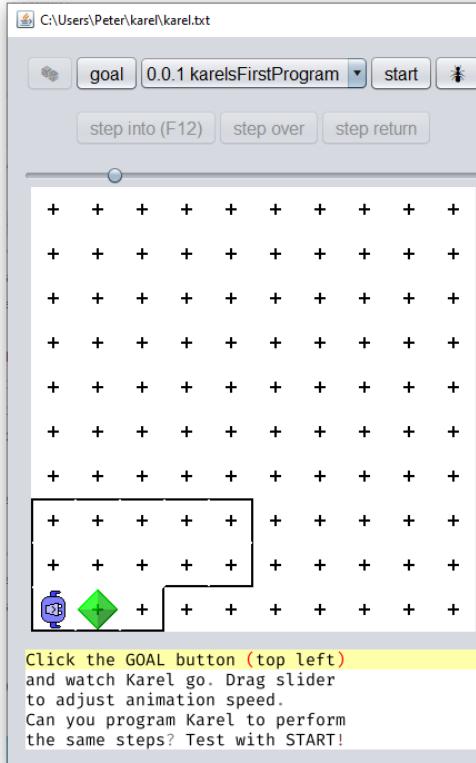


Karel the Robot

Karel the Robot ist ein Spiel mit dem man spielerisch die Grundelemente der Programmiersprachen lernen kann.



The screenshot shows the Karel the Robot software interface. On the left is a 10x10 grid with a Karel robot at the bottom-left corner (row 1, column 1). The robot is facing up and has a beeper. A green diamond-shaped beeper is located at the top-left cell of the grid. The right side of the interface displays a text-based code editor with the file path C:\Users\Peter\karel\karel.txt. The code is written in a pseudocode-like language:

```
1 /*  
2 F1 = moveForward();  
3 F2 = turnLeft();  
4 F3 = turnAround();  
5 F4 = turnRight();  
6 F5 = pickBeeper();  
7 F6 = dropBeeper();  
8  
9 Alt Gr 7 = {  
10 Alt Gr 0 = }  
11 Alt Gr is a single key, located to the RIGHT of the space bar  
12 */  
13  
14 void karelsFirstProgram()  
15 {  
16     // your code here  
17     moveForward();  
18     pickBeeper();  
19     moveForward();  
20     turnLeft();  
21     moveForward();  
22     turnRight();  
23     moveForward();  
24     dropBeeper();  
25     moveForward();  
26 }  
27  
28  
29 void climbTheStairs()  
30 {  
31     moveForward();  
32     repeat(6)  
33     {  
34         oneStep();  
35     }  
36 }
```

At the bottom of the code editor, there is a message: "Click the GOAL button (top left) and watch Karel go. Drag slider to adjust animation speed. Can you program Karel to perform the same steps? Test with START!"

Karel the Robot

Karel the Robot ist in Java geschrieben und benötigt eine **Java Runtime Installation**. Wenn kein Java auf dem Rechner vorhanden ist empfiehlt **fredoverflow** dieses freie javakompatible Paket:

<https://adoptium.net/>

Das Programm selbst kann als **karel.jar** von **fredoverflows** Website bei Github heruntergeladen werden:

<https://github.com/fredoverflow/karel>

Karel the Robot

- **Einfache Befehle:**

Shortcut Command

• F1	moveForward();	Meaning
		Karel moves one square forward in the direction he currently faces. Fails if a wall blocks the way.
• F2	turnLeft();	Karel turns 90° to the left.
• F3	turnAround();	Karel turns 180° around.
• F4	turnRight();	Karel turns 90° to the right.
• F5	pickBeeper();	Karel picks a beeper from the square he currently stands on. Fails if there is no beeper.
• F6	dropBeeper();	Karel drops a beeper onto the square he currently stands on. Fails if there already is a beeper.

Karel the Robot

Einfache Entscheidungen:

Shortcut	Condition	Meaning
• F7	onBeeper()	Karel checks whether a beeper is on the square he currently stands on.
• F8	beeperAhead()	Karel checks whether a beeper is on the square immediately in front of him.
• F9	leftIsClear()	Karel checks whether no wall is between him and the square to his left.
• F10	frontIsClear()	Karel checks whether no wall is between him and the square in front of him.
• F11	rightIsClear()	Karel checks whether no wall is between him and the square to his right.

Karel the Robot

Es lassen sich eigene Befehle schreiben:

Unterprogramme oder Funktionen.

```
void roundTrip()
{
    moveForward();
    moveForward();
    moveForward();
    moveForward();
    moveForward();
    moveForward();
    moveForward();
    moveForward();

    turnAround();

    moveForward();
    moveForward();
    moveForward();
    moveForward();
    moveForward();
    moveForward();
    moveForward();
}

void moveAcrossWorld()
{
    moveForward();
    moveForward();
    moveForward();
    moveForward();
    moveForward();
    moveForward();
    moveForward();
    moveForward();
}

void roundTrip()
{
    moveAcrossWorld();
    turnAround();
    moveAcrossWorld();
}
```

Karel the Robot

Wiederholungen oder Schleifen:

```
void dance()
{
    moveForward();
    turnLeft();
    moveForward();
    turnLeft();
    moveForward();
    turnLeft();
    moveForward();
    turnLeft();
}
```

```
void dance()
{
    repeat (4)
    {
        moveForward();
        turnLeft();
    }
}
```

Karel the Robot

If/else

Hier wird entschieden
ob der Beeper
aufgehoben oder
abgelegt werden soll.

```
if (onBeeper())
{
    pickBeeper();
}

if (onBeeper())
{
    pickBeeper();
}
else
{
    dropBeeper();
}
```

Karel the Robot

If/else if

```
if (leftIsClear())
{
    turnLeft();
}
else
{
    if (rightIsClear())
    {
        turnRight();
    }
}
```

```
if (leftIsClear())
{
    turnLeft();
}
else if (rightIsClear())
{
    turnRight();
}
```

Karel the Robot

While

```
void moveToWall()
{
    while (frontIsClear())
    {
        moveForward(); // This line is executed 0 to 9 times
    }
}
```

Karel the Robot

Not !

```
if (!frontIsClear())
{
    turnRight();
}
```

And &&

```
if (frontIsClear() && beeperAhead())
{
    moveForward();
    pickBeepers();
}
```

Or ||

```
if (!frontIsClear() || beeperAhead())
{
    turnRight();
}
```