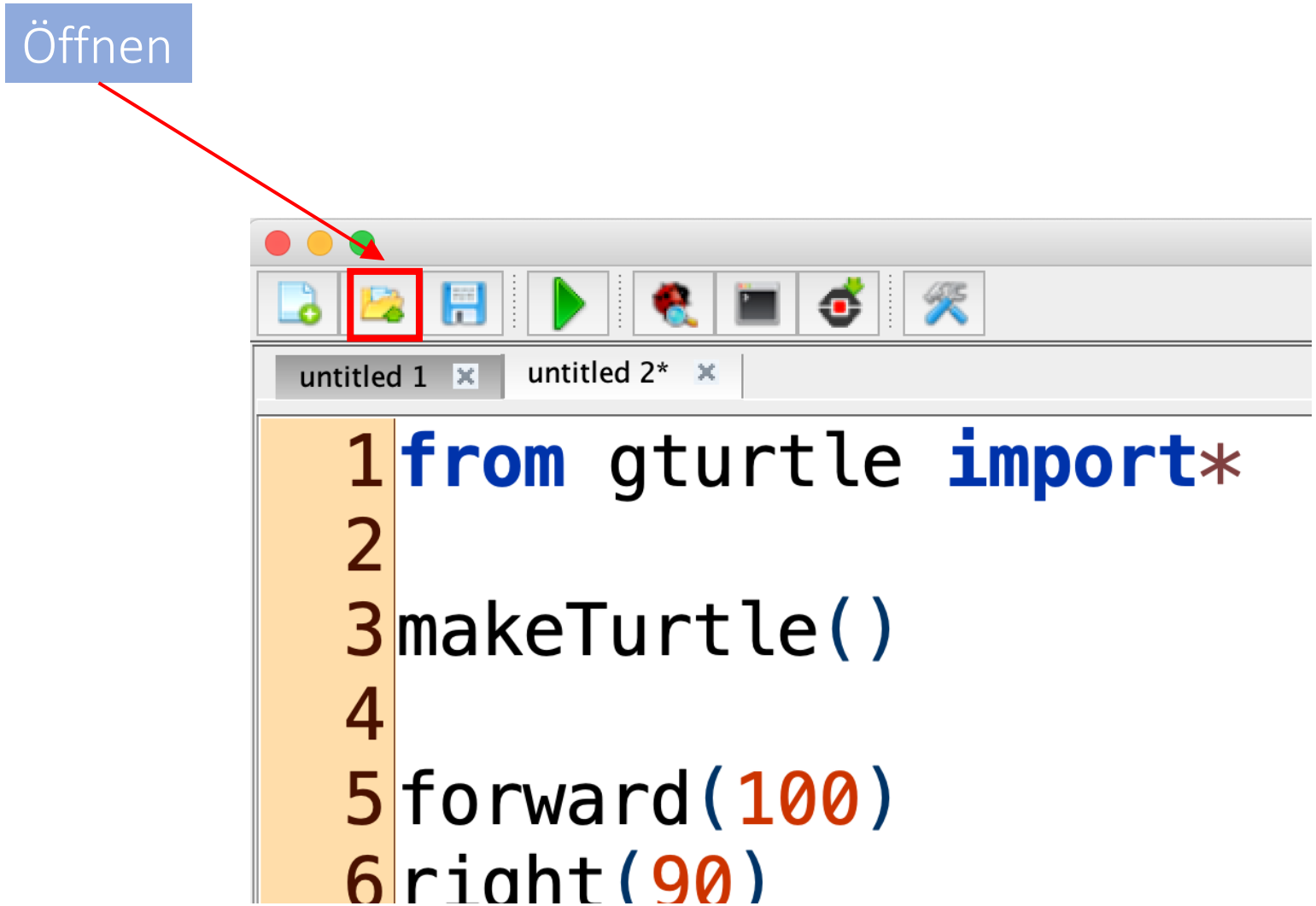
Programm speichern

Speichere die umfangreicheren Programme jeweils ab. Du wirst sie immer wieder brauchen und erweitern.



Programm öffnen

Ein *vorhandenes Programm* muss immer vom Editor aus geöffnet werden!



Übungsphase

Löse die Aufgaben 1.2 bis 1.5 am Computer

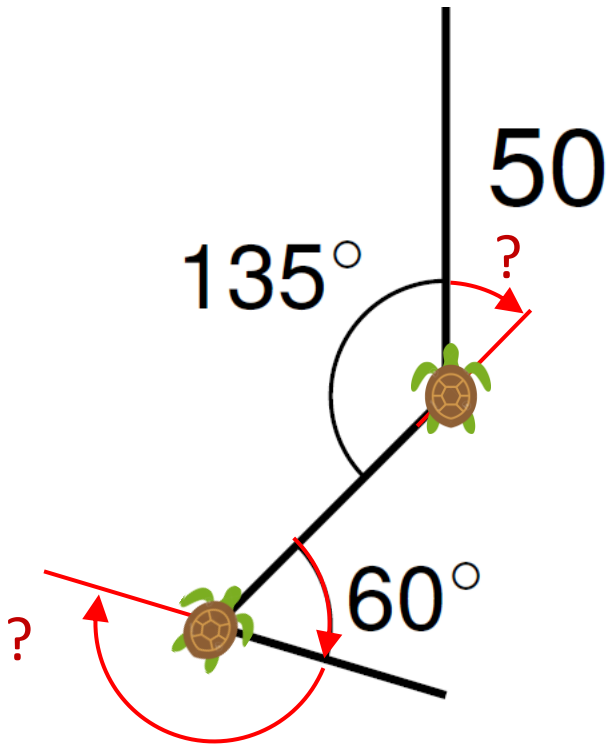
Hinweis : Übungsblätter und Folien findet ihr auch auf der Webseite https://webskin.ch/

Aufgabe 1.3

Lösung:

Einschränkung bei dieser Aufgabe:
Turtle kann nur **rückwärts fahren** und
nach **rechts drehen**

Wie gross müssen die Drehwinkel sein?

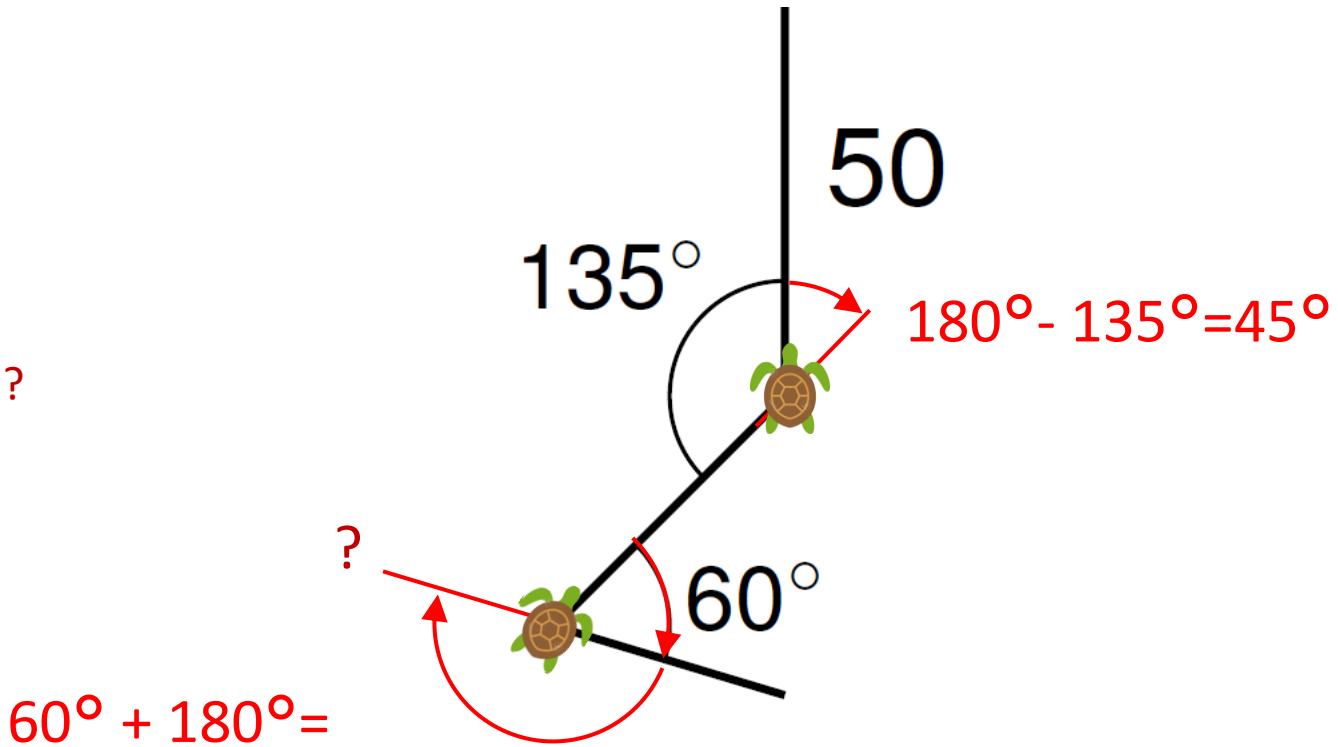


Aufgabe 1.3

Lösung:

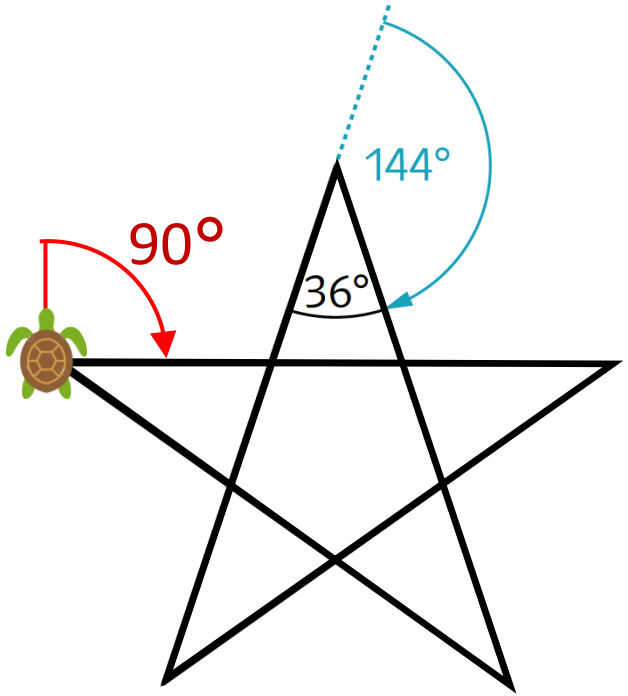
Einschränkung bei dieser Aufgabe:
Turtle kann nur **rückwärts fahren** und
nach **rechts drehen**

Wie gross müssen die Drehwinkel sein?



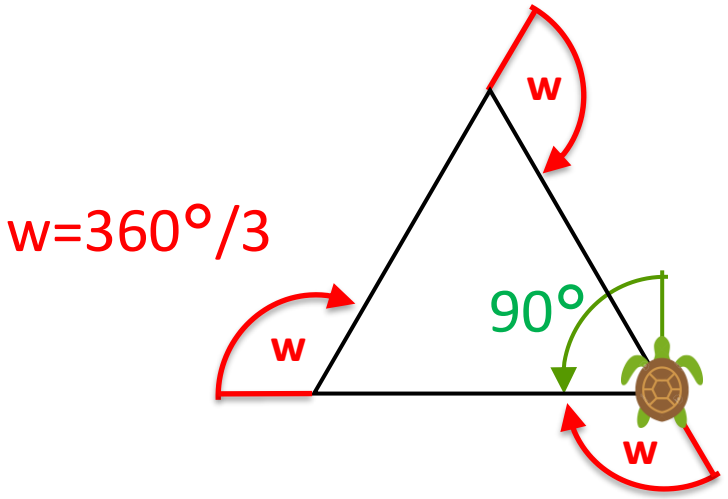
Aufgabe 1.4

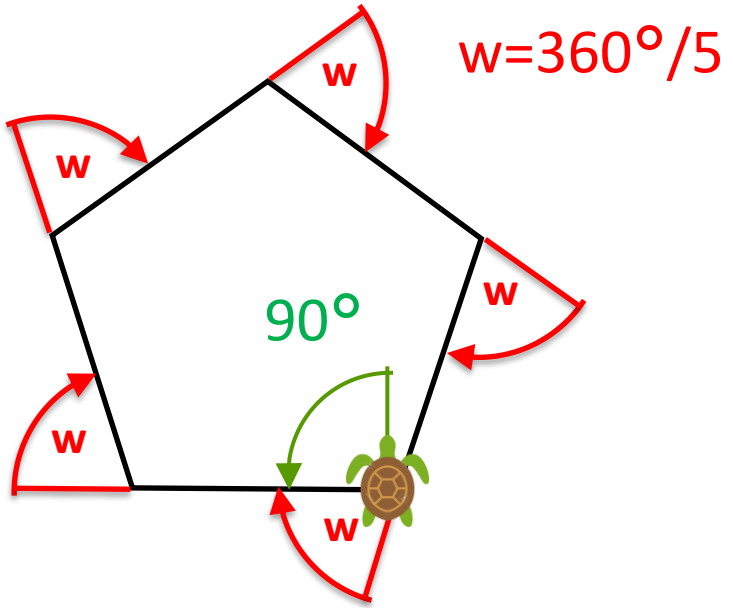
Es gibt mehrere korrekte Lösungen(Programme).
Aber ein clever gewählter Startpunkt der Turtle kann die Lösungsfindung vereinfachen.



Aufgabe 1.5

Drehwinkel: Die Turtle macht insgesamt immer eine ganze Umdrehung (360°)!



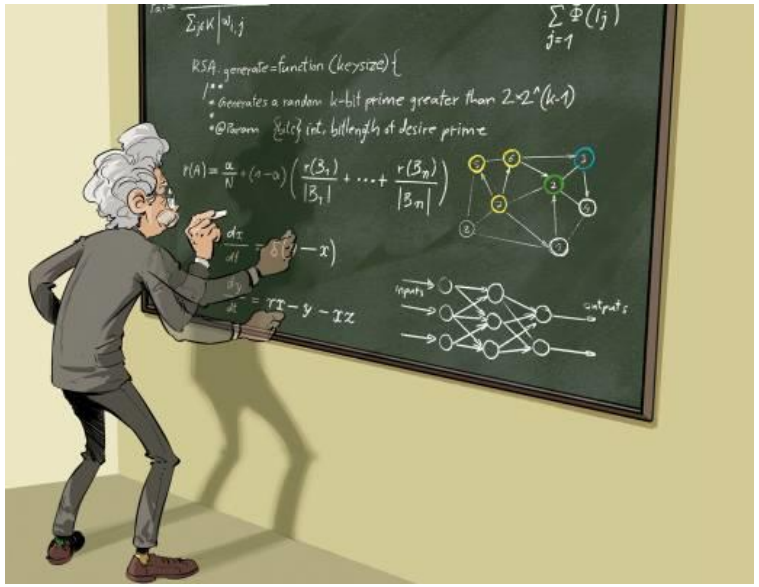


Take-Home Message

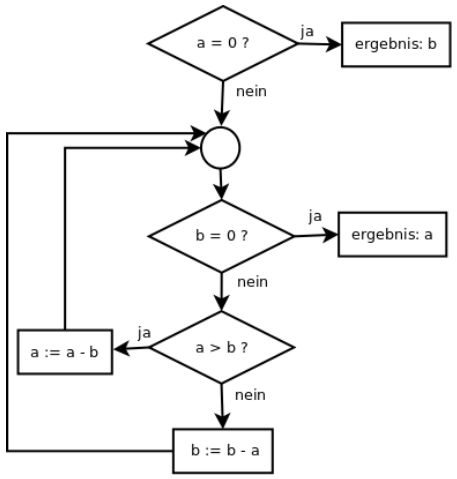
Was ist eigentlich ein Algorithmus?

Was ist der Unterschied zu einem Programm?

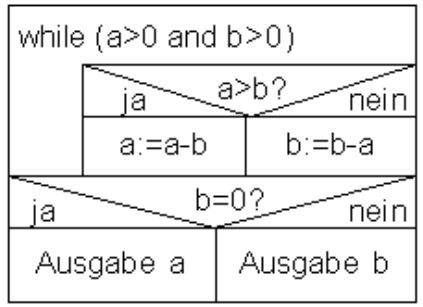
Ein Algorithmus ist eine Verarbeitungsvorschrift zur Lösung eines Problems, die so präzise und eindeutig formuliert ist, dass sie auch von einer Maschine abgearbeitet werden kann.



Flussdiagramm



Struktogramm



Ein Programm ist lediglich ein grammatikalisch korrekter Text (kann auch sinnlos sein).
Ein Algorithmus beschreibt eine allg. Lösung eines Aufgabentyps (unabhängig von der Programmiersprache).

Bsp.: Euklidischer Algorithmus zur Bestimmung des ggT.