

**Aufgabe:** KARA steht vor einem Baum, hinter dem sich ein Kleeblatt befindet. KARA soll um den Baum herumgehen, das Kleeblatt aufnehmen, es einen Schritt weiter ablegen, einen Schritt vorgehen und sich umdrehen (wie auf dem Arbeitsblatt „Das erste Python-Programm“).

**Das Neue:** KARA kann neue Kommandos lernen. Man nennt Kommandos in der Informatik auch **Methoden**. Hier wird gezeigt, wie er die neuen Methoden `turnAround()` und `geheUmBaumHerum()` „lernt“.

```
1 def turnAround():
2     kara.turnLeft()
3     kara.turnLeft()
4
5 def geheUmBaumHerum():
6     kara.turnLeft()
7     kara.move()
8     kara.turnRight()
9     kara.move()
10    kara.move()
11    kara.turnRight()
12    kara.move()
13    kara.turnLeft()
14
15 geheUmBaumHerum() # Aufruf der Methode.
16 kara.removeLeaf() # Kara hat gelernt, wie es geht!
17 kara.move()
18 kara.putLeaf()
19 kara.move()
20 turnAround()      # Kara hat gelernt, wie es geht!
```

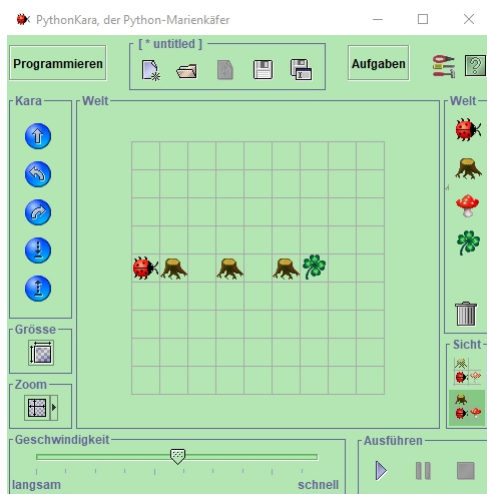
### Erläuterungen:

1. Dem Namen einer neuen Methode folgt ein **Doppelpunkt** und hinter dem Doppelpunkt in einer neuen Zeile beginnt die Erklärung.
2. Die gesamte Erklärung wird **engerückt**. Dabei ist zu beachten, dass alle Zeilen der Erklärung gleich weit eingerückt werden. Die Weite der Einrückung ist nicht festgelegt. Im dargestellten Programm ist die Erklärung von `turnAround()` um 2 Stellen, die von `geheUmBaumHerum()` um 4 Stellen eingerückt.

Beim Schreiben eines Programmes sollte man allerdings die Weite der Einrückungen einheitlich gestalten.

### Zusatzaufgabe 1:

Das Programm soll auch in der unten dargestellten Situation funktionieren. Der Programmtext soll nur geringfügig geändert werden.



### Zusatzaufgabe 2:

KARA soll uns ein kleines Tänzchen aufführen. In der Welt von Aufgabe 1 soll er sich hinter jedem Baum einmal komplett um 360° drehen und dann erst weitergehen. Formuliere dazu erst eine Methode `drehe360()` und rufe sie dann an den geeigneten Stellen auf.