CODEMONKEY
LESSON PLANS

CODING ADVENTURE

FUNDAMENTALS

Lesson Plans
1-16

Table of Contents

Table of Contents Cont.

Introduction

Thank you for choosing CODING ADVENTURE to teach your students coding. With fun challenges, cute characters and a unique user experience, Coding Adventure is a great way to introduce your students to the basics of computer science. The following course does not require any prior coding courses or experience and with the following 16 lesson plans, you can jump right into teaching your students real code at any time. The following lesson plans will cover all the challenges featured in CODING ADVENTURE : FUNDAMENTALS .

Each lesson is made up of 3 parts, the introduction, playtime, and debriefing, and is designed to be 45 minutes long. Each section is further divided into the amount of time it takes. At the end of this document you will find a Reference Card that will summarize each coding concept.

For information regarding setting up a class, please read A Beginner's Guide to CodeMonkey. The guide can be found here or in the Teacher's Resources Menu on your homepage. Please feel free to email us at info@cm-studios.com for any questions you may have along the way.

Good Luck!!
The CodeMonkey Team

TOP

Lesson 1 – Let's Get Started

This lesson introduces students to the fascinating world of computers and to the CodeMonkey platform. Some of your students may be familiar with the terms "coding" and "programming" and feel comfortable working with computers. For others, learning to code may be an intimidating experience. As educators, our goal is to help students learn and explore a variety of subjects, including computer science. We want to create an environment where students can learn from their mistakes and build their foundational knowledge to create something new.

Good luck!

TOP

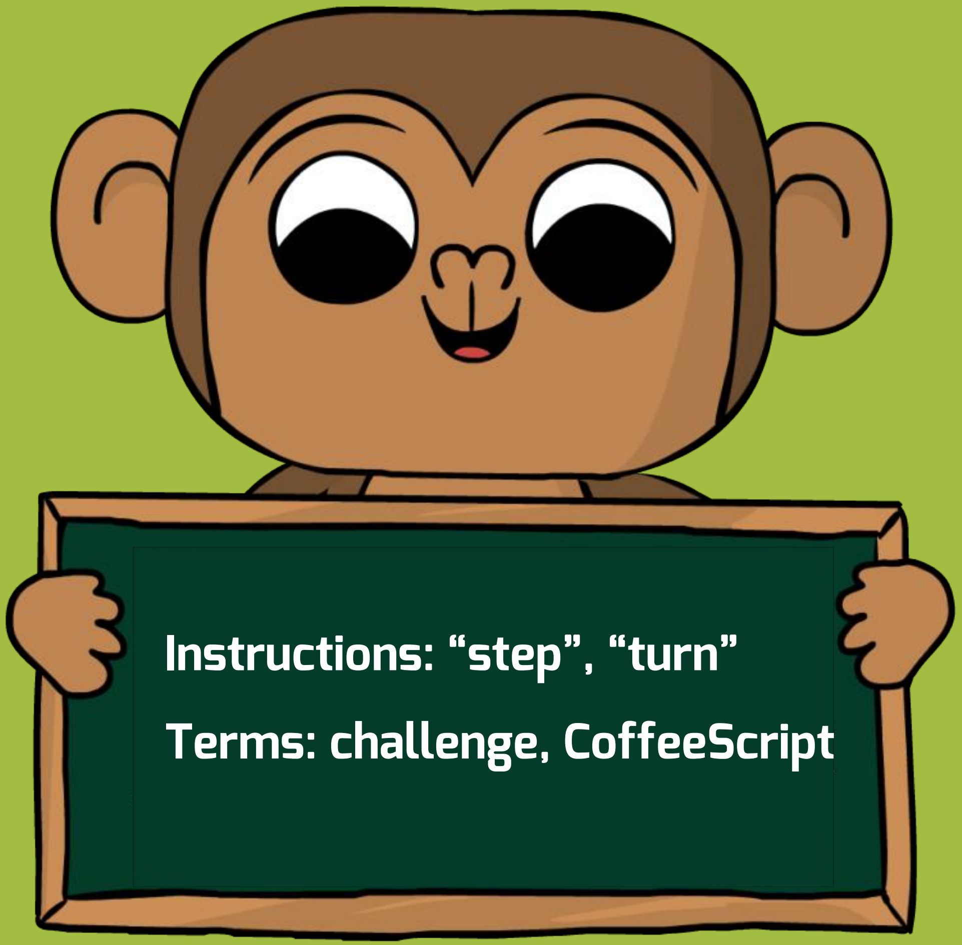
Objectives

In this lesson, students will:

- ☆ Define *coding* and *computer programming*
- ☆ Become familiar with the CodeMonkey platform
- ☆ Complete challenges 0-5

TOP

Components



Instructions: "step", "turn"

Terms: challenge, CoffeeScript

TOP

Part 1: 10 Minutes
Introduction

Discussion	2 mins.
1. How many of you have ever used a computer? 2. Have you ever created something on a computer, like a presentation, a drawing, or maybe even a game? 3. Let two or three students tell the class what they created.	
Explanation	**1 min.**
In order for all of our favorite applications and games to work on a computer, we have to give instructions to the computer. Computers can't think for themselves, they do whatever we tell them to do. Giving instructions to the computer is called computer programming or coding.	
Activity	**3 mins.**
Play a short game with your students to illustrate instructions. Place an object somewhere visible in the classroom. Ask your students to give you instructions to guide you from where you are standing to the object. What instructions did the students use? Step, turn right, turn left?	

TOP

Part 1: 10 Minutes

Introduction Cont.

Discussion	**2 mins.**
Do computers speak the same language as people? Computers have their own languages; they cannot understand human language as we understand it. Java, JavaScript, and Python are just a few of the languages computers speak. Each language is different, but they all have something in common: they require a certain way of thinking, clear instructions and structure. Basically, learning a coding language is just like learning a new language.	
Video	**2 mins.**
Today you will start learning basic coding principles through a game called Coding Adventure. The language we will learn is called CoffeeScript. Show the CodeMonkey Trailer to your class.	

TOP

Part 2: 25 Minutes

Playtime

Log-in Information	3 mins.
Go to www.playcodemonkey.com. Instruct your class on how to log in to their CodeMonkey accounts. If your students use usernames and passwords to login, make sure they store their usernames and passwords where they can easily access them in the future. Optional: hand out user log-in cards. If a student forgets their password, you can reset it by visiting the classroom dashboard, locating the student's username, and clicking on the edit button which will appear if you hover over the username.	

TOP

Part 2: 25 Minutes

Playtime Cont.

Walk-through (1)	**4 min.**
Walk your students through the basic appearance of Coding Adventure: • Click on the “play now” button on the homepage • Watch the short introductory clip • Read the instructions out loud • Coding Adventure is built out of levels called challenges. This is what a challenge looks like. • The editor on the right is where you will write your code. You can also use the buttons at the bottom of your screen for easy access. • On the left is the stage. This is where you will see your code come to life. Your goal is to complete every challenge by helping the monkey catch the banana. • The monkey on the lower left corner is called Gordo; he will give you instructions and sometimes even hints if you are stuck. At any time, you can click on Gordo to see the instructions again.	

TOP

Part 2: 25 Minutes

Playtime Cont.

Walk-through (2)	3 mins.

- In every challenge, you will execute the code by clicking on the “run” button to see what the starting code will do.
- The code on the right says “step 15”, so when we will click on “run” the monkey will step 15 steps forward.
- Click on run.
- We completed the first challenge. After every completed challenge, you will get a star-score rating your solution. 3 stars is the highest score and is rewarded for catching all the bananas, implementing newly-learned topics and writing short code. If you get less than 3 stars, a hint will help you get them all. You can try to solve a challenge as many times as you want, it will not affect your star score!
- Click on replay to see your solution again.
- Edit the solution to change it from

```
step 15
```

to

```
step 5
step 10
```

- Click “run” again to execute your solution. Show the students that this solution only got 2 stars and draw their attention to the hint that tells them how to get the third star.
- Click replay again, fix the solution to get 3 stars and execute it again by clicking “run”.

TOP

Part 2: 25 Minutes

Playtime Cont.

Walk-through (3)	**3 mins.**

- Let us move onto the next challenge, click on "next challenge".
- Read the instructions out loud
- The code on the right says "step 10", let us click on run and see what happens.
- The monkey did not walk far enough, and the hint told us to try "step 15", so let us change the number 10 to 15, and click "run" again.
- It is always a good idea to use the pre-existing code, since it helps guide us to the correct code. Before we try to change the code, click "run" to see what happens, read the hint, and then try to solve the challenge. Since your score does not get affected by running a wrong solution, it is always a good idea to click on "run" before you begin a challenge. After all, the code is there for a reason, so there is no reason to delete it.
- Another good strategy for when you are stuck is to start again from the beginning, in cases like these you can reset your code by clicking on the reset button.
- Click the replay button and then click the reset button to show your students how to reset the code to what it was in the beginning.
- Solve again by editing the code and click "run" to execute the solution.
- Show your students how to go back to challenge 0 by clicking on the map in top right hand corner and clicking on challenge 0. Note that unlike you as a teacher, your students will not be able to skip forward beyond challenges they have not yet solved.

TOP

Part 2: 25 Minutes

Playtime Cont.

Playtime	**10 mins.**
All students should complete challenges 0-5 with at least two stars. (Students from the age of 12 and up should get three stars.) Use your classroom dashboard to keep track of students' achievements. If students are having trouble confusing right and left, draw their attention to the watch on the monkey's left wrist. Tell them that turning in the direction of the watch is left.	
Review	**2 mins.**
Open challenge 2 and show the ruler animation. Follow the instructions to measure the distance between the monkey and the banana, and then use that distance to fix the code. Make sure your students understand how to use the ruler.	

TOP

Part 3: 10 Minutes

Debriefing

Discussion	5 mins.
1. What instructions did you learn today? 2. What did you like most about Coding Adventure? 3. Besides instructions, what else did you learn today? 4. How do you get 3 stars in a Coding Adventure challenge? Does it matter how many times you try to solve the challenge? (No, it does not!) 5. What do you do when you are stuck? 6. In a CodeMonkey challenge, how do you display the instructions again? 7. In a CodeMonkey challenge, how do you reset the code to what it was in the beginning?	

TOP

Part 3: 10 Minutes
Debriefing Cont.

Review	**3 mins.**
Open challenge 6 and solve it with your class. They will solve it by themselves in the next lesson.	
Assignment	**2 mins.**
Due next lesson, create a map with your route to school by writing the directions as computer instructions, just like you learned today. You can also route the way from your room to other places in your house, or even from your homeroom at school to the playground. Be sure to use the basic instructions used in CodeMonkey. Show an example of such a sequence of instructions on the whiteboard.	

TOP

Reference Card

Keyword/Button	Description
step	To make the monkey “step” a certain distance, we have to write “step X” using the number of steps we want the monkey to take. For example, if we want the monkey to walk 10 steps, we will write “step 10”. Pressing the step button at the bottom of the editor will enter “step” into your code.
turn	“Turn” should be accompanied by a direction (left/right) or degrees (45º, 90º, 180º). Examples: “turn right”, “turn 90” Pressing the turn button will write the word “turn” in your code.
left right	“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.
turnTo	“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.

TOP

Reference Card Cont.

Keyword/Button	Description

times	A simple loop is a sequence of instructions that repeats a specified number of times.
	For example:
	`3.times ->`
	`    step 5`
	`    turn left`
	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 - >”.

`x = 10`
`step x`
	Assignments to variables. A variable is like a storage unit: we store data in it, and we use it only when we need it. An assignment to a variable is constructed out of an identifier and a value. This separation of name and value allows the name to be used independently of the information it represents. We can use X when writing the program, without knowing what its value will be when the instructions will be carried out.

TOP

Reference Card Cont.

Keyword/Button	Description
 say	“say” will make a speech bubble appear next to the monkey with the text we entered, for example: will make the monkey say “Boo!” We use quotation marks (“ ”) around the phrase we want the monkey to say in order for the computer to understand that the text we entered is not a variable. Try using “say” when there is a rat around. Pressing the say button will write the word “say” in your code.
 distanceTo	“distanceTo” is used with another statement like “step” or “say” and an object. Using “distanceTo” is like asking a question, for example, “What is the distance to the banana?” The answer is a number, calculated by the computer, that represents the distance. For example: `step distanceTo banana` The computer will measure the distance between the monkey and the object (banana). Then it will use the resulting number to carry out as instructed, using the measured value as the argument for “step”. Pressing the distanceTo button will write the word “distanceTo” in your code.

TOP

Reference Card Cont.

<table>
<tr><th>Keyword/Button</th><th>Description</th></tr>
<tr><td>for</td><td>

This is a “for” loop. A “for” loop is used when we have a collection of objects and we want to repeat an action that relates to each one of them specifically. The “for” loop will keep going until all the actions are done on all the objects in our collection (array). When the computer executes this loop, it replaces the variable name with the first item in the collection. After it is done with the first item, it moves on to the second, and so on. We can also use a “for” loop inside of a “for” loop; the example on the left is taken from challenge #75. Pressing the for button will write the following text in your code.
Note the comment line –

```
for name in array
    # Your code here
```

Example:

```
for b in bananas
    for c in crocodiles
        c.turnTo b
    turnTo b
    step distanceTo b
```

</td></tr>
<tr><td>RUN!</td><td>Pressing the run button will make the code on the right run. You can see the outcome by looking at the scene on the left.</td></tr>
</table>

TOP

Reference Card Cont.

Keyword/Button	Description
↻	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.
📏	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 angle from the first object to the second one, use it with the “turn” statement.

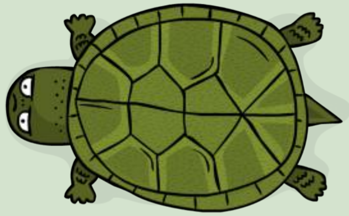
TOP

Reference Card Cont.

Keyword/Button	Description
	After each challenge, you'll receive a star-rating for your solution. The stars are distributed as so: ● First star is given if you got all bananas ● Second star is given if you used what you learned ● Third star is given if your code is short and to the point

TOP

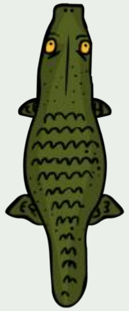
Character Review

Character	Description
	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.
	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.
	In challenge #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 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.step 10`

TOP

Character Review Cont.

Character	Description
	In challenge 55 you will meet the beavers. The Beavers like wood very much, and they have agreed to help you cross the water and get more bananas by stepping on their wood. They can only “step” forward and backwards. To use them we need to use a dot (.) between their name and the action we want them to take. For example: `beavers[0].step 10`

	Crocodiles are introduced in challenge #66. They are used to form a bridge on the water, to help the CodeMonkey reach bananas. They can only “turn” or “turnTo”. We usually use crocodiles with “for” loops.
	For example:
	`for c in crocodiles`
	`    c.turn right`

TOP