

## Course: Introduction to Computer Science in JavaScript (Bulldog) I Module: Programming With Karel



### Lesson 1.4: Functions in Karel

<https://codehs.com/course/699/lesson/1.4>

<b>Description</b>	In this lesson, students learn in more detail about functions, and how they can use functions to break down their programs into smaller pieces and make them easier to understand.
<b>Objective</b>	<p>Students will be able to:</p> <ul style="list-style-type: none"> <li>• Create functions to teach Karel new commands</li> <li>• Explain the difference between defining and calling a function</li> <li>• Utilize these functions to write higher level Karel programs that go beyond the basic toolbox of commands that Karel starts with</li> </ul>
<b>Activities</b>	<p><a href="#">1.4.1 Video: Functions in Karel</a>  <a href="#">1.4.2 Quiz: Functions in Karel Quiz</a>  <a href="#">1.4.3 Example: Turn Around</a>  <a href="#">1.4.4 Exercise: Pancakes</a>  <a href="#">1.4.5 Exercise: Mario Karel</a></p>
<b>Prior Knowledge</b>	<ul style="list-style-type: none"> <li>• Basic syntax rules in writing commands and functions.</li> <li>• How to create a functions in Karel</li> </ul>
<b>Planning Notes</b>	<ul style="list-style-type: none"> <li>• Revisit student created functions from earlier if needed.</li> <li>• Speakers that enable students to hear music are needed for the add-on activity.</li> </ul>
<b>Standards Addressed</b>	N/A
<b>Teaching and Learning Strategies</b>	<p><b>Lesson Opener:</b></p> <ul style="list-style-type: none"> <li>• Review material from the previous day of class using discussion questions. Have students as a warm up write a function that gets Karel to put <code>threeBalls()</code> on Karel world space. Have students share what they wrote with a partner, and allow one student to write</li> </ul>

their answer on the board. Explain that this lesson will teach students a bit more about functions. [5-10 min]

- Alternatively, use the *Dancing with Functions* handout to have students write out functions for real life problems.

### Activities:

- Watch *Functions in Karel* and complete the corresponding quiz. [5-10 min]
- Encourage students to play around with the *Turn Around* exercise. [5 min]
- Have students complete *Pancakes* and *Mario Karel*. [10-15 min]
  - For *Pancakes*, pay attention to the naming conditions that students use. Make sure they are creating function names that aptly describe what the function is doing. If some students are struggling with naming functions, have them complete the *Naming Functions* handout.
  - For *Mario Karel*, explain to students that their function names must match the ones that they are instructed to create, or else they won't be able to pass the autograders.

### Lesson Closer:

- Have students respond to the discussion questions, or provide the associated handouts as an exit ticket or homework assignment. [5-10 min]

### Beginning of Class:

- What is a function?
  - *A function is a way of teaching Karel a new command using preexisting commands*
- How do we go about creating functions in Karel?
  - *Based on what students know, functions need to be given names, using () at the end, and function at the beginning. The commands given to that function are enclosed in {}.*

### End of Class:

- What does it mean to define a function?
  - *Defining a function is to give the method a series of commands. Whenever the function is called, it will execute the commands in the definition.*
- What does it mean to call a function?
  - *Calling a function means to use the function in one's code. If the function is defined without being called, the commands of the function will not be executed.*
- Describe the steps you would take to teach Karel a new command called `spin()` that has Karel spin around in a circle.
  - *Create and define a function called `spin()` including `turnLeft()` four times. Then call that exact function.*

### Discussion Questions

**Resources/Handouts**

[Dancing with Functions \(student\)](#).

[Naming Functions \(student\)](#).

[Naming Functions \(teacher\)](#).

[Dancing with Functions \(teacher\)](#).

[Debugging Functions \(teacher\)](#).

[Debugging Functions \(student\)](#).

**Vocabulary**

Term	Definition
<a href="#">Define a Function</a>	Defining a function means to teach the computer a new command and explain what it should do when receiving that command.
<a href="#">Call a Function</a>	Calling a function actually gives the command, so the computer will run the code for that function.
<a href="#">Curly Bracket</a>	An open curly bracket is { and a close curly bracket is }
<a href="#">Function body</a>	The part of a function that contains the commands

Modification: Advanced	Modification: Special Education	Modification: English Language Learners
<ul style="list-style-type: none"> <li>Have students create an original Sandbox program that defines and calls a minimum of 4 (four) functions.</li> </ul>	<ul style="list-style-type: none"> <li>Allow students to practice writing functions using the <i>Naming Functions</i> handout.</li> </ul>	<ul style="list-style-type: none"> <li>Complete a flowchart that diagrams how a function works in a program.</li> <li>Allow students to practice writing functions using the <i>Naming Functions</i> handout.</li> </ul>