Computational Thinking Guide

by sphero

Computational Thinking Guide by Sphero

For information about this title or to order other books and/or electronic media, contact the publisher:
Sphero, Inc
education@sphero.com
www.sphero.com

Table of Contents

Computational Thinking Guide 4

Overview 4

Definition 4

Computer Science vs CT 5

The Four Pillars of Computational Thinking 6

Decomposition 7

Examples of Decomposition. 8

Unplugged Activities to Develop Decomposition 9

Pattern recognition 10

Pattern Recognition Examples 11

Unplugged activity ideas to develop Pattern Recognition 12

Abstraction 14

Abstraction Examples 15

Unplugged activity ideas to develop abstraction 16

Algorithmic Design 18

Algorithmic Design Examples 19

Unplugged activity ideas to develop Algorithmic Design. 20

BOLT Blocks 1-8 Computational Thinking Activity Cards 22

BOLT Blocks 1 Computational Thinking 23

BOLT Blocks 2 Computational Thinking 24

BOLT Blocks 3 Computational Thinking 25

BOLT Blocks 4 Computational Thinking 26

BOLT Blocks 5 Computational Thinking 27

BOLT Blocks 6 Computational Thinking 28

BOLT Blocks 7 Computational Thinking 29

BOLT Blocks 8 Computational Thinking 30

Computational Thinking Guide

Welcome to Sphero's Computational Thinking Guide. Computational Thinking (CT) provides a systematic set of processes for thinking through complex problems, whether they involve computers or not.

This guide will:

- define computational thinking
- discuss the overlap between computer science and CT

Overview

Computational Thinking (CT) is a term first referenced by Seymour Papert in 1996, as "procedural thinking," pertaining to the logic and procedural flow of computer languages. Although the definition of CT varies from researcher to researcher as well as between the different academic subjects, it defines the relationship between a problem, its solution, and the structuring of the data. By combining critical thinking and problem-solving with computing power, CT provides the foundation for innovating solutions to real-life problems that can be expressed in terms a computer can apply.

Definition

Computational thinking is a problem-solving process that includes (but is not limited to) the following characteristics:

Abbreviated: Computational thinking is thinking or problem solving like a comput by er scientist. ~*EdSurge*.com

Detailed: Computational thinking is a problem-solving process that includes (but is not limited to) the following characteristics:

- Formulating problems in a way that enables us to use a computer and other tools to help solve them
- Logically organizing and analyzing data
- Representing data through abstractions such as models and simulations
- Automating solutions through algorithmic thinking (a series of ordered steps)
- Identifying, analyzing, and implementing possible solutions with the goal of achieving the most efficient and effective combination of steps and resources

~CSTA & ISTE

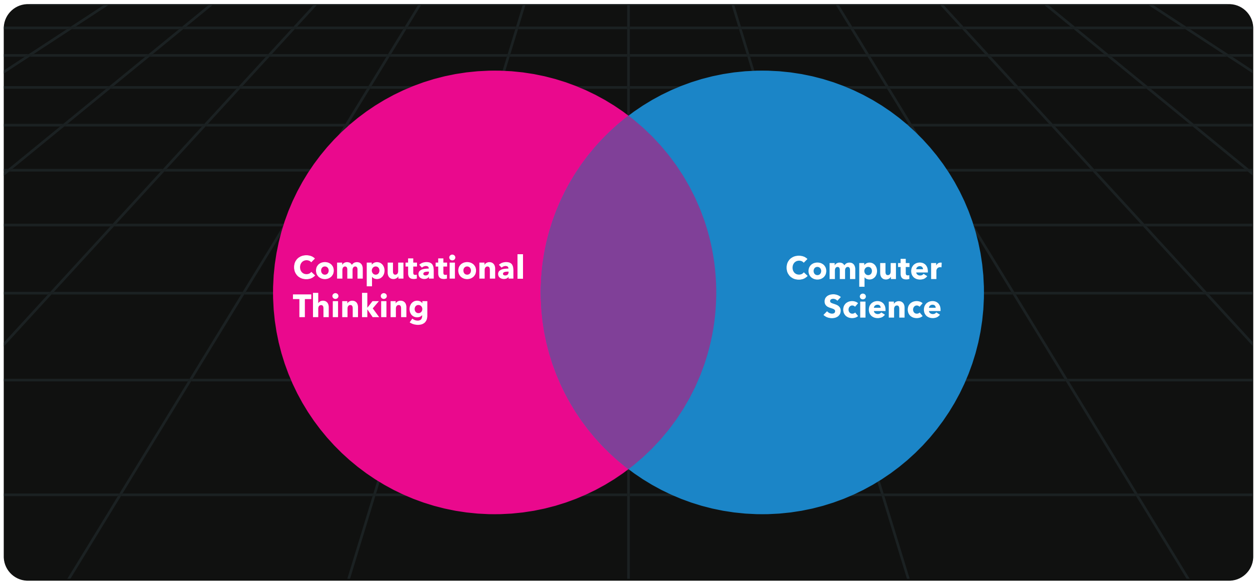
Consider the following ways CT benefits students in computer science, other subject areas, and their everyday lives.

Computational thinking can help students…

- **solve problems related to computers** by creating a flowchart to help them think through how they want their robot to respond to light levels in the room before writing the program.
- **solve problems in the world around them** by using computers to aggregate, analyze, and visualize trends in weather data in their region.
- **address problems unrelated to computing** by using CT to methodically think through and develop a solution to a problem like food waste in their school cafeteria.

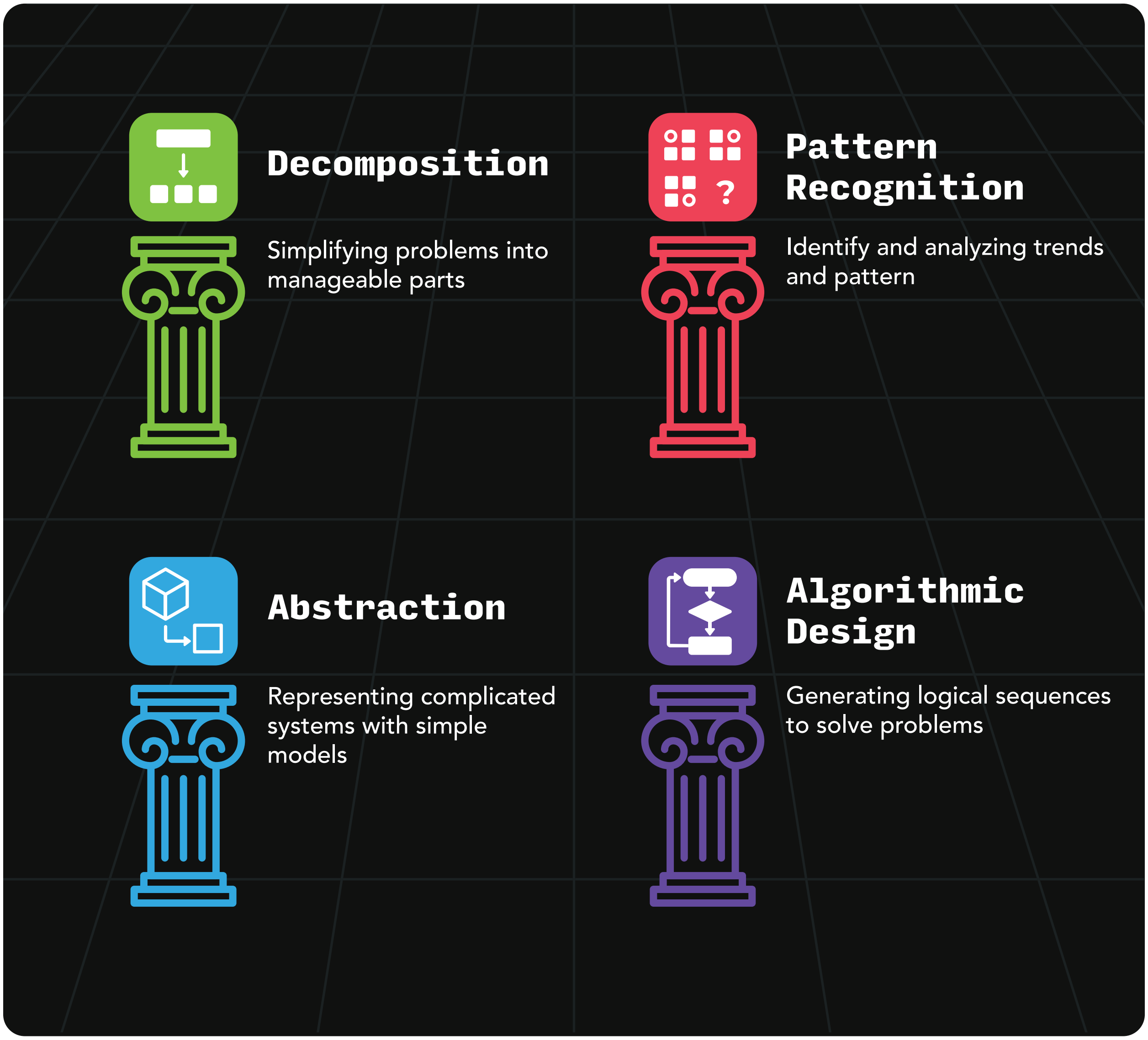
Computer Science vs CT

Computational thinking and computer science are often used interchangeably, but there are some key differences. The benefits of CT skills gained by tackling challenging problems with computers or robots expand beyond computer science alone and into every aspect of everyday lives.



The Four Pillars of Computational Thinking

CT is often broken down into four processes or pillars.



Decomposition

Simplifying problems into manageable parts

Pattern Recognition

Identify and analyzing trends and pattern

Abstraction

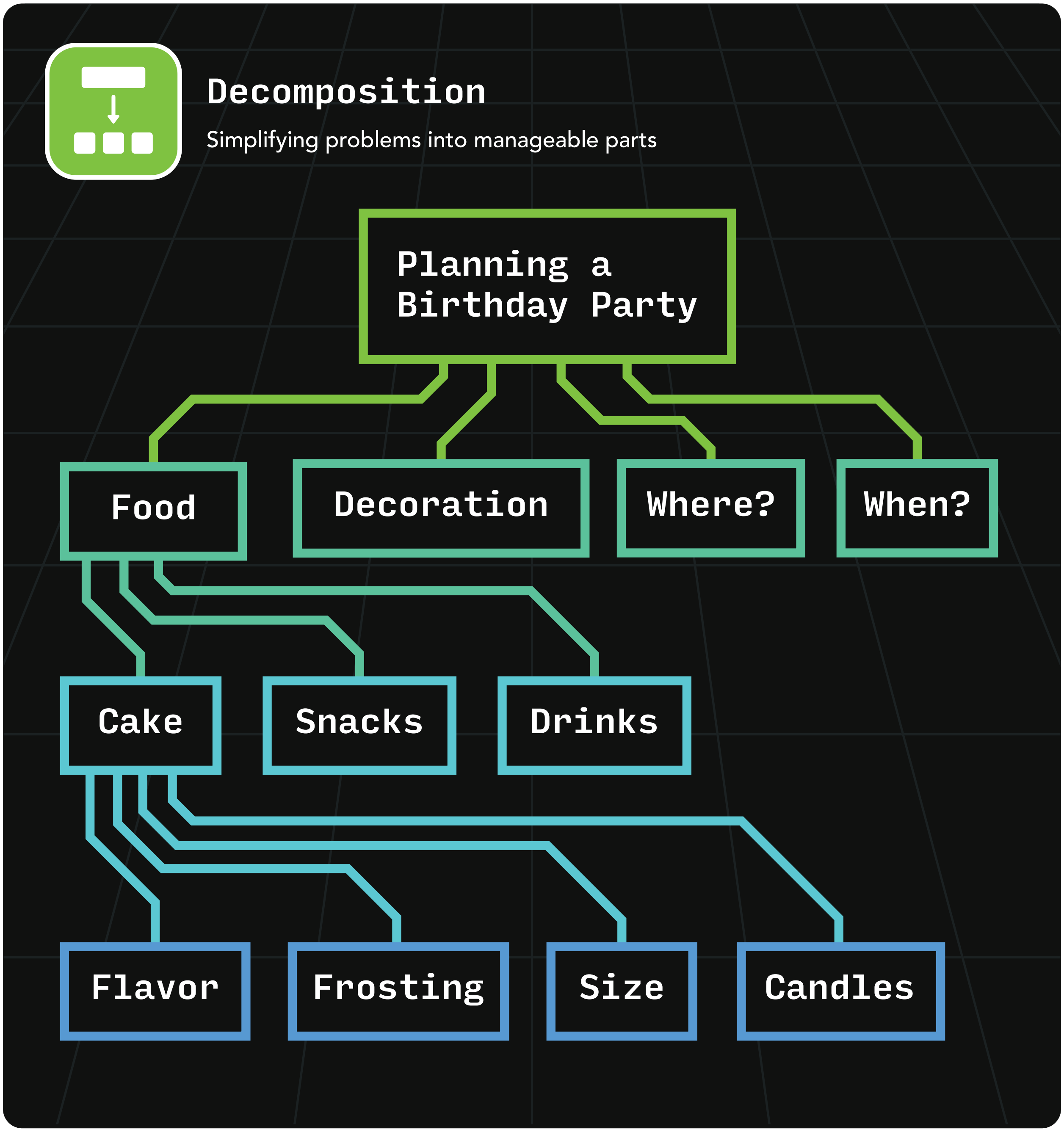
Representing complicated systems with simple models

Algorithmic Design

Generating logical sequences to solve problems

Decomposition

Decomposition is the breaking down of something, like a complex problem, into small manageable parts. Decomposition sets the context and helps us fully understand a process or problem before working with it or developing a solution.



Decomposition

Simplifying problems into manageable parts

- Planning a Birthday Party
  - Food
    - Cake
      - Flavor
      - Frosting
      - Size
      - Candles
    - Snacks
    - Drinks
  - Decoration
  - Where?
  - When?

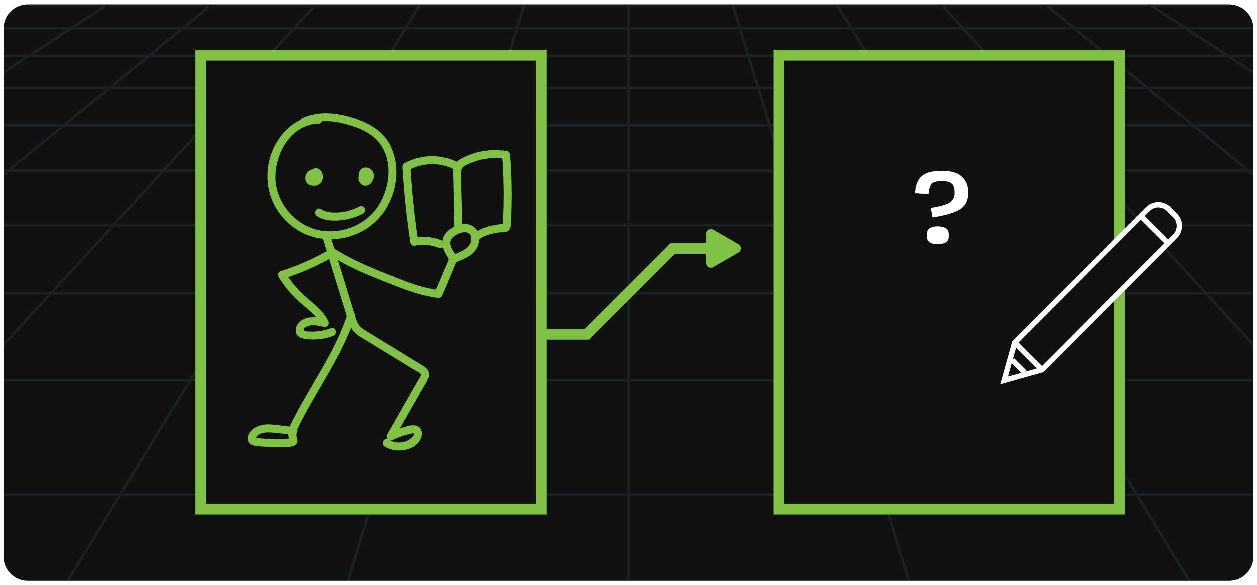
Examples of Decomposition

	Elementary School	Middle School	High School
Everyday Life	Breaking down the steps to learn how to tie your shoes.	Football coach individually explaining coverage assignments for a defensive play or scheme.	Planning a trip including: cost, travel, packing, destination, activities, hotel, food, etc…
School	**Language Arts** Listening to a story and breaking it down into the setting, characters, plot, and resolution.	**Arts** Determining and assign the necessary roles to put on the school talent show.	**Social Studies** Analyzing a historical event or period, such as the civil rights movement or the onset of WW2, by breaking it down into political, sociocultural, economic, and geographical factors.
STEM Career	**Archaeologist** Mapping out an archaeological dig into sections to organize, excavate, and record their findings one square at a time.	**User Experience Designer** Enhancing user experience for a product based on user feedback, personas, prototypes, and business constraints.	**Environmental Engineer** Designing a new wastewater treatment facility that supports local water needs, adheres to regulations, manages land usage effectively, and fits within budget constraints.

Unplugged Activities to Develop Decomposition

#1 Stick Figure Instructions

Start by drawing a very simple stick figure. It could be reading a book, walking to the bus stop, or playing volleyball. Then redraw the stick figure paying close attention to each step. As you draw, list each step in the process. Find a partner and keep your stick figure a secret. Read your list to your partner and have them redraw your picture. How close to the original is the copy? How good are your instructions?



#2 Invent a Handshake

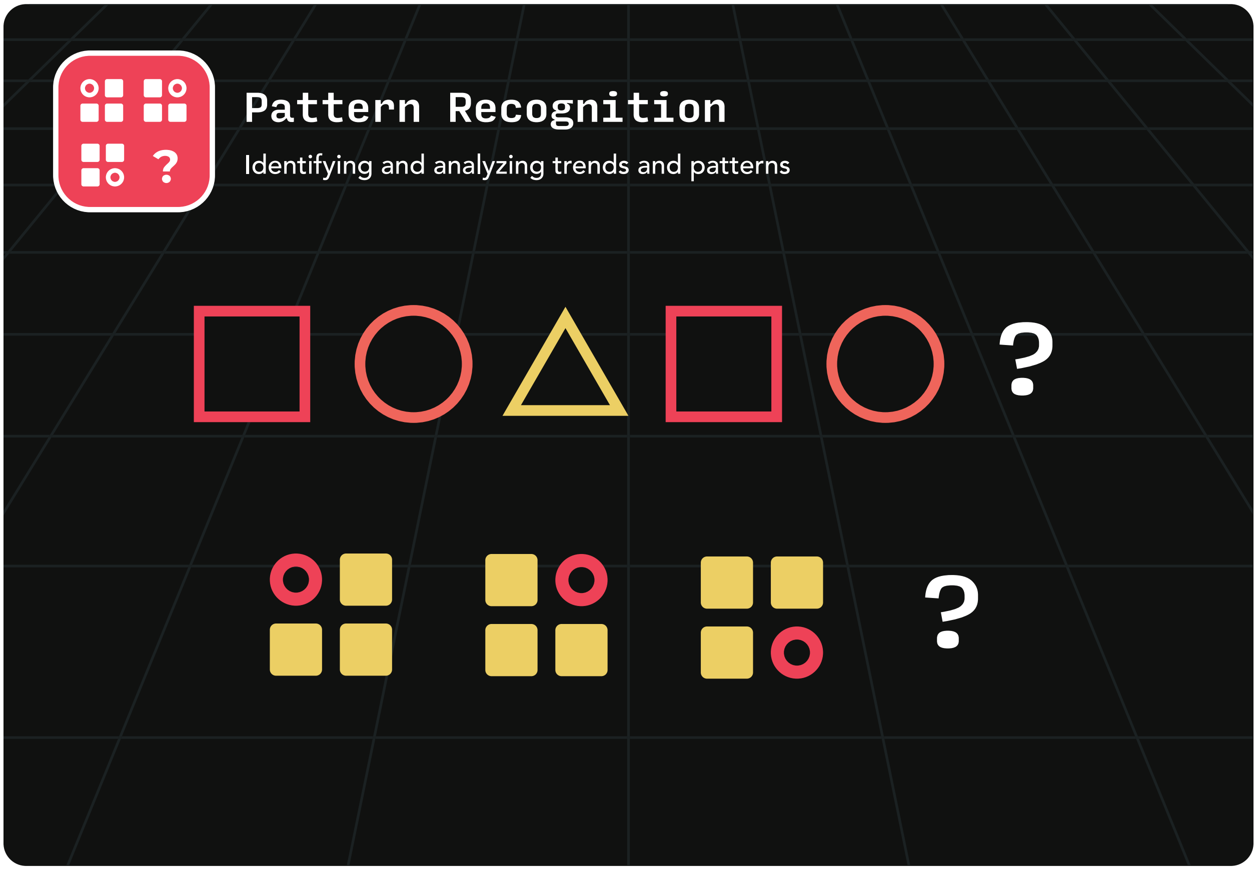
Complex actions are made up of small parts. Find a partner and brainstorm a new handshake. It might include a traditional handshake, but it also might include fist bumps, claps, snaps, or other movements. The only rules are that it must have at least three separate actions and you and your partner must be able to remember it! After you work out your handshake, make a written list of each action in the sequence. Give it to another group and see if they can reproduce your handshake accurately. You've just decomposed one great handshake!

#3 Completing a task

Things we do every day without really thinking about it would take some pretty complex rules for a robot to do successfully. Choose something that you do often like brushing your teeth, going for a walk, or making a sandwich, and write down each step in the process. A partner or group will pretend to be robots and follow each of your steps to try to complete the task. When it's your turn to be the robot, be sure to follow the steps literally: if a step says "brush your teeth" how do you know what teeth are? Or where they are? Or what "brush" means?! Remember: robots and computers are not nearly as smart as you are and they need to be told everything.

Pattern recognition

Pattern recognition helps us identify patterns that exist in the world all around us. Once they've been identified, they can often be applied to make predictions or formulate solutions in similar situations.



Pattern Recognition

Identifying and analyzing trends and patterns

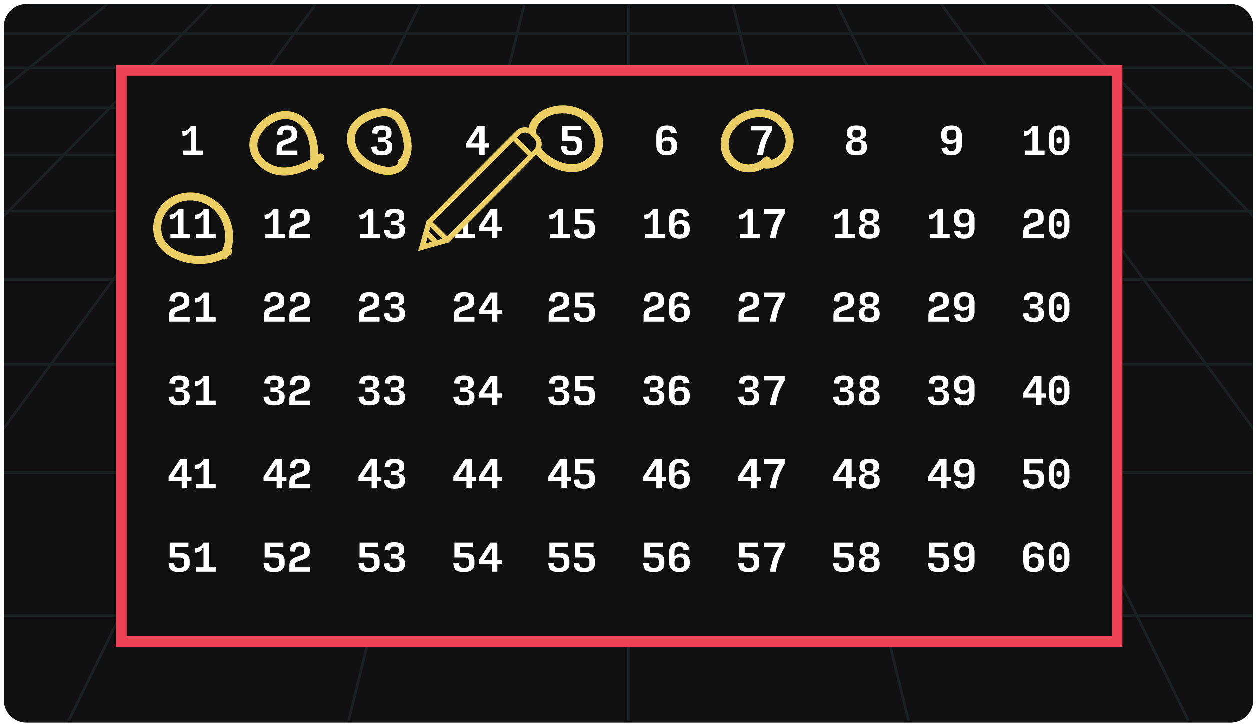
Pattern Recognition Examples

	Elementary School	Middle School	High School
Everyday Life	Using the sun as a guide to tell us when it is time to get up and to get ready for bed.	Cell phones relying on pattern recognition in order to unlock using facial recognition.	Identifying motifs or repeating series of notes within a musical arrangement to not only learn the piece faster but also understand how it evokes emotion in the listener.
School	**Language Arts** Students singing songs with rhyme and beat patterns to learn the names each month in the year.	**Math** Learning how to calculate the perimeter, area, and volume of various shapes.	**Science** Learning how DNA base pairs align with each other, combine in sets of three to make codons, and how repeating sets of codons code for amino acids in protein synthesis.
STEM Career	**Animal Nutritionist** Weighing and distributing the food necessary to keep animals in the zoo healthy.	**Physical Therapists** Using exercises and treatments in repeated patterns to help others rehabilitate.	**Financial Analyst** Applying market research techniques to analyze stock market trends.

Unplugged activity ideas to develop Pattern Recognition

#1 Prime Numbers Patterns

List out numbers 1 through 60 in rows of ten. Circle all of the prime numbers. Remember prime numbers are numbers that have only two factors, the number one and the number itself. The first five have been circled for you.



1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60

Look for patterns in the numbers you circled. Think about:

- Can you make predictions about which numbers will or won't be prime?
- Can you identify steps to check whether a number is prime?
- Will prime numbers be more or less common the higher you count?

#2 Image Recognition

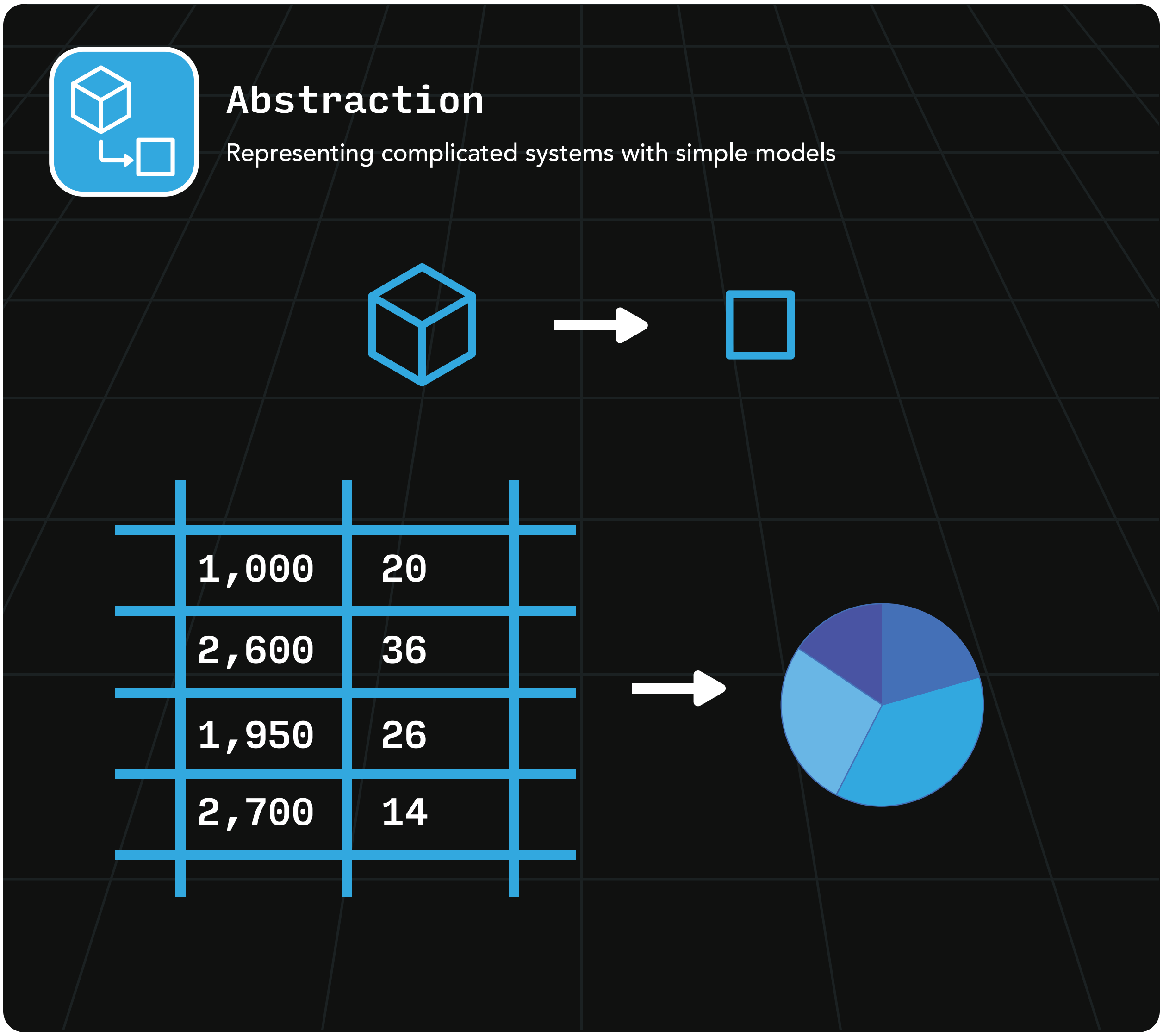
Gather images of two different living organisms, say cats and dogs. Look at your collection of images closely. What features are the same? Which features are different? Computers can use a form of artificial intelligence called machine learning to begin to identify objects in a picture. The more pictures they analyze, the better they become at identifying objects. Think like a computer and make lists of characteristics you would use to distinguish between your two living organisms. Find some new pictures and see if your lists work!

#3 Patterns everywhere!

Where else in your life can you notice patterns? They're all around us and it just takes a bit of thinking to notice them. Are there patterns in the music you listen to or tv shows you watch? The sports you play? What you do on Saturdays? How might your Saturday patterns be different or similar to days when you go to school or are on vacation? Being able to identify differences among patterns is just as important as being able to identify the patterns themselves. Come up with a few patterns you notice in your life and discuss them with a partner or group. How are the patterns you noticed similar or different from the patterns your partner or group came up with?

Abstraction

Abstraction is the process of identifying the most important information and ignoring irrelevant details. By focusing on what's most important we can create models that allow us to see through the noise and create solutions.



Abstraction

Representing complicated systems with simple models

1,000	20
2,600	36
1,950	26
2,700	14

Abstraction Examples

	Elementary School	Middle School	High School
Everyday Life	Planning a trip to the grocery store by making a list of items you need to buy.	Creating music playlists for different occasions.	When flying, you only look for the gate for your flight, ignoring the rest of the scheduled flights.
School	**Language Arts** Reading or listening to a story and identifying the main characters and/or the main idea.	**Math** Using a variable such as "x" in order to represent a generic property or value to be solved for in a mathematical equation.	**Science** Representing an atom using a model such as the Bohr or Quantum Mechanical model, which leaves out complex details and focuses on the subatomic particles and their relationship to each other.
STEM Career	**Pilots** To prepare for a flight, pilots checking the weather reports in the areas they will be flying through.	**Veterinarians** Reviewing blood panel reports to diagnose health issues.	**Architect** Considering a client's needs and budget when designing a house.

Unplugged activity ideas to develop abstraction

#1 Abstract Your Day

Think about a recent day that you can remember clearly. Make a list of everything you did that day from the time you woke up until the time you went to bed.

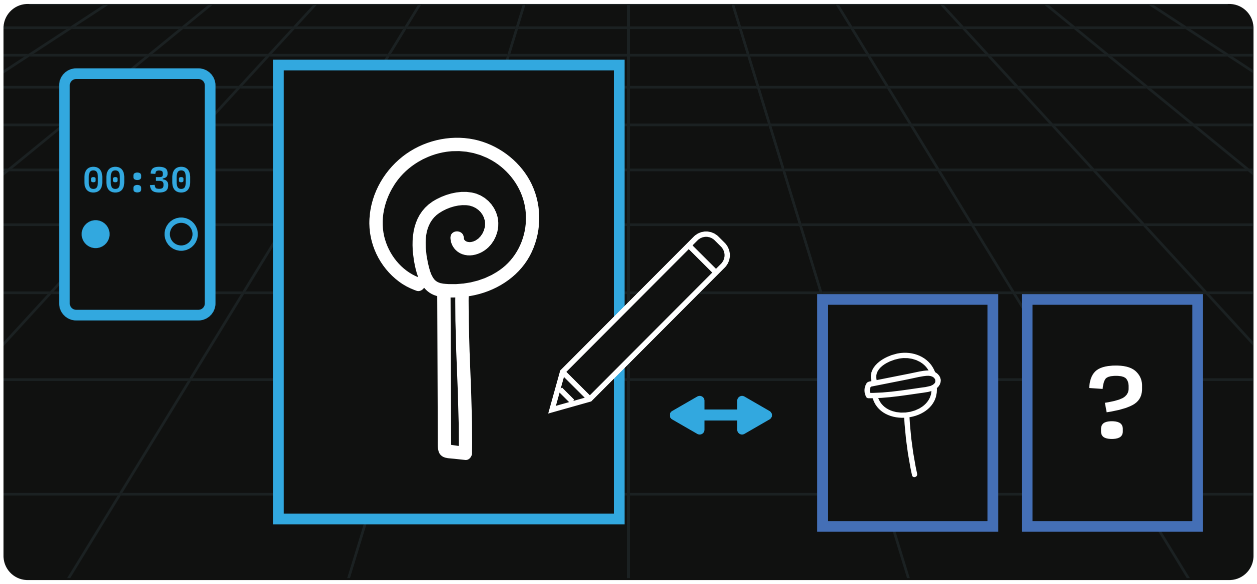
If your list begins with…

- Opened eyes.
- Yawned.
- Sat up in bed.
- Removed covers.
- Got out of bed.

…it's going to be a really long list! Limit yourself to only 10 items. What would make the list? What would you skip? How did you decide what would make the list and what wouldn't?

#2 Picture Perfect

When you draw a picture, you take something from the real world or something from your mind and make a representation of it. No picture is an exact representation, you are adding the most important details to make it recognizable. This is an abstraction. Get a piece of paper, a pencil, and a timer. Take 30 seconds to draw each of the following words: truck, lollipop, airplane, pineapple, and T-shirt. Which features of each item were most important? What details would you add if you had more time for each drawing? Compare and contrast your pictures with a partner's pictures. How are they the same and different?



#3 Songs, Movies, and Books

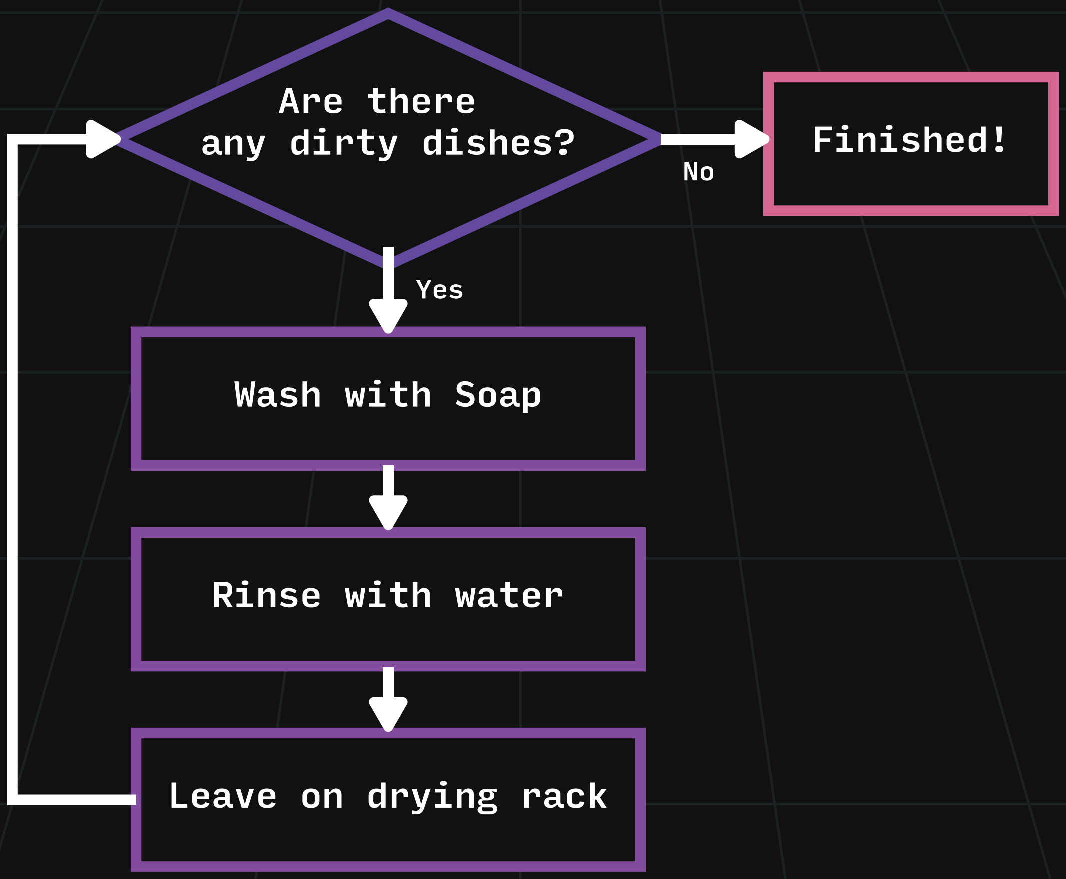
Think about your favorite song, movie, or book. If you want to tell someone why you like it, you probably wouldn't sing them the whole song or read them the whole book, you would pick out the important parts and explain why you like them. That's an abstraction! You're taking a lot of information and creating a simple "model" to share with other people. Try it out: think of your favorite song, movie, book, or place you like to go and try sharing the most important parts with a partner or in a group to get them interested in whatever it is you chose. Sometimes thinking about the most important parts is hard! If you're not sure which parts to keep and which to get rid of, try asking a partner or the group which parts they thought were the most important.

Algorithmic Design

Algorithmic Design is the development of steps we can use, often over and over, to solve a problem. Algorithms, whether designed for humans or computers, are a clear set of sequenced instructions.

Algorithmic Design

Generating logical sequence to solve problems



Algorithmic Design Examples

	Elementary School	Middle School	High School
Everyday Life	Learning how to follow a recipe in order to cook a meal or bake a dessert.	Understanding how your interaction with social media networks influences the content that they put in your feed.	Analyzing and prioritizing the factors that influence postsecondary educational or career opportunities.
School	**Math** Learning the "regrouping" method when adding together numbers greater than 10.	**Science** Creating a dichotomous key to identify an organism's family and species based on its observable traits.	**Foreign Language** Students practicing their vocabulary acquisition and speaking skills by explaining how to order dinner at a restaurant.
STEM Career	**Nurse** Needing to follow specific safety procedures (directions) when working with their patients.	**Project Managers** Planning and designing the steps that need to be completed and the order they should be completed.	**Video Game Designer** Building if/then algorithms to program the game to predictively respond to user inputs.

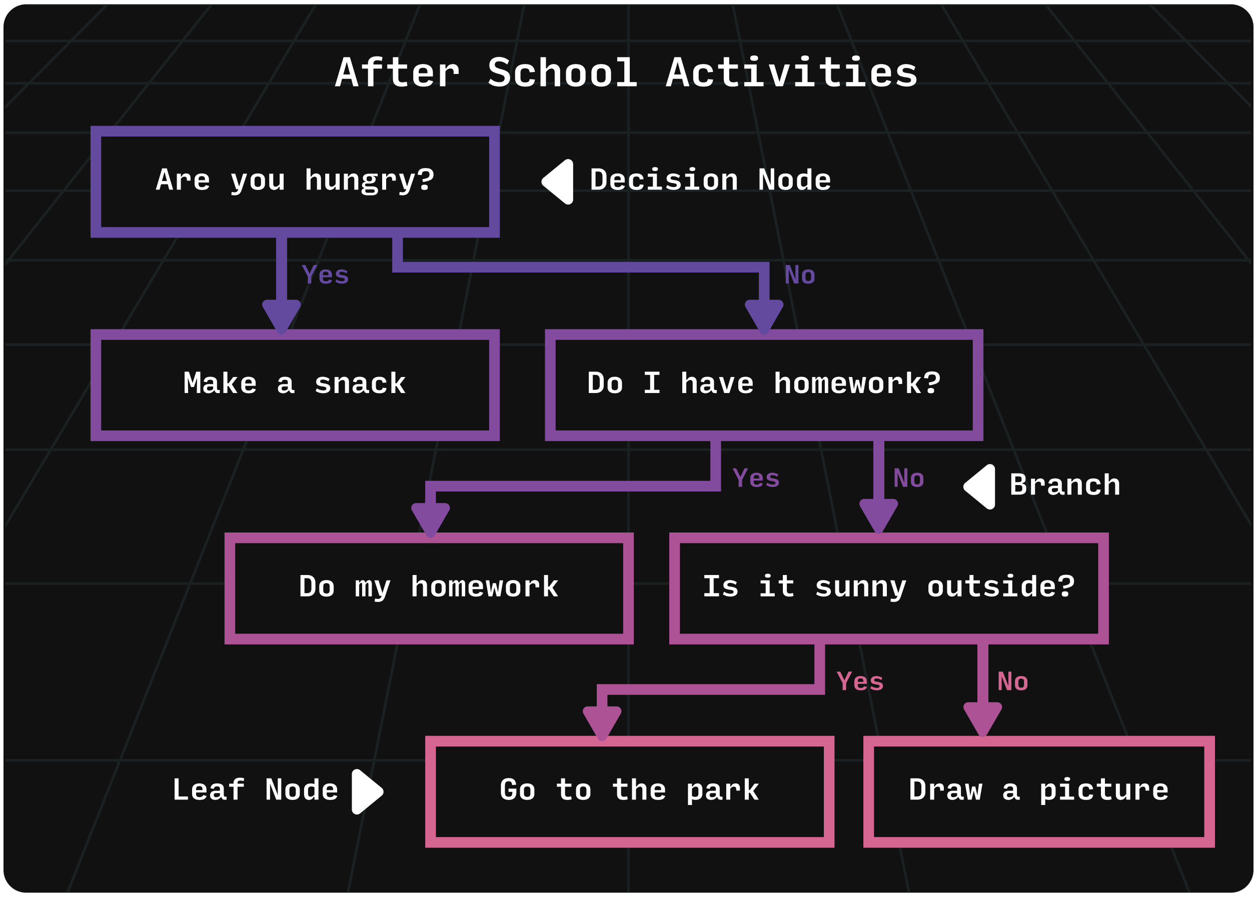
Unplugged activity ideas to develop Algorithmic Design

#1 Decision Trees

Examine the decision tree below. A decision tree shows all the possible outcome for a questions and has three components:

- The decision nodes are the questions: "Do I have homework?"
- The branches are the possible outcomes: "Yes" or "No"
- The leaf nodes are the end results: "Do my homework."

Computers work in binary–transistors are either on or off–so most decision trees work with yes or no questions.



Make your own decision tree to help you navigate a decision you make often in your life. Once you have a decision tree, you can program a computer algorithm to help you make the decision!

#2 I'm Thinking of a Number…

Find a partner and play "I'm Thinking of a Number." Partner 1 thinks of a number between 0 and 20. Partner 2 tries to guess the number. If they are incorrect, Partner 1 replies, "higher" or "lower". Partner 2 should keep guessing until they guess the number. Switch roles and play a few more rounds. How many guesses does it take to guess the number? With your partner, design an algorithm–or set of steps–to help you play the game. What number should you guess first? What number should you guess second? What algorithm will minimize the number of guesses? Test your algorithm to see if it works.

Now, change the game so Partner 1 can think of any number between 0 and 100. Does your algorithm still work? What would you need to change?

#3 Eyebrow Communication

Eyebrows don't just prevent moisture and dust from getting in your eyes, they help you communicate. Your brain already has this algorithm written into your subconscious. It turns out that even apes and other primates use their eyebrows in a similar fashion. Do some backwards thinking to better understand this algorithm. Think about how you move your eyebrows when you are:

- angry
- surprised
- confused
- interested
- scared
- acknowledging a friend

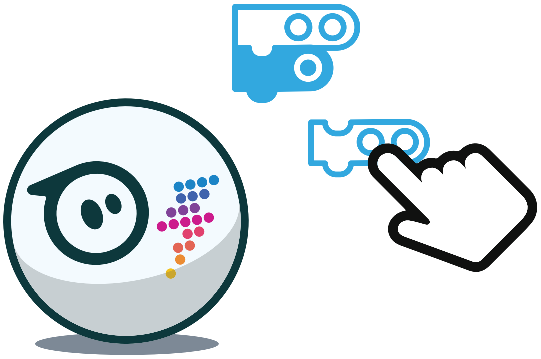
Make a cheat sheet with the emotions and the eyebrow actions then use it to convey a feeling to a partner using only your eyebrows.

BOLT Blocks 1-8 Computational Thinking

Activity Cards

If you have Sphero BOLTs in the classroom, you and your students are practicing your CT skills all the time. You just may not be thinking about it.

The following eight activity cards are developed to be used with the BOLT Blocks 1-8 Sphero Edu activities. Print them out and share them with your students. Read through the activity cards with students. Some will help your students notice how they used CT while programming BOLT. Others will challenge them to extend their learning after completing the activity.

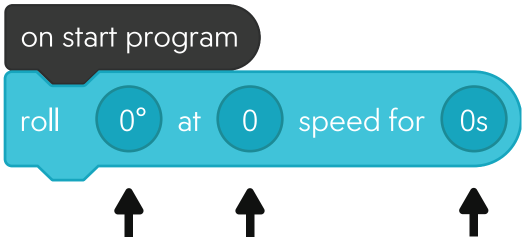


Activities

BOLT Blocks 1 Computational Thinking

Decomposition

Change the inputs to see what they control.



```
on start program
roll 0° at 0 speed for 0s
```

Pattern Recognition

Which color category contains blocks that can make BOLT start, stop, and spin?

Abstraction

Modify your program to make a shape other than a square.

Algorithmic Design

Build a new program that creates 3 different polygons.

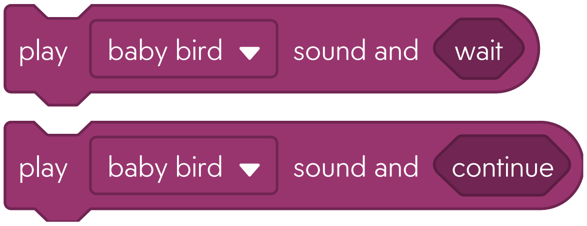
Activities

BOLT Blocks 2 Computational Thinking

Decomposition

Using the **play sound block**, run your program using **wait** and then **continue**.

What's the difference?



play baby bird sound and wait

play baby bird sound and continue

Pattern Recognition

Name at least three block categories that you will use repeatedly for each scene in your story?

Abstraction

Using what you know, explore other block options you have in the light, sound, and movement categories that could enhance your story.

Which blocks would you like to use?

Algorithmic Design

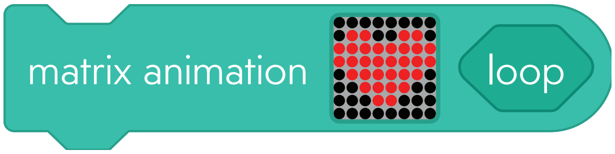
Program your story one scene at a time. Run your program to make sure it matches your plan.

Activities

BOLT Blocks 3 Computational Thinking

Decomposition

Play one animation and guess how many frames it took to create. Click on the image to find out if your guess is correct.



Pattern Recognition

Loop forever and **loop x time blocks** repeat the blocks inside of them. Add one block inside of a loop block and press Start.

What happens? What happens if you add more blocks?

Abstraction

What other movements do we see in the world? Which one could you animate on BOLT's matrix? A bird's wings? A person walking?

Algorithmic Design

Make your movement idea into a matrix animation.

Are there any spots that are jumpy and unnatural? Make them look smooth and realistic.

Activities

BOLT Blocks 4 Computational Thinking

Decomposition

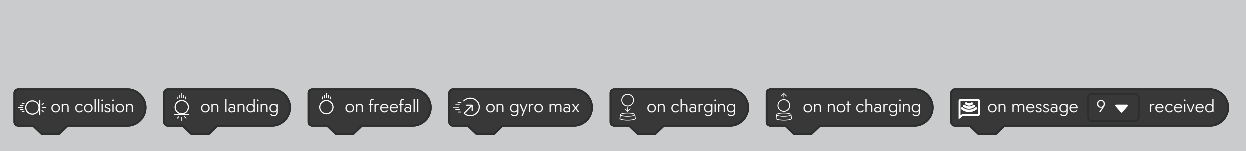
Create an ordered list of steps/actions to draw a simple picture. Trade lists with someone else. Follow each others directions, only using the steps on the list. What happened?

Pattern Recognition

What should BOLT do when it collides with an object? Program a pattern that includes a change in movement, light, and sound.

Abstraction

Take a look at the other blocks in the Event category. Do any of the actions sound familiar? Choose one and add it to your program. What will any attached blocks be executed?



Algorithmic Design

Change the order or swap out blocks in your Human Pong Game in a way that changes what happens, but the game still works.

Activities

BOLT Blocks 5 Computational Thinking

Decomposition

Play the Animal Toss game with 3 - 5 people. One person observes the game writing down the steps that BOLT is performing. Take turns until everyone has written their own list.

Pattern Recognition

Compare lists that you created in the Decomposition activity with your group members.

Do you see any patterns?

Abstraction

Think about how your favorite computer game works. How did the programmer use the **if then block** when creating the game? Complete this sentence: If I _____, then _______.

Algorithmic Design

Take a look at the program you created in Blocks 5. With a partner, discuss what you could change to make BOLT play the animal sound after the ball is caught? Try it? Did it work? Why or why not? Make changes and try again.

Activities

BOLT Blocks 6 Computational Thinking

Decomposition

If you clean your room before 2 PM and you help wash the dishes, you can have a friend over. If it is hot outside, you and your friend can go swimming. What needs to happen for you and your friend to go swimming? How did you decide to breakdown the problem?

Pattern Recognition

Write a funny if/then story following the pattern of beginning your next if statement with the then statement in the previous sentence.

For example: If I were at home, then I would make a sandwich. If I made a sandwich, then ....

Abstraction

Using pseudocode, plan what sounds you will code to your pitch, yaw, and roll movements. Make sure to indent each if statement to demonstrate its nested structure.

Algorithmic Design

Create algorithms using the **if then block** and the **if then else block**. How are they similar? How are they different?

Activities

BOLT Blocks 7 Computational Thinking

Decomposition

While running the Tug 'o War program, compare BOLT's movements to the coding sequence. What strategy will help you win?

Pattern Recognition

What patterns can you identify between lux values and areas of your classroom? Which areas have the highest luminosity? The lowest?

Abstraction

Knowing how BOLT's light sensor works, design another game that uses its sensor values.

Algorithmic Design

How can the program be adapted for younger students to play? How can it be adapted to make it more challenging?

Activities

BOLT Blocks 8 Computational Thinking

Decomposition

Examine the code in step 5. Which conditions needs to be met for the program to exit the **loop until block**?

(Hint: **tossNumber** is set to 0 and **expireNumber** is set to any number between 3 and 10.)

Pattern Recognition

Initially the **tossNumber** variable is set to 0 and is never programmed to change. Using an **operator block**, program **tossNumber** to increase by 1 every time BOLT is caught. See step 6 for extra help.

Abstraction

Experiment with the **matrix animation block**. Can you make a better abstraction of a BOLTato on the BOLT LED matrix?

Algorithmic Design

How can your recreate the algorithm for this game to use the **on collision event block** instead of the accelerometer sensor?

Computational Thinking Guide

by sphero