



Write code. Catch bananas. Save the world

CSED week 2015

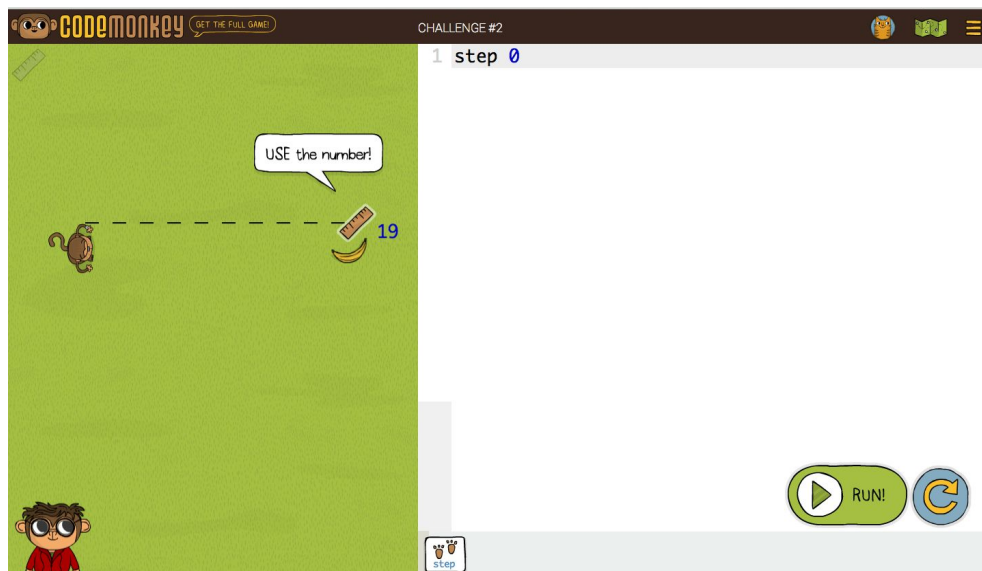
Overview

Our Hour of Code activity is a set of 30 fun challenges that will introduce the basics of computer science to your students. No matter how much experience you or your students have, or if it's your first hour of code, CodeMonkey is engaging, entertaining and educational to all. Students advance through a self-paced progression of challenges in which they help the CodeMonkey catch bananas by writing lines of code. The activity starts off with very simple tasks in order to slowly and gradually teach students different aspects of coding. Along the way your students will meet more cute and funny characters like Gordo the monkey and the helpful turtle.

We also offer **free** access to our teacher dashboard and challenge builder with our [teacher trial](#).

Specifications

- CodeMonkey is best suited for students between the ages of 9-16
- Web based: runs on all web browsers.
- Technical requirements- relatively up-to-date version of either Firefox, Chrome, Explorer or Safari, and an internet connection.





CoffeeScript

The programming language used in CodeMonkey is called CoffeeScript. It's a language that compiles to JavaScript, and similarly to JavaScript it is used in the industry primarily for web applications. We chose this language for a few reasons, but mainly because of its friendly syntax, which resembles the way we write in English, compared to other programming languages.

In CodeMonkey your students will see the impact of their code writing skills right away: The screen is split into two, on the right you write your code, and on the left you'll see your code come to life. For easier access, there are buttons at the bottom to assist with writing some aspects of the code.

Feedback

After every challenge your students will complete, they will get a star score on their solution to that challenge. The stars are distributed as so:

- The first star is given if you got all bananas
- The second star is given if you used what you learned
- The third star is given if your code is short and to the point

We encourage students from the age of 12 and up to try and get 3 stars in all challenges so they can practice another important coding skill, although receiving 2 stars is also great because it shows your students used the new information they learned.



Common core and CSTA standards aligned

Check out the educational standards our activity addresses [here](#) (Common Core, NGSS and CSTA standards). CodeMonkey also teaches students true problem solving skills, as well as multiple-step thinking, critical and analytical thinking skills. It can also show your students that computer programming can be fun and amusing!

Beyond the Hour of Code

Sign up for a free teacher trial to get access to our teacher dashboard and lesson plans. Track your student's progress in the game, see the actual code they write and get access to perfect solutions to all challenges. Plus get a limited time gift for CSEdWeek - free access to our challenge builder for one month, let your students create their own challenges for their classmates to solve!

[Click here to sign up for free!](#)

Get more out of CodeMonkey



In our full course, CodeMonkey has over 200 challenges available, allowing more practice and covering more advanced topics such as for loops, until loops, variables, conditionals, boolean operators and more. We also created a curriculum especially for CodeMonkey that will help guide you through the course. If you and your students enjoy our Hour of Code activity, consider teaching the whole course.

Prep for the Hour of Code

1. Go through the first few challenges or even the full 30 challenges yourself so you can help students during class time.
2. CodeMonkey runs with HTML5 so it should run on your students computers just fine.
3. Sound isn't necessary to succeed in CodeMonkey, but it does make it more fun. If you enjoy the sounds of the safari, you don't need to prepare in advance. Otherwise, you can provide headphones for your students or ask them to bring their own.
4. Don't have enough computers? That's ok, students can work in pairs! Make sure they discuss their solutions and work collaboratively. Working in pairs can sometimes be more productive because discussing the process with someone makes us learn better.
5. Optional: print out [certificates](#) for your students to take home. Or if students have their own email address, they can request a certificate at the [end of the hour of code](#).

Right before your class period

1. Write this link on your whiteboard: www.playcodemonkey.com
2. Load our [CodeMonkey trailer](#)
3. Load the [getting started video](#) (optional)

During the Hour of Code in your classroom

1. To start off your class period, inspire students about computer science by showing them one of [these videos](#), featuring Bill Gates, Mark Zuckerberg, Black Eyed Peas founder will.i.am and NBA star Chris Bosh talking about the importance of programming. (There are [1 minute](#), [5 minute](#), and [9 minute](#) versions).
2. Briefly explain what computer science means. Here's an example:
"Computer science is the art of blending human ideas and digital tools to increase our power. Computer scientists work in so many different areas: writing apps for phones, curing diseases, creating animated movies, working on social media, building robots that explore other planets and so much more. Think about things in your everyday life that use computer science: a cell phone, a microwave, a computer, a traffic light... all of these things needed a computer scientist to help build them."
3. Introduce them to a few tips that will help make your first hour of code go smoothly:
 - "Ask 3 and then me" rule. Students should ask 3 classmates, and if they don't have the answer, then they should ask the teacher.
 - Tell students: "Learning to program is like learning a new language; you won't be fluent right away."



- Persistence is key in computer science, and it's worth it because of the satisfaction when you achieve something that you've worked really hard on.
- Encourage students to run their program often to be able to de-bug (fix) their program more easily. Risk-taking is important in computer science. Scores in CodeMonkey are never affected by how many times you tried to run your solution.
- 4. Show your students the [CodeMonkey Trailer](#). We find that the trailer helps getting the students enthusiastic about coding with CodeMonkey.
- 5. (optional) Show your students our [getting started video](#) to introduce them to our platform.
- 6. Tell students to turn on/log-in to their computer and go to playcodemonkey.com to get started.
- 7. While students are doing the activity, walk around the classroom and answer any questions that they might have. Also, be sure to identify students that are far behind their classmates, or pairs that are not working cooperatively.
- 8. Point your student's attention to our star rating and the meaning of it.
- 9. Once in a while, you can ask for your student's attention and discuss different options to solve different challenges, our recommendation is doing so with challenge 16 and 25.

When your students come across difficulties






- Most of the issues your students will encounter will result from not reading the instructions or the code itself properly. Encourage them to read the instructions/code slowly. They can see the instructions again by clicking on the Gordo (lower left corner).
- Some of the challenges are "de-bugging" challenges, where there is a mistake in the code and you need to fix it. Encourage your students to start each challenge by pressing "run" and watching what the initial code does. This will help clarify where the problem is.
- Encourage students and offer positive reinforcement: "You're doing great, so keep trying."
- It's okay to respond: "I don't know. Let's figure this out together." If you can't figure out a problem, use it as a good learning lesson for the class: "Technology doesn't always work out the way we want. Together, we're a community of learners."






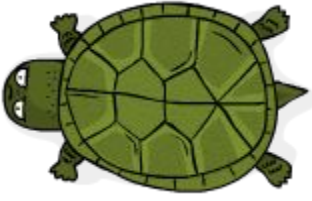
What to do if a student finishes early?

- Encourage them to go back and try to get 3 stars in all 30 challenges.
- Ask students who finish early to help classmates who are having trouble with the activity.

What are the basic things I need to know about computer science and CodeMonkey to teach the Hour of Code?

A program is a set of instructions for your computer to follow. You build a program using commands. Here is a list of the basic information you need to know to play CodeMonkey:

	<p>To make the monkey “step” to a certain distance, we have to write “step X” using the number of steps we want him to take, for example, “step 10”. Pressing the step button will write the word “step” in your code.</p>
	<p>“Turn” should be accompanied by a direction (left/right) or degrees (45, 90, 180). Examples: “turn right”, “turn 90” Pressing the turn button will write the word “turn” in your code.</p>
	<p>“Left” and “right” are used after the statement “turn” to make the monkey turn in the desired direction. Pressing the left or right buttons will write the word “left” or “right” in your code accordingly.</p>
	<p>“turnTo” is another way of turning. Instead of using direction or degrees, we are asking the monkey to turn to a specific object, for example, “turnTo banana”. Pressing the turnTo button will write the word “turnTo” in your code.</p>
	<p>A simple loop is a sequence of instructions that repeats a specified number of times.</p> <p>Example:</p> <pre>3.times -> . . . step 5 . . . turn left</pre> <p>In this example, the monkey will repeat “step 5, turn left” three times. The instructions we write in the loop should be written underneath it with an indentation (....). You can do that by pressing the Tab key on the keyboard. Pressing the times button will write the beginning of a simple loop in your code: “3.times ->”.</p>

	<p>Pressing the “run” button will make the code on the right to run. You can see the outcome by looking at the scene on the left.</p>
	<p>The reset button will erase everything you wrote in the code on the right and will reset the code to how it was at the beginning of the challenge.</p>
	<p>The ruler is a tool to help you measure the distance between different objects in the game, for example, the distance between the monkey and the banana. The ruler can also help you measure angles that the monkey or turtle has to turn in order to face another object on the screen, like a banana. To use the ruler, click it once, and then use your mouse to move the ruler to the point you want it to start measuring from. Click the mouse again, and then drag it to the end point. A number will appear at the end point to indicate the distance. Use this number with the “step” statement. Another number will appear near the starting point, this number indicates the angel from the first object to the second one, use it with the “turn” statement.</p>
	<p>Gordo, named after the first ape in space, is the guide who will help you and give you instructions along the way. His remarks are both funny and helpful. You can always click him to re-read the instructions.</p>
	<p>The monkey is the main character. You will have to help him collect bananas by writing lines of code. Just so you know, monkeys don’t like to get wet, and they are very friendly.</p>
 <pre>turtle.step 10 </pre>	<p>In challenge number 13 you will meet our trusty turtle. The turtle will help you get those sneaky bananas. In order to instruct the turtle to “turn” or “step” we have to click on it first, this will write its name in the code, and then separate it from the action we want it to take using a dot (.) For example: turtle.turn left</p>



Need more help? Contact us:

email - info@cm-studios.com

twitter - www.twitter.com/codemonkeystu

facebook- www.facebook.com/codemonkeystu