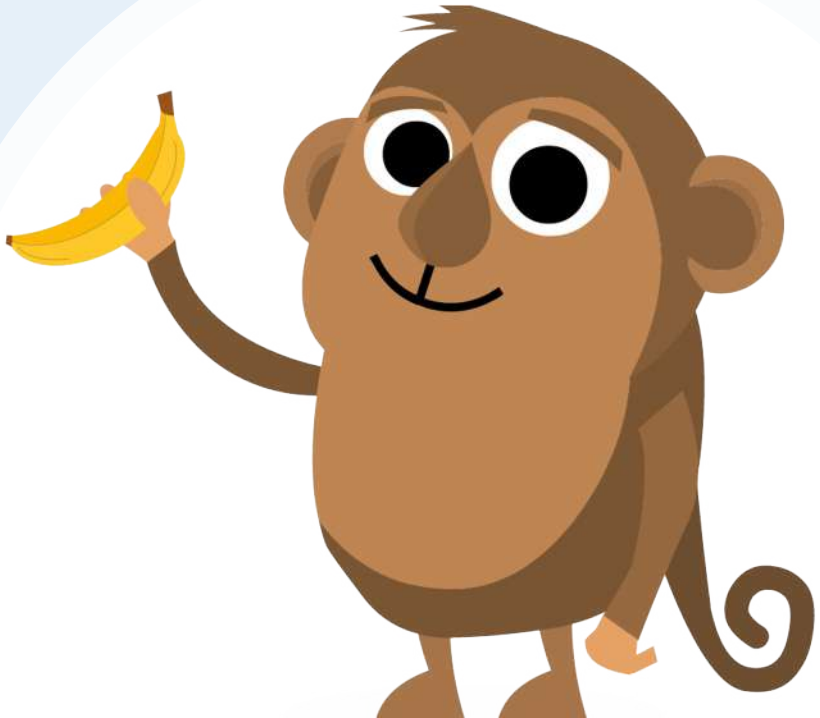
CodeMonkey
LESSON PLANS

CodeMonkey Jr.

A PRE-CODING GAME FOR 4-6 YEAR OLDS

Sequencing & loops

Lesson plans
1-10



CodeMonkey
LESSON PLANS

2345 Yale St., 1st floor
Palo Alto, CA 94306
info@codemonkey.com
www.codemonkey.com

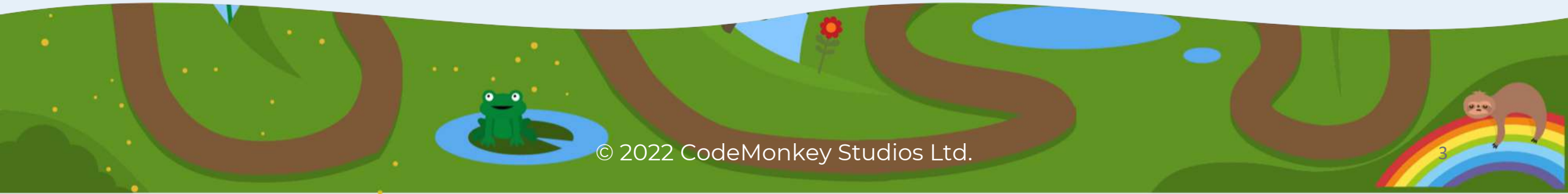
Introduction

Thank you for choosing to introduce your preschoolers to the basics of computational thinking and coding principles through gaming. This course is recommended for **4-6 year-olds**. The following 10 lessons will cover 2 chapters (sequences and loops) in , which include a total of 30 challenges. These challenges will introduce students to fundamental concepts and terms from the world of computers and programming.

Each lesson is set to be 35 minutes long and is made up of 3 parts: an Introduction, Playtime and Debriefing. If you think you need more time to complete a lesson, feel free to split it into 2 sessions. Not all of the lessons are based upon playing the game itself as some are more theoretical and teach new concepts. All of the lessons are playful and designed for younger students. Not all lessons require a computer or tablet and active Internet.

Before you get into the lesson plans, we recommend reading CodeMonkey Jr.'s Teacher Guide. At the end of this file, you will find a Glossary that summarizes each coding concept. Please refer to it throughout the course. Feel free to email us with any questions or comments.

Good Luck!
The CodeMonkey Team

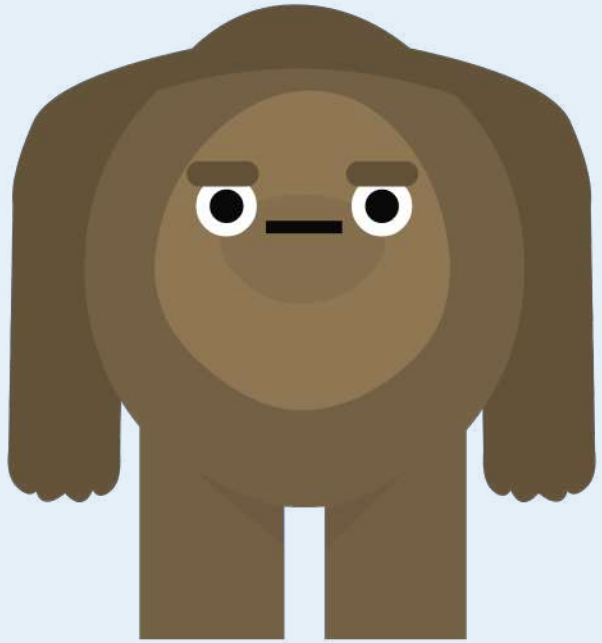


SOME NOTES ON TEACHING CODEMONKEY JR.

- **ALL Beginners are welcome!**
  - We will provide you with everything you need in order to learn the basics of coding and enjoy teaching the game. You do not need *any* background in Computer Science!
- **A love of learning is a plus**
  - A love of learning is key for learning new topics like programming, which may seem intimidating at first. Don't worry though, we will provide you all the information you need to successfully teach your students coding.
- **A love for playing is also a plus!**
  - We believe learning through playing is a joyful and engaging way to introduce concepts to your students. CodeMonkey Jr. will open up a new world for your students so give them time to replay the game if they want. Afterall, we want students to have fun.
- **Mistakes are OK!**
  - Mistakes are the best way to learn so use them to challenge and engage your class. You can even encourage students to play with wrong sequences to see where the monkey will go.

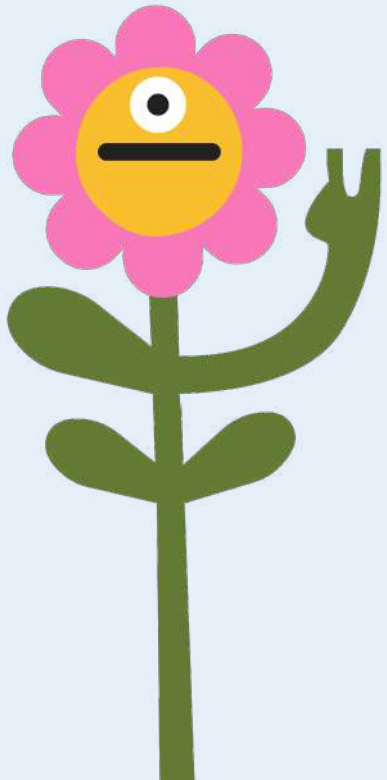
Guidelines

- We recommend going over the Teacher Guide and the entire lesson plans before starting the course
- The following lesson plans are only suggestions so feel free to change, edit and adjust them to your class. At the end of the day, only you know what is best for your students!



Technical Requirements

- You will need tablets or desktops for students. You can use 1 device per 2 students.
- It is best if your computer is connected to a projector or screen so that you can present challenges.
- An Internet connection is required.



LESSON PLANS

Table of Contents

NOTE: The type of lesson can be either theoretical or practical. **Theoretical** signifies lessons that take place without playing CodeMonkey Jr.. They serve to introduce concepts to your students and help them understand them through offline activities. **Practical** lessons take place online, using the CodeMonkey Jr. platform. In these lessons students will solve challenges on CodeMonkey Jr..
Looking at the type of lesson in advance can help you know when you need to reserve devices for your class.

Lesson 1 – Intro to Computers

Through this lesson, students will understand what a computer is and how almost everything in our world is digitized or computerized.

U.S. STANDARDS ADDRESSED

CSTA-K12 Computer Science Standards	
• 1A-IC-16	

PREPARE IN ADVANCE:

For this activity, please prepare Images of different computers over time (from the first computer to the ones we use today). See next slide for examples.

PART 1: 15 MINUTES
INTRODUCTION

Discussion	**15 mins.**

Your class' understanding of computers and technology may differ from student to student. The following discussion will help you understand their challenge. If you see that most of your students lack understanding on what a computer is or what it does, consider splitting this lesson into two parts.

Lead a discussion with your class about the computers in their lives by asking the following:

1. **What is a computer?** Let students define a computer in their own words and terms. You can guide them with a leading question such as: *"Imagine that a kid from the Stone Age is coming to class and you need to explain to him or her what a computer is. What would you say?"*
    - Your students will probably start with answers related to what you use computers for.
    - Try to lead with questions that will focus them on defining what computers are. Provide them with an example such as is a computer an object or person? Is it mechanical or digital? Is it durable or fragile?
2. Ask students what types of devices they have at home.
    - Present different computer images - desktops, tablets, smartphones, etc. (see next slide)
3. Discuss the fact that computers today go beyond the standard devices we all know. For example, autonomous cars, smart TVs, robots, etc. are all computers.
4. Ask students what they do with computers at home.
    - What do other family members do?
    - Examples could be playing games, searching the web, watching videos, working, etc.
5. Ask them what is connected to their computer. Examples could be a keyboard, mouse, screen, camera, etc.
    - This is a good point to speak about the difference between software and hardware - explain that we develop the software and without software, computers are just a machines (or hardware) that cannot do anything.

COMPUTER EVOLUTION

GOOD TO KNOW IN ADVANCE:

- The first computer was invented more than 70 years ago, it was the size of a room
- The evolution of computers started from very large computers to small devices like the tablets we use today. They became smaller and more efficient over time

Useful links:
Computer facts for kids
Next steps in computers evolution

PART 2: 15 MINUTES
PLAYTIME

PREPARE IN ADVANCE:

Download Akinator for free on the App or Google Play Store or use the web link

Make sure to Turn ON "Child Mode"!

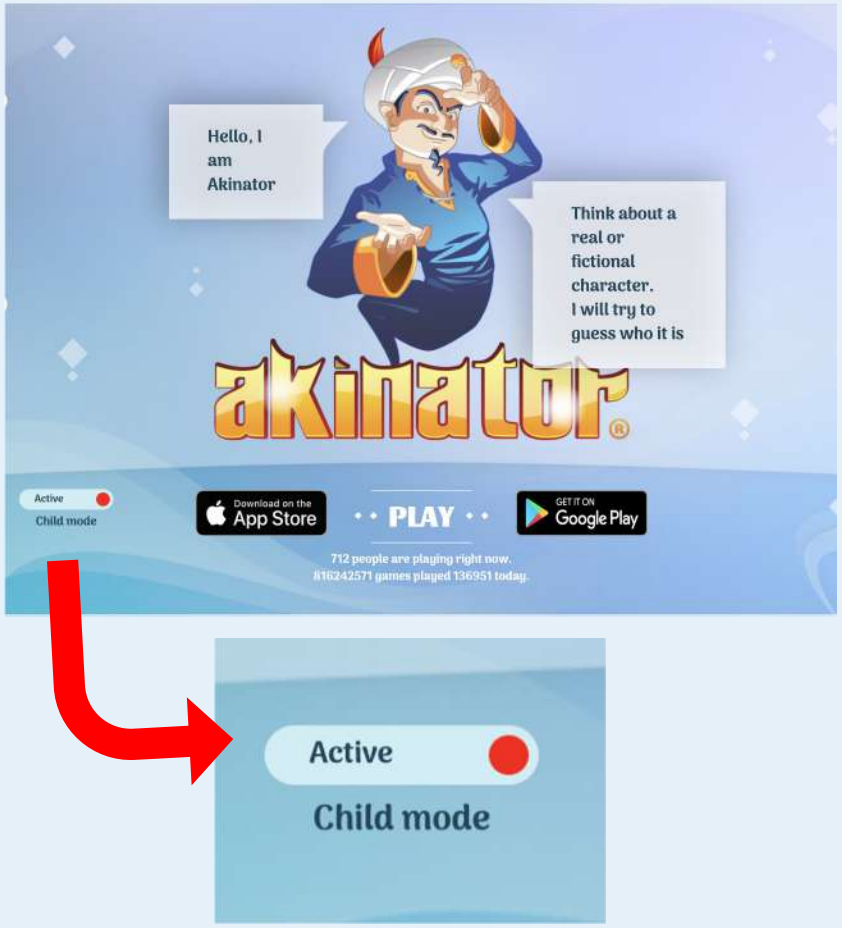
Present	**15 mins.**

This part of the lesson will be based on a 6-minute video and the Genie Guessing Game

- Start with this awesome introductory YouTube video on what computers are for kids.
- After the video, play the Genie Guessing Game on Akinator with your class
  - Ask students to think about a real or fictional character. The genie will try to guess who it is.
  - After the game, ask students:
    - Does the genie know everything? Yes
    - Is there a genie in the computer? No, it is software that was developed by programmers

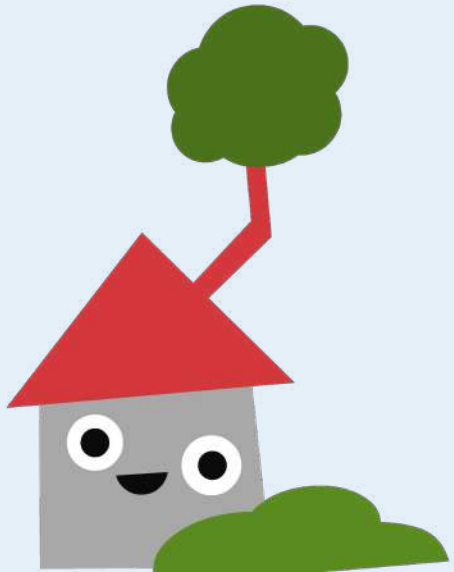
How is this game connected to computer programming?

When kids play the game, they get the impression that computers can read their thoughts, but that is not the case. Somebody programmed this game to be smart enough to have enough information to be able to "read their thoughts", ask the right questions and ultimately guess the character that they are thinking of.



PART 3: 5 MINUTES
DEBRIEFING

Discussion	**5 mins.**
Ask your class to name something that they learned today. Remind them of the following: 1. Computers - desktops, laptops, tablets 2. Keyboard, Mouse, Screen 3. Software - games, videos, emails Ask students about what new computers or software ideas they would like to invent.	



Lesson 2 – Intro to coding

After this lesson students, will know that what it means to make a computer perform a task. They will also practice basic algorithms.

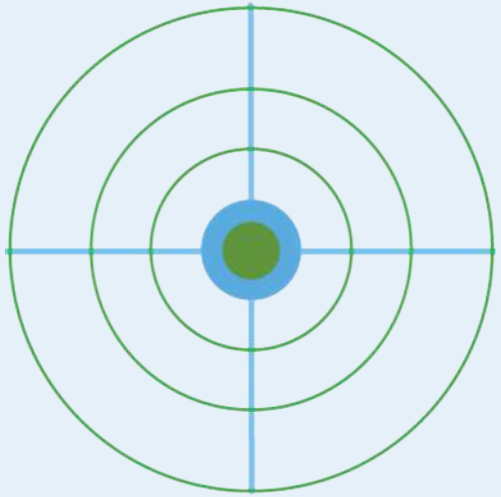
The lesson covers the following:
-Logic
-Algorithms
-Sequencing
-Direction/orientation
-Counting

U.S. STANDARDS ADDRESSED

CSTA-K12 Computer Science Standards	
• 1A-AP-11	

PART 1: 15 MINUTES
INTRODUCTION

Play	**10 mins.**
Game #1 - Instruction-based drawing 1. Print or draw a shape like the bullseye on the right (simplify it as you see fit) 2. Ask for a volunteer 3. Give the volunteer the drawing 4. Handout paper and pens to all of your students 5. Ask the volunteer to guide the class to draw the bullseye, without showing the paper to them	
Discuss	**5 mins.**
Ask students to present their drawings and have the volunteer show the bullseye. • How many drawings look like the original? How come? • What was difficult about the game? Ask both the volunteer and the class.	

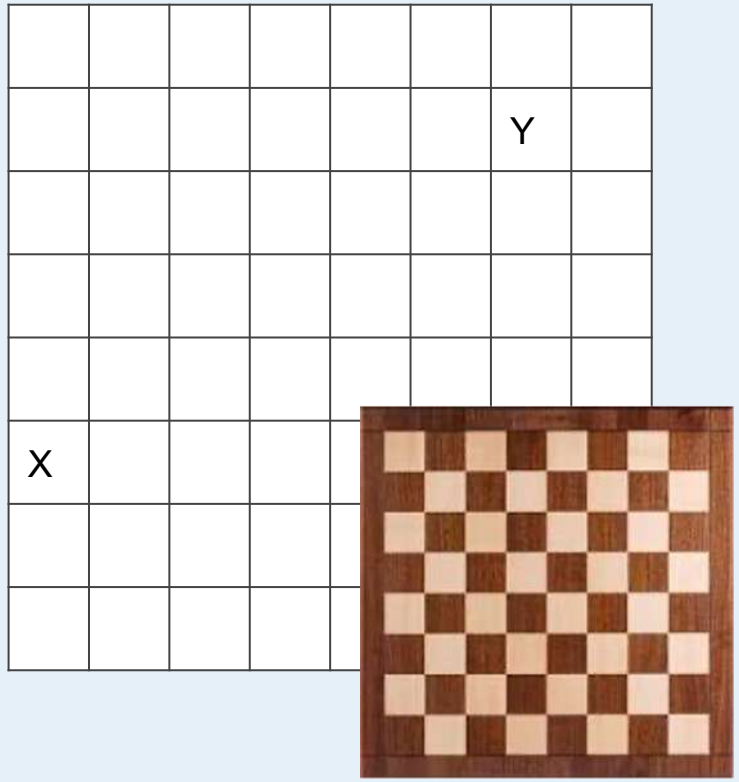


PART 2: 15 MINUTES
PLAYTIME

PREPARE IN ADVANCE:

- An 8 x 8 inch grid with two objects placed on the grid
  - The grid can be a chessboard or chalk grid that you drew
- Index cards with the following Instructions:
  - turn left, turn right, move one tile forward (see next slide)

Playtime	**10 mins.**
Game #2 - Software Vs. Computer In this game, students will learn how computers and software work through visual representation. Choose two volunteers - Student 1 and Student 2. Student 1 will be the programmer. He or she will be in charge of building up a set of instructions (the program) given to the computer (Student 2). In this case, the program will be responsible for getting the object from point X to the object on point Y. Student 2 will act as the computer. He or she will move the objects based on the exact instructions given by the Student 1 (the programmer).	
Discussion	**5 mins.**
After this game, students will understand the importance of the following: • Defining clear steps and giving accurate instructions for students to follow • Counting from one space to the next • Actively following instructions • Analyzing wrong instructions • Giving accurate instructions at a pace that the student can follow	



INSTRUCTIONAL CARDS

Instruction	Meaning
	turn right
	turn left
	Move one tile forward

Tip:

- Let students choose their own two objects (i.e. toys) that will be placed on the board
- Students may have difficulties in differentiating between left and right so you may print cards in different colors
- Start the game with a simple sequence where the instructions are only move forward.
- After one or two rounds, make the path more complex.

PART 3: 5 MINUTES
DEBRIEFING

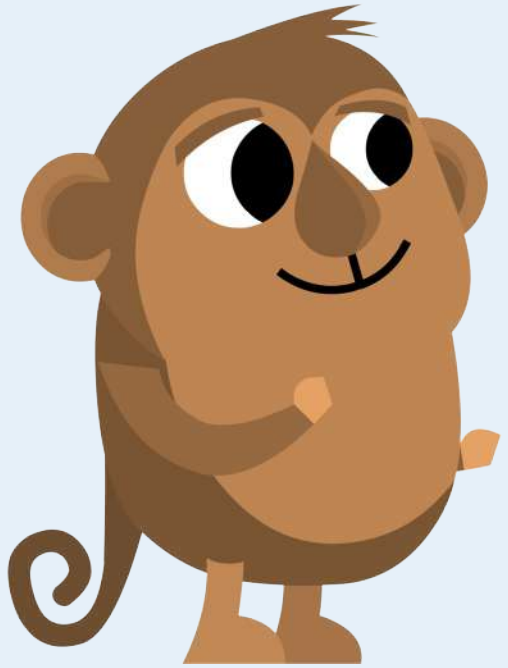
Discussion	**5 mins.**

Explain how the game mimicked what it is like to be a programmer in real life

- What was the main challenge in both games?
  - Defining clear instructions
- Why is it important to provide clear instructions?
  - When instructions are not precise, student 2 could interpret a different meaning that what student 1 meant. This would lead to the game reaching a different outcome.
- Have you ever been given unclear instructions and ended up doing the wrong thing? When?
  - For example, what happens if your parents tell you to put on your shoes but then you forget to put on your socks first

Lesson 3 – Meet the Monkey

After this lesson, students will understand the objectives of the game and learn how to play CodeMonkey Jr.. By end of this lesson, students will complete challenges 1-4. Students will also learn about the move-left and move-right blocks.



Please note that the minimum class time required for this lesson is 35 minutes. If you are rushed for time, please skip to Part 2.

U.S. Standards Addressed

CSTA-K12 Computer Science Standards	
• 1A-AP-14	

PART 1: 10 MINUTES
PRESENT

Introduction	**10 mins.**

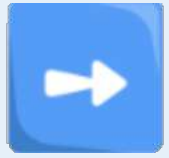
Introduce CodeMonkey Jr. to your students. The game takes place in a world filled with captivating creatures and bright colors, where students need to help a monkey collect bananas and unlock a treasure chest. In order for the monkey to get to the treasure chest, students need to code the path. The coding instructions are carried out through graphical blocks. In the scope of this version, the game consists of four chapters that either increase in difficulty or introduce a new concept.

1. Open the game and present the home screen.
2. Enter the first chapter (Sequencing) and go to the first challenge. This challenge has an animated tutorial.
3. Instructions can be either tapped or dragged to the code line.
4. Once the code is complete, tap play to see where the monkey goes.

User Interface

Home screen with 2 chapters

First challenge with an animated tutorial



Move left/right instructions

PART 2: 20 MINUTES
PLAYTIME

Playtime Flow	**20 mins.**
In the following lesson, we will play challenges 1-4. After presenting challenge 1, emphasize the following: 1. Counting: Each tile is a single step. Make sure all students can count to 10. 2. Directions: Go over the difference between left and right Playtime can take up to 20 minutes. Students will get to challenge 4 or higher. If they understand the game fast, consider combining lessons 3 & 4.	

Challenge	**Scope**
1	Introduction/tutorial
2	1 step left
3	2 steps right + catch banana
4	3 steps left + catch banana

Tips & Tricks -
Before the end of the lesson, walk between students to see which challenge are they at and if they need help.

Learning from errors - Make sure you spend time presenting to students wrong solutions. This is very important in developing the ability to analyze the cause of the error and how to correct it.

PART 3: 5-10 MINUTES
DEBRIEFING

Discussion	**5 mins.**

The students are now real programmers.

- Before the end of the lesson, have students complete the challenge they are at.
- Compliment students on their first programming/coding experience.
- Lead a discussion with the students:
  - What does it mean to be a programmer?
    - Speak about solving problems
  - Ask students when they need to solve problems in their daily lives
  - What challenge is the monkey facing?
    - Speak about two matters - getting to the treasure chest and collecting bananas
  - What did you do to help the monkey?
    - Designed the right code to help it get to the treasure and bananas
  - Is there only one right way to solve a problem?
    - There are challenges where you can solve them in more than one way and still get 3 stars. There are also challenges where you can get to the chest without 3 stars using different code.
  - Programming as a language - discuss the fact that they have learned a new language
  - What signs/words do we have in our coding language?
    - So far, we learned 2 words - move right and move left. In the next lessons, we will learn new words.

Lesson 4 – Finding Treasure

In this lesson, students will continue their progress by playing challenges 5-8. Your class will learn the jump blocks (jump right and jump left). This lesson will also focus on advanced planning and solving subproblems in order to solve the bigger problem.

Please note that the minimum class time required for this lesson is 35 minutes. If you are rushed for time, please skip to Part 2.

U.S. STANDARDS ADDRESSED

CSTA-K12 Computer Science Standards	
• 1A-AP-12	• 1A-AP-14

PART 1: 5 MINUTES
PRESENT

Discussion	**5 mins.**

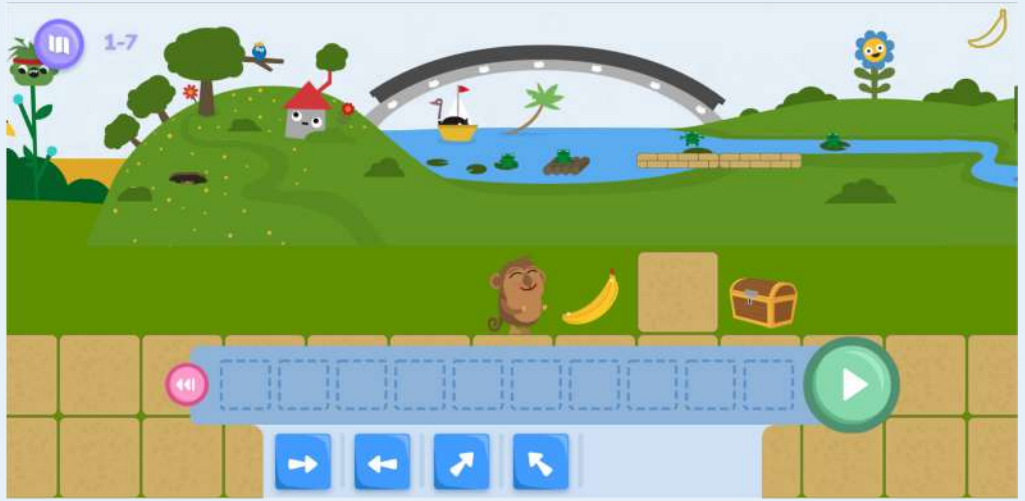
This lesson is dedicated to playing CodeMonkey Jr.. Students will learn 2 new blocks and will deal with more complex pathways.
Explain the following:

- The monkey may need to jump or go over tiles - he can jump up in both directions.
- Going down is done with the move left/right blocks.

PART 2: 25 MINUTES
PLAYTIME

Introduction	**5 mins.**
How a programmer works: • Plans in advance • Defines the problem - cuts it into small chunks • Builds the sequence for each chunk	
Playtime	**20 mins.**
Have students solve challenges 5-8 (see table below)	

Challenge	**Scope**
5	1 step with jump right
6	2 steps with jump left
7	1 step with jump right and walk down
8	3 steps with jump right and walk down



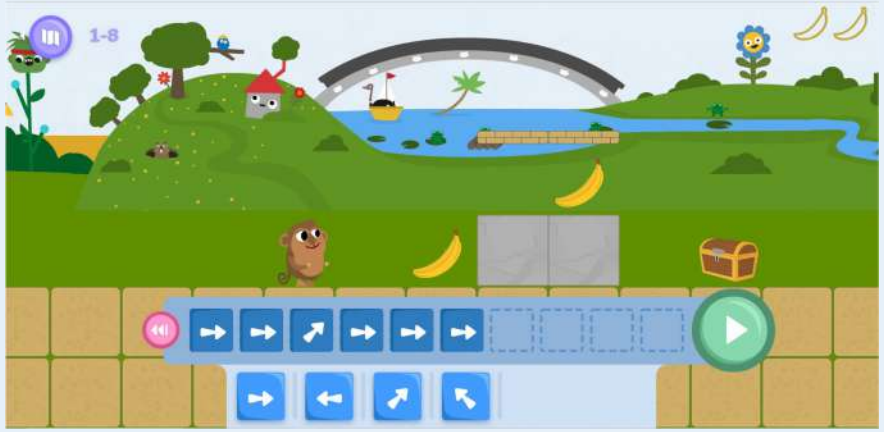
PART 3: 5 MINUTES
DEBRIEFING

Discussion	**5 mins.**

Explain that when a challenge is long and includes several instructions, there is a distinct need to plan in advance.
It helps to define goals/sub-sequences, build the sequence and then define the next goal until you complete the sequence. Make sure you count the steps and walk in the right direction.
In the solution for challenge 8 (see below), you can see the 3 parts:

- 2 steps right
- Jump right
- 3 steps right

Building the solution based on the sub problems makes it easier.



Lesson 5 – Collect all bananas

This lesson lets the students practice more their planning of the code. They need to first get the bananas, then go to the treasure chest. In some challenges, they will need to go in both directions or jump up to get the bananas.
By end of this lesson, students will complete challenges 9–12.

U.S. Standards Addressed

CSTA-K12 Computer Science Standards	
• 1A-AP-12	• 1A-AP-14

Part 1: 5 Minutes
Present

Discussion	**5 mins.**

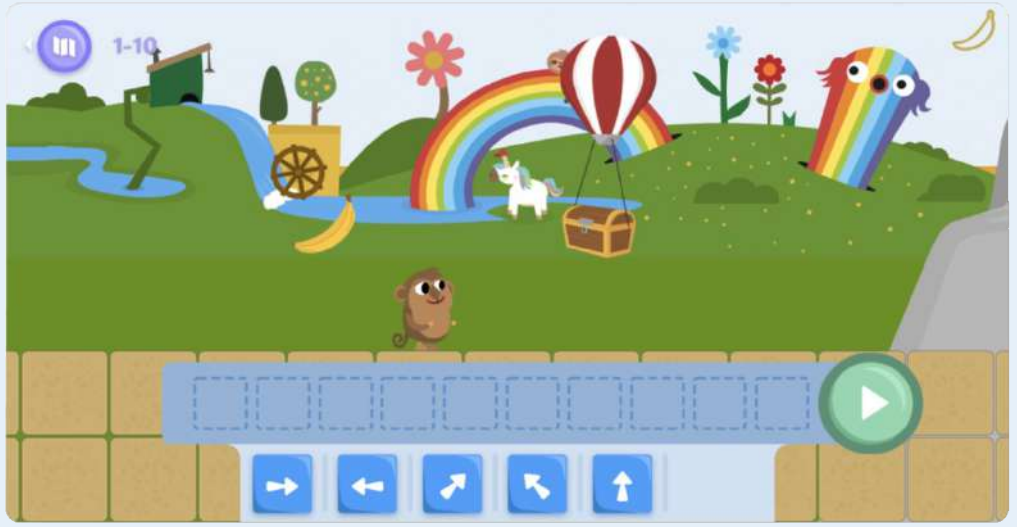
This lesson is dedicated to playing CodeMonkey Jr.. Students will learn another new block and will deal with more complex pathways.
Explain the following:

- Jumping up moves the monkey up one tile and back down again
- It does not advance the monkey to the left or right, but helps him get higher bananas or if the treasure is in the air

Part 2: 25 Minutes
Playtime

Playtime	**25 mins.**
Have students solve challenges 9-12 (see table below). Encourage students to earn 3-stars (or at least 2) and to not leave any bananas behind.	

Challenge	**Scope**
9	Jump up
10	4 steps with 2 jump up (first move left to get the banana and then move right to get the treasure)
11	2 steps with jump right and walk down
12	6 steps with jump right and walk down



Part 3: 5 Minutes
Debriefing

Discussion	**5 mins.**
Explain that in this lesson some of the bananas were not in the monkey's path. The monkey has to jump to get them or move to the opposite side first. Encourage the students to solve the challenges when collecting all the bananas. Tell them that although the code is longer, their mission is to collect both the bananas and get to the treasure chest.	

1-star solution

3-stars solution

Lesson 6 – I have a plan

In this lesson your students will continue to practice sequencing.
They will need to plan their journey to collect all the bananas and get to the treasure chest.

Once completing this lesson, your students will finish the first chapter of CodeMonkey Jr - sequencing by solving challenges 13–15.

U.S. Standards Addressed

CSTA-K12 Computer Science Standards	
• 1A-DA-07 • 1A-AP-12	• 1A-AP-14

Part 1: 10 Minutes
Present

Activity	**5 mins.**

Place chairs in a row. Place two items on two chairs. For example:

1. window/table
2. empty chair
3. book on chair
4. empty chair
5. bag on chair
6. empty chair
7. door

Ask for a volunteer. Sit them on chair #4. Tell them they need to collect both the book and the bag and get to the door with minimum moves. Ask them which item they think they need to collect first – the book or the bag.

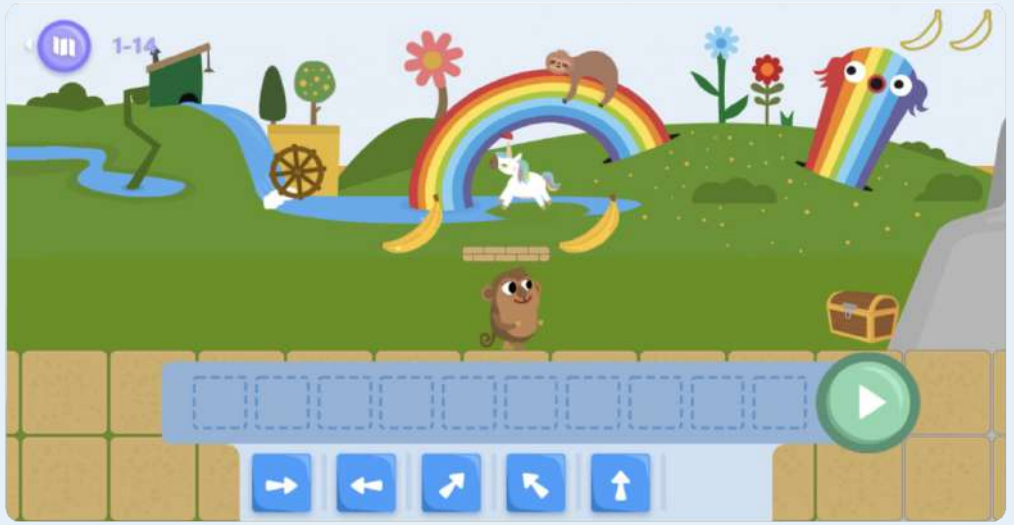
They need to describe their steps: move right, get book, move left, move left, get bag, move left, move left.
You can tell them to describe both options, to see that the other one is longer.

Repeat the activity, this time tell them they need to get to the window/table instead of door.

This activity emphasize some of the challenges, where we first need to move to one side and then move to the other side. Also, the fact that it is shorter to first get the item further from the target and then move to target.

Part 2: 20 Minutes
Playtime

Playtime	**20 mins.**
Have students solve challenges 13-15 (see table below). Encourage students to earn 3-stars (or at least 2) and to not leave any bananas behind.	



Challenge	**Scope**
13	4 steps with 2 jump up (start with jump up to get the banana)
14	6 steps with 2 jump up (first move left to get the banana and then move right to get second banana and the treasure)
15	3 steps with 3 jump right and walk down

Part 3: 5 Minutes
Debriefing

Discussion	**5 mins.**
Open challenge #15 and ask your students how the solve it. Ask your students if they tried solving it with just the jump right block? What do they think will happen? Try it with your class and see that the monkey is blocked by bricks. This is a classic challenge where they need to plan their moves: start with 2 jumps right, then move right twice, jump right and lastly move right.	

Please note that now you can either:
- *start the loop chapter – continue to next lesson*
- *start the advanced sequences chapter – open this lesson plan (lesson 1-4)*

Lesson 7 – On Repeat

In this lesson, the students will learn the concept of a loop.
There are several types of loops in programming. In our scope, students will learn how to use a preset conditional loop.

By end of this lesson, students will complete challenges 1-2.

Loop - repeat an action until asked to stop

Please note that the minimum class time required for this lesson is 35 minutes. If you are rushed for time, please skip to Part 2.

U.S. STANDARDS ADDRESSED

CSTA-K12 Computer Science Standards	
• 1A-AP-10 • 1A-AP-12	• 1A-AP-14 • 1A-AP-15

Part 1: 5 Minutes
Present

Discussion	**5 mins.**

What happens when you need to perform the same action many times?
For example, what would the code look like if the monkey needs to walk 7 tiles to the right?

- MR-MR-MR-MR-MR-MR-MR

Is there a shorter way?
What about the following ways?

- Move right 7 times
- Move right until you reach the goal

PREPARE IN ADVANCE:

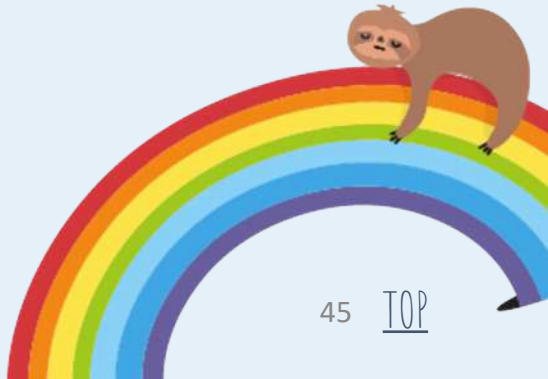
- Dice for the class - each pair of students will need 2 die

PART 2: 20 MINUTES
PLAYTIME

Instructions - The Dice Game	**15 mins.**
Split the class into couples and give each pair a set of two dice. The game is simple - the first one who gets a double wins the round. Ask students to document the game by writing how many throws they needed each round to get a double. Repeat this game 3-4 rounds (depending on the students).	
Play	**5 mins.**
Ask students to open Chapter 2 and play challenges 1 & 2. These challenges are simple sequences which means that a single instruction is repeated several times. For example, challenge 1 requires 7 MR (move-right).	

PART 3: 5 MINUTES
DEBRIEFING

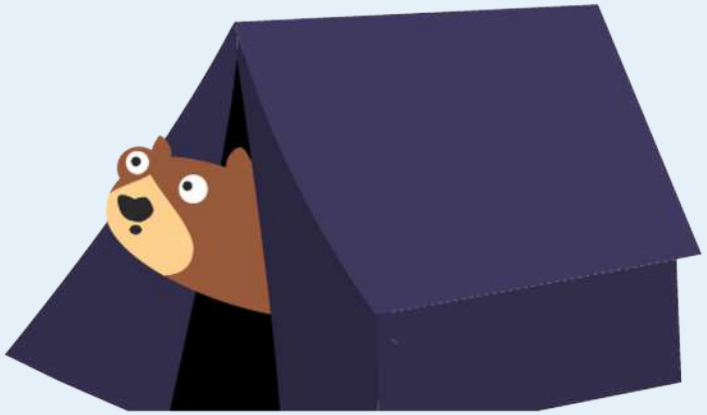
Discussion	**5 mins.**
Ask students for feedback about challenges 1 and 2. Do they have an idea for doing it in a different way? Lead students to understand that sometimes, repeating instructions can be combined together. This is a loop!	



Lessons 8 – Loops

In this lesson, students will solve challenges using loops. By end of this lesson, students will complete challenges 3-6.

Encourage students to use the loop block in order to receive 3 stars.



U.S. Standards Addressed

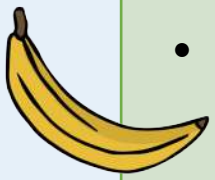
CSTA-K12 Computer Science Standards	
• 1A-AP-10 • 1A-AP-12	• 1A-AP-14 • 1A-AP-15

Part 1: 5 Minutes

Present

Discussion	**5 mins.**

- In the previous lesson we introduced the concept of **loops**. Start this lesson by asking the students the following:
  - Do you remember our previous lesson? We learned a new concept, do you remember what it was?
  - If students don't remember loops, use leading questions such as "what happens when we need to perform the same action many times?" or "In real life, is there a limit to the number of times we need to repeat an action?"
  - If students do remember loops, you can solidify what they already know by going over loops again and explaining why they are so important (for example, loops are helpful in executing one or more statements a certain number of times).
- Define:
  - A loop is a way to instruct computers to repeat a set of actions or in other words, doing the same action multiple times.
  - Until Loop - Until loops are a way to instruct computers to repeat a set of actions until a certain event occurs. For example, keep walking until you reach the end, keep working until you finish your homework or keep playing until your team scores 10 points.
  - Loops are a fundamental part of programming - it is an essential part of any programming language and as becoming a programmer you will master the concept of controlled repetition.



Part 2: 25 Minutes
Playtime-1

Instructions	**5 mins.**

1. Start by presenting the loop tutorial in challenge #3
2. Present the challenge and ask students what would be the solution - [ML,ML,ML,ML,ML,ML,ML]
3. What is unique about this problem/solution? The same instruction repeats until the monkey gets to the treasure chest
4. Follow the tutorial
    a. Present the new LOOP instruction and drag it to the coding area
    b. Show students how to tap the instruction that will be within the loop (ML) and then drag it into the loop
    c. Play
5. Explain that the instructions within the loop are performed in order of appearance and repeated in the same order until the condition is met

Pay attention to the following:

- Tap/drag to add a loop
- Tap to add an instruction after the loop
- Drag to place it within the loop or change the order

Challenge	**Scope**
1-2	A long-repeated sequence which is part of previous lesson
3	A loop tutorial on how to use the loop block

PART 2: 25 MINUTES
PLAYTIME-2

Instructions	20 mins.
Pay attention to the following: • Tap/drag to add a loop • Tap to add an instruction after the loop • Drag to place it within the loop or change the order	

Challenge	Scope
4	Loop[JR]
5	Loop[JL]
6	Loop[ML]

Part 3: 5 Minutes
Debriefing

Discussion	**5 mins.**

Let's emphasize the power of loops. Open Challenge #5.
Ask the students how they would solve the challenge without the loop block

- We need to use the Jump Left block 7 times (JL-JL-JL-JL-JL-JL-JL)

Now, ask the student how they solve the same challenge using the loop block

- Use Loop block and inside the loop have Jump Left block

Code is short and simple. You did not need to count how many times we need to jump left.

Without a Loop block

With a Loop block

Lessons 9 – Do it again

In this lesson, students will continue to practice using the Loop block.
The students will need to use more than one block inside the loop.

In this chapter, we will see that some challenges can be solved in multiple ways so check with your class to see what creative solutions they came up with.

By end of this lesson, students will complete challenges 7-10.

U.S. Standards Addressed

CSTA-K12 Computer Science Standards	
• 1A-AP-10 • 1A-AP-12	• 1A-AP-14 • 1A-AP-15

Part 1: 15 Minutes
Present

Review	**5 mins.**

- Ask your students what loops are useful for, why we use them and when.
- Explain the fact that we use loops in cases when we need to repeat an action many times.
- Loops are one of the most important computing structures.
- The main challenge is identifying which actions should be repeated. Once defined, you can build the code accordingly.
- It is important to understand that the entire code within the loop is performed at each loop repetition. The order of the instructions within the loop doesn't change.

Part 1: 15 Minutes
Present

Game	**5 mins.**
The game will emphasize that we can have more than one block inside the loop. It will also demonstrate how the sequence of blocks are executed. Line up a few students – between 5 to 7. Ask for a volunteer, tell them to stand in front of the first student, shake their hand and move right/left to get to the next student. Repeat this for all students lined up.	

Part 1: 15 Minutes
Present

Explain	**5 mins.**
Ask the students to write the actions taken here (let's assume there were 5 students lined up) and write the code on the board. Ask the students to describe what is repeated here (shake hands and move right/left). Now ask the student to suggest a code that uses a loop. If you have time, you can repeat the game – this time perform 3 actions each time. For example, shake hands, clap and move right/left.	

Code without a loop
Shake hands Move right/left Shake hands Move right/left Shake hands Move right/left Shake hands Move right/left Shake hands Move right/left

Code with a loop

Loop:
  Shake hands
  Move right/left

Part 2: 15 Minutes
Playtime

Instructions	**20 mins.**
Pay attention to the following: • Tap/drag to add a loop • Tap to add an instruction after the loop • Drag to place it within the loop or change the order • Mention that these challenges require move than 1 block inside the loop	

Challenge	**Scope**
7	Loop[JU,ML]
8	Loop[JR] or Loop[MR,JU]
9	Loop{MR,JU,MR] **Note**: Loop[MR,JU] also solve the challenge, but then the monkey bumps, so it is better to use the longer code
10	Loop[JR,JU,MR]

Part 3: 5 Minutes
Debriefing

Discussion	**5 mins.**

Ask the students how many times will the block inside the loop be executed?

- Depending on how far the treasure chest is

For example, open challenge #4. If you solve the challenge using a loop and one Jump Right block, how many times will the code be repeated? (Answer: 6 times)

Now, if you place two Jump Right blocks inside the loop, how many times will the code be repeated? (Answer: 3 times).
You can run the code and let the students count how many times the arrow points to the first block in the loop.

Repeated 6 times

Repeated 3 times

Lessons 10 – careless endless

By end of this lesson, students will complete challenges 11-15 and will complete this chapter and course.

U.S. Standards Addressed

CSTA-K12 Computer Science Standards	
• 1A-AP-10 • 1A-AP-12	• 1A-AP-14 • 1A-AP-15

Part 1: 5 Minutes
Present

Review	**5 mins.**

Explain again that in programming there are different types of loops. For example, a loop that is repeated for a predefined time.
Ask the students when does the loop in these challenges ends?

- When the monkey reaches the treasure chest.

What will happen if the monkey does not reach the treasure?

- Nothing will stop the loop!
- In programming, it is very important to make sure your loop ends.

Open challenge #9. Write the following code:

- Loop[MR]

Before running the code, ask your students what they think will happen?

- The monkey will continue moving right

Run the code and show the students that the monkey passes the Treasure chest and continues moving. Draw their attention to the hand that points to the reset button.

Hand points to press the reset button

Part 2: 25 Minutes
Playtime

Instructions	25 mins.

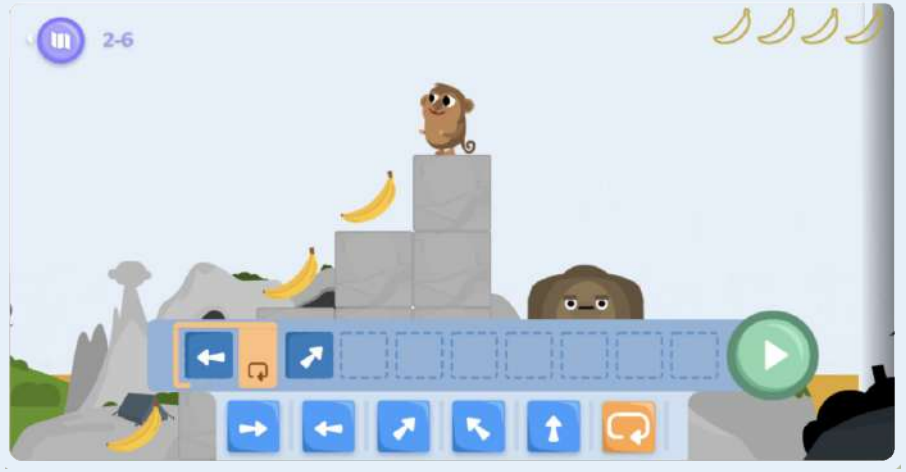
Pay attention to the following:

- Tap/drag to add a loop
- Tap to add an instruction after the loop
- Drag to place it within the loop or change the order
- Mention that these challenges require more than 1 block inside the loop
- Make sure to collect all bananas too
- In these challenges, there might be 3 or 4 blocks within the loop

Challenge	Scope
11	Loop[JL,JL,ML,ML]
12	Loop[ML,ML,JU]
13	Loop[JU,ML,ML]
14	Loop[JR,MR,MR]
15	Loop[JR,JU,MR,MR]

Part 3: 5 Minutes
Debriefing

Discussion	**5 mins.**
The loop here ends when the monkey reaches the treasure chest. Any code that is added after the loop will not run, because the challenge is solved. If you add code before the loop, it will run once since it is not part of the loop. We will practice this in the Advanced Loops chapter. For now, you can show the students that a code after the loop is not reached. Open challenge #6. Write the following code: Loop[ML], JR (see blocks below). Run the challenge and see that the monkey does not jump right.	



Glossary

Term	Definition
Computer	A **computer** is a machine that is able to take information (input), do some work on or make changes to the information, to make new information (output). Modern computers are very different from early computers. They are now very powerful machines that are able to do billions of calculations every second. Most people have used a personal computer in their home or at work. Computers are useful for many different jobs where automatic tasks are useful. Some examples are controlling traffic lights, vehicle computers, security systems, Washing machines and Digital Televisions. (source: https://wiki.kidzsearch.com/wiki/Computer)
Software	Computer software (often called just software) is made of one or more computer programs. Sometimes it means one specific program, or it can mean all the software on a computer, including the applications and the operating system. Applications are programs that do a specific thing, such as a game or a word processor. (source: https://wiki.kidzsearch.com/wiki/Computer_software)
Hardware	Hardware (computer hardware) includes the physical parts of a computer, such as the cabinet, motherboard, central processing unit, storage, and more. Peripheral hardware include - monitor, keyboard, speakers, mouse and more. In some computers the peripheral devices are built-in (like laptop or tablet)
Until loop	Until loops are a way to repeat a set of actions until a certain condition is satisfied

Great Job!

You have Completed

CodeMonkey Jr.

Sequencing & loops