

Programming Kara

What you need to know to program Kara for the exercises on this Sheet:

Actions

You program Kara, you write a list of actions that will be processed in order. The available actions are:

<code>move();</code>	move one step in the current direction
<code>turnLeft();</code>	change direction 90° to the left
<code>turnRight();</code>	change direction 90° to the right
<code>putLeaf();</code>	put a leaf on the current position
<code>removeLeaf();</code>	remove leaf from the current position

To use them, you just put them one after another into your program.

Sensors and the If-Statement (Conditional)

Kara can react to its environment using sensors.

The available sensors are:

<code>onLeaf()</code>	returns true if the current tile contains a leaf
<code>treeFront()</code>	returns true if the next tile in the current direction contains a tree
<code>treeLeft()</code>	returns true if tile to the left – again relative to current direction contains a tree.
<code>treeRight()</code>	Same for <code>treeRight</code> , only to the right.
<code>mushroomFront()</code>	true if the next tile in the current direction contains a mushroom.

If-Statement:

To use the sensors, you need a conditional statement: the Java if-statement. (the else part can be omitted)

Example:

```
if (onLeaf()) {  
    // put the commands that should be executed if onLeaf() is true here  
} else {  
    // put the commands that should be executed otherwise here
```

```
}
```

Example without else-case:

```
if (onLeaf()) {
    removeLeaf();
}
```

Defining own Actions for Kara (Java Methods)

(you won't need this before exercise 6)

If you need to do the same set of actions repeatedly, you can define an own action for Kara by defining a Java method:

This is especially useful in Exercise 6, where you need to walk around a tree several times. So you would define a method named "walkAroundTree" in Java like so:

```
public void walkAroundTree() {
    //put instructions to walk around tree here
}
```

You can then use the method as any other action listed above:

```
walkAroundTree();
move();
walkAroundTree();
move();
move();
... .
```

Special Greenfoot Methods for stopping the run() loop

Calling `run()` calls the `act()` method indefinitely. If you want to stop this, you can call

```
Greenfoot.stop();
```

For example:

```
if (treeFront()) { Greenfoot.stop(); }
```

Or use the `stopAfterStep`

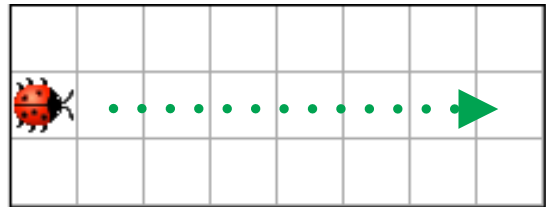
```
stopAfterStep(4); // this will call Greenfoot.stop(); after
the forth call.
```

Exercises

Exercise 1: Walk across the lawn

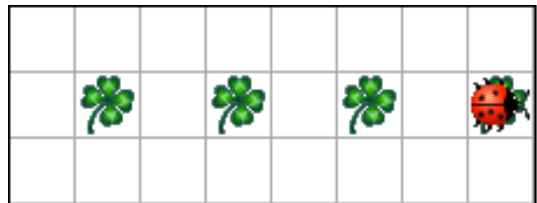
Make Kara walk across the lawn.

Scenario for Greenfoot: 01-01-walk



Exercise 2: Walk across the lawn and drop a leaf on every second tile.

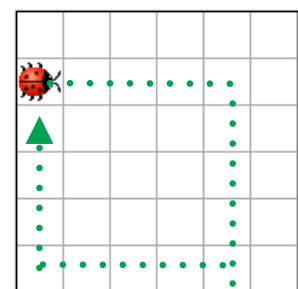
Scenario for Greenfoot: 01-02-walk-drop



This is an alteration of the previous exercise. How can the solution be broken up into repeating steps?

Exercise 3: Walk in a square.

Scenario for Greenfoot: 01-03-square



Exercise 4: Walk in a Square, and drop a leaf in every corner.

Scenario for Greenfoot: 01-04-square-drop

Again, this is very similar to the last one, and can be broken up into repeating the identical operation 4 times. What is a good name for this operation? (method)

