

Program Toolkit

Junior: Coding Basics Badge

Curious about computer science but unsure where to begin? Start with earning your Coding Basics Badge! You'll dive into the fundamentals of coding through fun, interactive activities (all without a device).

Time: 2-3 hours

Steps Overview:

- Learn about algorithms
- What are conditionals?
- Efficiency trail mix activity
- Loops handshake activity
- Women in STEM

Materials Needed:

- Terms Glossary

Activity #1: Learn about algorithms

- Paper and pencils OR whiteboards and whiteboard markers

Activity #2: What are conditionals?

- Conditional printouts
- Music to dance to (optional)

Activity #3: Efficiency trail mix

- Recipe printouts
- 5 trail mix ingredients (recipe attached but ingredients can be switched for alternatives)
- Plastic bags or bowls
- Measuring cups/spoons

Activity #4: Loops handshake

- Pencils
- Loop handshake template

Activity #5: Women/Careers in STEM

- Video or book about a woman computer scientist OR a guest speaker who works in computer science
 - This activity kit uses the book *Margaret and the Moon* (about Margaret Hamilton) as an example.
 - Copy of story is not provided, but a book can be loaned from a library, or you can find a read-aloud on YouTube.
- Profile cards, colors (optional)

Introduction

Today, we're going to learn about coding and how computer scientists use coding for good! When you think of a computer, you probably think of a laptop, tablet, desktop computer, or smartphone.

Did you know that there are computers in other machines that you use every day? Computers are machines that are programmed to do things automatically. What else do you think could be a computer? (Answer: Cars, some washing machines, phones, TVs, etc.)

What sorts of problems do these different computers solve? (Girls may say: Phones help people to communicate with others, a washing machine cleans clothes, etc.) Those are great examples!

Computers and technology help us to solve problems all day long and all over the world! They help us communicate with friends and family more easily. They make it faster and easier to do jobs like delivering mail, manufacturing clothing, and farming. They also help doctors analyze data as they research cures for diseases.

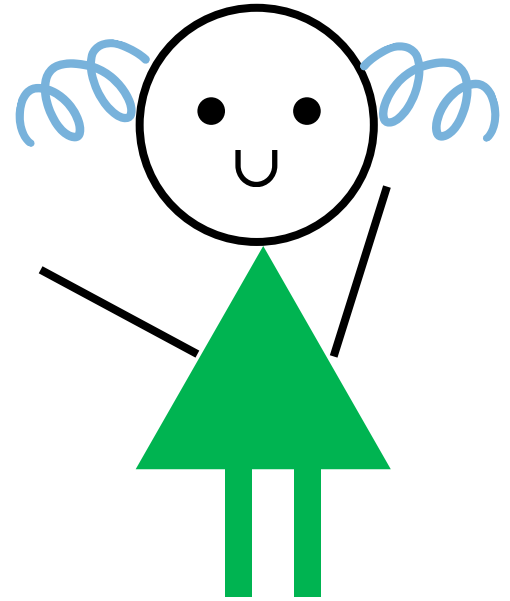
The people who create computers and tell them what to do are called computer scientists. They help people by creating computers to solve problems. Today we're going to learn how they do it!

Step #1: Learn about algorithms- (15 minutes)

- Computers may seem really smart because they can do lots of things for us. But, computers can't actually think for themselves. They can only do things they're told to do.
- Computers don't understand English very well, but they do understand a special language computer scientists use called code.
- For example, a computer scientist can code a list of steps for a robot. The robot follows the steps to do anything from moving through a maze to doing surgery.
- **A list of steps to complete a task is called an algorithm!**
- Computers need clear directions, or algorithms, to work.
- Now you are all going to be computers, and I'm going to give you an algorithm – are you ready?

Activity - Pictionary

- Your one rule is absolutely NO PEEKING at each others'! Also, you cannot ask me questions for clarification. Now, I'm going to give you your algorithm. So just draw what I tell you to.
 - i. Draw a circle in the middle of your whiteboard
 - ii. Under your circle, draw a triangle
 - iii. Draw 2 lines coming out of your triangle
 - iv. Now draw squiggly lines coming out your circle
 - v. In your circle, draw 2 more circles
 - vi. Now in your circle, put 2 dots
 - vii. In your circle, draw a letter U
 - viii. Finally, under your triangle, draw 2 rectangles



Discussion

- What does everyone's pictures look like?
 - Do they look the same? Different? Way different?
- What was easy about this? What was hard?
- The point of this activity is to learn that computers need clear directions to know what to do! This means computer scientists need to write very specific algorithms.

Step #2: What are conditionals- (20 minutes)

- Now we're going to learn about something that can help make your instructions more specific. These are called conditionals.
- **Conditionals are statements that only run under certain conditions or situations.**
- Conditionals are written with IF and ELSE Statements.
 - IF statements only perform actions IF the specified condition is met. When an IF action *isn't* met, the ELSE statement will run.
- You may not know it, but you already use conditional statements every day! Any time you check the weather before you get dressed, you're using conditionals!
- Help me finish this sentence to get dressed for the day: IF the weather is: ____, THEN I should wear my raincoat and boots. What condition could tell me I should wear my raincoat and boots? (Answer: Rainy or snowy).
 - Awesome! So IF the weather is rainy, THEN I should wear my raincoat and boots. ELSE, I can wear my sneakers.

- Can you think of other conditionals you see in everyday activities? Allow for discussion.
 - IF I am tired, THEN I will take a nap. ELSE, I will play a game.
 - IF I miss the bus, THEN I will have to walk to school. ELSE, I will ride the bus to school.
 - IF the cookies are golden brown on top, THEN I will take them out of the oven. ELSE, I will let them keep cooking.
- In programming, computer scientists use conditionals, too!
 - For example, when you're playing a game, a condition may be IF a player's score is greater than 10, THEN the player wins. ELSE, there is no winner yet.
 - Or maybe on a weather app, IF the temperature is 85 degrees, THEN the app will send a notification to “drink water!”. ELSE, do not send a notification.
- Now, let's practice using conditionals to play a game called Iffy Dance!

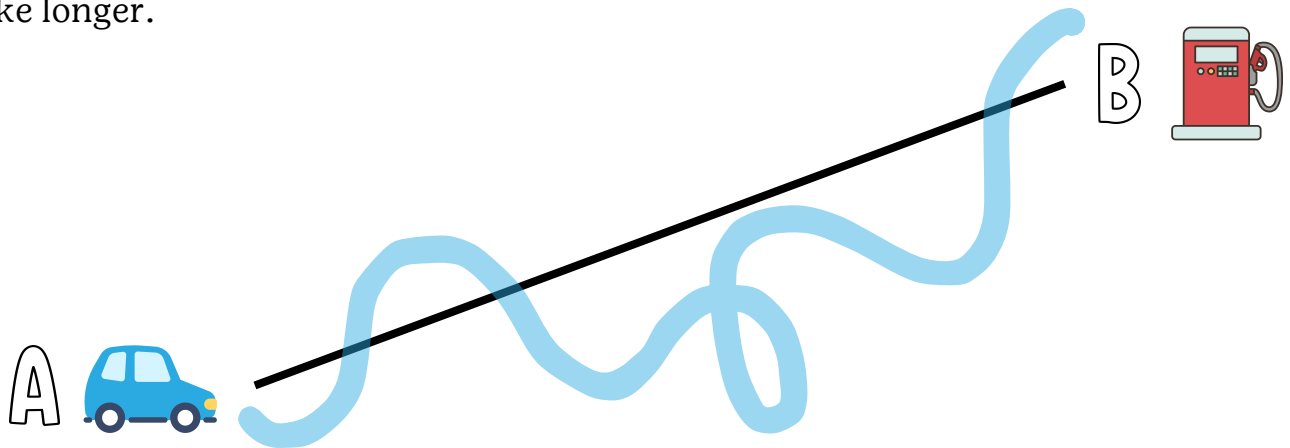
Activity - Iffy Dance

- To play this game, I'm going to call out conditional statements for you to follow a specific dance move. For example:
 - IF you're wearing tennis shoes, THEN wave your hands in the air. ELSE put your hands on your knees
 - IF your hair is up in a ponytail, THEN spin around three times. ELSE spin around one time.
 - IF you're wearing jeans, THEN disco. ELSE do the scuba diver.
- Optional: play music
- Continue calling out some conditional dance moves and then let Girl Scouts be the computer scientist and call out the conditionals using IF/THEN commands.



Step #3: Efficiency Trail Mix - 25 minutes

- Next we're going to talk about efficiency. Does anybody know what efficiency means?
- Here is an example (*can draw figure for easier visualization*). Imagine that my car is running out of gas, and I need to get from point A to the gas station (Point B) before I run out. Which line should I take? Why's that the best line to take?
 - (Answer: The straight line is the shorter route and will take up less gas, so I'll make it to the gas station before I completely run out.)
- The straight line is the most efficient way to get from point A to point B. **Efficient means you do something as fast and easy as possible.** A straight line is more efficient because you go straight to your destination. If you go in a roundabout way, it will take longer.



- We can also practice coming up with efficient ways of doing something in your everyday life.
- Think about washing the dishes. You can clear the table and put the dishes in the dishwasher. Or you can clear the table and put the dishes on the counter, then move the dishes to the dishwasher. Which is more efficient?
 - (Answer: Using one step to put dishes in the dishwasher is more efficient than using two steps.)
- What are some other examples of everyday things you do that are efficient (or not efficient)?
- In computer science, we want to be efficient when we write code.
- Efficient programs are those that respond quickly and don't take up a lot of memory and power.
 - Think about a time where you were sitting at your computer waiting for the loading symbol to finish. You want your computer to respond as quickly as possible without having to wait too long.

Activity - Efficiency Trail Mix

Divide the Girl Scouts into two groups and give each group one of the two recipes from the Trail Mix Recipe Remix handout. One group will be Team Count (they will be counting out the ingredients) and the other will be Team Measure (they will be measuring out the ingredients).

SAY:

- A recipe is a great example of how we can follow an **algorithm** to complete a task (reminder that an algorithm is a list of steps). For this activity, each group will follow their recipe to make the same trail mix.
- However, one group has a recipe with a detailed algorithm to make their mix. The other group has a recipe with an efficient algorithm to make their mix.
- With your team, work together and follow the steps of the recipe exactly as they are written to make trail mix.

Give groups 10 minutes to complete the activity, mixing their trail mix according to their recipe.

When they finish, bring the troop back together and have a discussion about their experience with each recipe.

Discussion

- Let's start by having each group share the directions for each step of the recipe (i.e. Have Team Count read step 1 from their recipe, then have Team Measure read step 1 from their recipe.)
- Team Count, your algorithm had lots of details!
 - How did that help you? (Girl Scouts may say: I always knew what ingredients I should use in each step of the recipe.)
 - How did all those details frustrate you? (Girl Scouts may say: It felt like the instructions kept repeating themselves, it made us move more slowly through the steps, etc.)
- Team Measure, what about you? Your algorithm was efficient.
 - How did that help you? (Girl Scouts may say: We could easily read and follow the directions in each step even though it didn't explain everything in detail.)
 - How did it frustrate you? (Girl Scouts may say: It isn't clear how big the spoon should be when it says a spoonful. What if we were each given different sized spoons?)

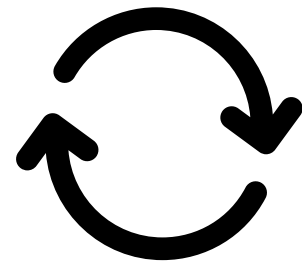
- Which recipe algorithm would you prefer to follow, Team Count or Team Measure? Why? (Girl Scouts may say: Team Count because I need more directions when cooking, Team Measure because I like to follow short instructions, etc.)
- When computer scientists write code, they have to find a balance between efficiency and being specific enough so that the computer knows what to do.
 - For example, for Team Count, your process took longer, but each batch of trail mix is exactly the same (e.g. everyone should have 10 chocolate chips). You were given very detailed instructions.
 - For Team Measure, your process may have taken less time, but everyone's batch of trail mix is going to be slightly different (e.g. some people's trail mix might have 10 chocolate chips and others' might have 15 chocolate chips), and sometimes that could be a problem if the coding project you are working on requires you to be very accurate.

Step #4: Loops Handshake - 25 minutes

Let's learn about a way we can make things more efficient.

Have scouts stand up and follow your instructions.

- SAY: Spin around.
 - Wait for scouts to spin around.
- SAY: Spin around again.
 - Wait for scouts to spin around again.
- SAY: Spin around again.
 - Wait for scouts to spin around again.
- SAY: If I want you to spin around three times, I could say, "Spin around. Spin around again. Spin around again." But, what would be an easier way to tell you to do that? (Answer: Spin around 3 times!)
 - That's right! I could just tell you to spin around 3 times.
- In computer science, programmers use something called **loops to make the steps of their algorithms repeat over and over again.**
- And guess what, we use loops in our everyday lives! Instead of telling you wash and fold each piece of laundry separately, I can tell you to wash, dry, and fold all your clothes.
 - That's an example of how a loop can make the directions shorter to give and easier to understand.



- What are some other everyday activities you do that are repeated over and over?
 - Examples: eating soup (taking a bite over and over again), jumping rope (skipping over the rope over and over again), walking (taking steps over and over again)
- Loops are also very important to computer scientists. They make code shorter, easier to write, and easier for a computer to understand.

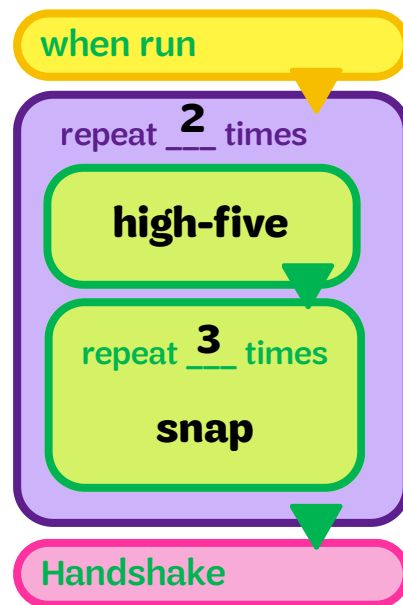
Activity: Loop Handshake

- Has anyone ever made a secret handshake with a friend? Did you know that the handshake may have had a loop in it?
- We are going to practice making BFF (best friends forever) handshakes that use loops!
- Let's first take a look at the Looping Handshake Example and see if we can figure out how to do it correctly. *Show the Looping Handshake Example.*
 - Hmm, notice how this green shape says "repeat 2 times." I wonder what that means? (Answer: It means we should repeat the actions inside of it 2 times.)
 - Would anyone like to come up to the front and see if they can do our example handshake with a friend?
 - Invite two Juniors to demonstrate the Looping Handshake Example for the troop.
 - The handshake should look like: high five, fist bump, low five, high five, fist bump, low five, handshake. Does anyone have any questions?
- Now let's see if you can make your own BFF handshakes.
- First, pair up and spend a few minutes deciding on the different steps for your BFF handshake.
- Your handshake needs to use at least one loop to repeat at least one action.

Divide into pairs or groups of 3 and give time to work on their handshakes.

- It looks like you all came up with some pretty neat handshakes. Now let's take turns showing them off to the rest of the troop!
- Ask each pair to perform their handshake in front of the group. Make sure each pair gets a chance to show off their handshake.

- Great job! Let's jazz up our handshakes even more!
- To add more motions, we can put loops *inside* of *other* loops. **These are called nested loops.**
- For example, if we wanted to repeat high-five, snap, snap, snap twice we would want to use a loop for the three snaps and another loop to repeat the whole thing twice, like this:



- Now, see how crazy you can make your handshake! Add in at least one nested loop.

Give scouts some time to create a handshake with a nested loop.

- Does anyone want to share their remixed handshake algorithm with a nested loop?
- For Girl Scouts watching: could you write the algorithm for the handshake just by watching their demonstration?

Step #5: Women/Careers in STEM - 20-45 minutes

Lastly, we are going to learn about a real computer scientist!

- Introduce who Girl Scouts will be learning about. This can be done through watching a video, reading a book, or even having a live guest speaker who works in computer science. [This example uses the computer scientist Margaret Hamilton.]
- Her name is Margaret Hamilton. In real life, the first men who landed on the moon almost didn't make it. But, a programmer named Margaret Hamilton saved the Apollo 11 mission!

- When she was in her 30's, Margaret led teams of scientists at NASA to create the code that guided the Apollo moon landings. She created software to track the speed and location of the space capsule.
- Three minutes before landing Apollo 11 on the moon, one of the astronauts turned on a radar system that overloaded the computer. This made it impossible for the computer to complete its tasks and land the space capsule on the moon.
- Margaret's program saved the day as it reviewed the system's tasks and accomplished them in order of what was most important. This caused the system not to overload and alerted the astronauts that something was wrong.
- We will learn more about Margaret Hamilton by reading the book (or watching the book read aloud on YouTube) *Margaret and the Moon*.
 - While the book is being read, take note if you notice anything that we learned today!



Conclusion

Congratulations! You have all earned your Coding Basics badge!

(optional) Have scouts complete a profile card listing out what they're curious about, a problem they want to solve, and their favorite technology. Complete the card with a username and profile avatar. Have scouts share their cards.

What did you learn today? Do you remember the new terms we talked about? (algorithm, efficiency, loops, conditionals) Who is interested in learning more about coding?

If you are interested in continuing to learn about coding, check out Girl Scouts' partnership with CodeSpark! (<https://codespark.com/girl-scouts>)

[Supporting Materials on following pages](#)

Coding Basics Glossary

Algorithm - a list of steps to solve a problem or complete a task. For example, when you follow the steps of a recipe, you're using an algorithm.

Code - the language that computer scientists create and use to tell computers what to do. Beginning programmers usually start coding with block-based coding (visual blocks that link together) to build their algorithms. As they progress, they'll move into using written coding languages such as JavaScript or Python.

Computer - any device that stores information and follows instructions from a program. For example, a laptop is a computer; so is a coffee maker that can be programmed.

Conditionals - statements that only run under certain conditions or situations. Conditionals are written with if/else statements (see definitions below)

Efficient Programs - programs that respond more quickly and take less memory and power.

Else Statement - when an IF action isn't met, the ELSE action will run.

Event - an action that causes something to happen

If Statements - statements/actions that occur only IF the specified condition is met

Loop - a set of instructions that is repeated over and over again.

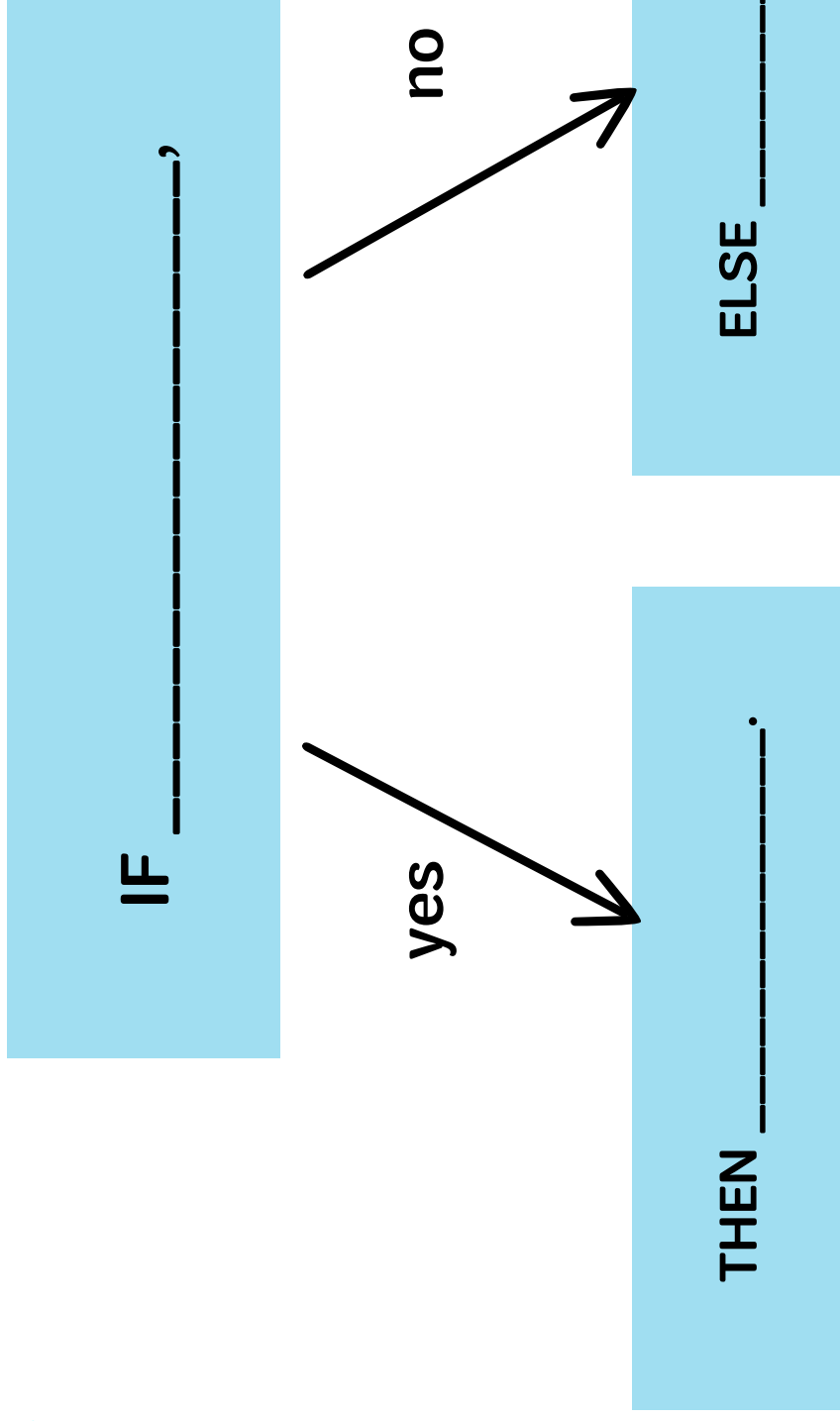
Nested Loop - a loop within a loop, an inner loop within the body of an outer one. The first pass of the outer loop triggers the inner loop. Then the second pass of the outer loop triggers the inner loop again. This repeats until the outer loop finishes.

Perseverance - the ability to work through challenges. For example, when anyone starts to learn coding, they will find some activities challenging. When they keep trying until they succeed, they've demonstrated perseverance.

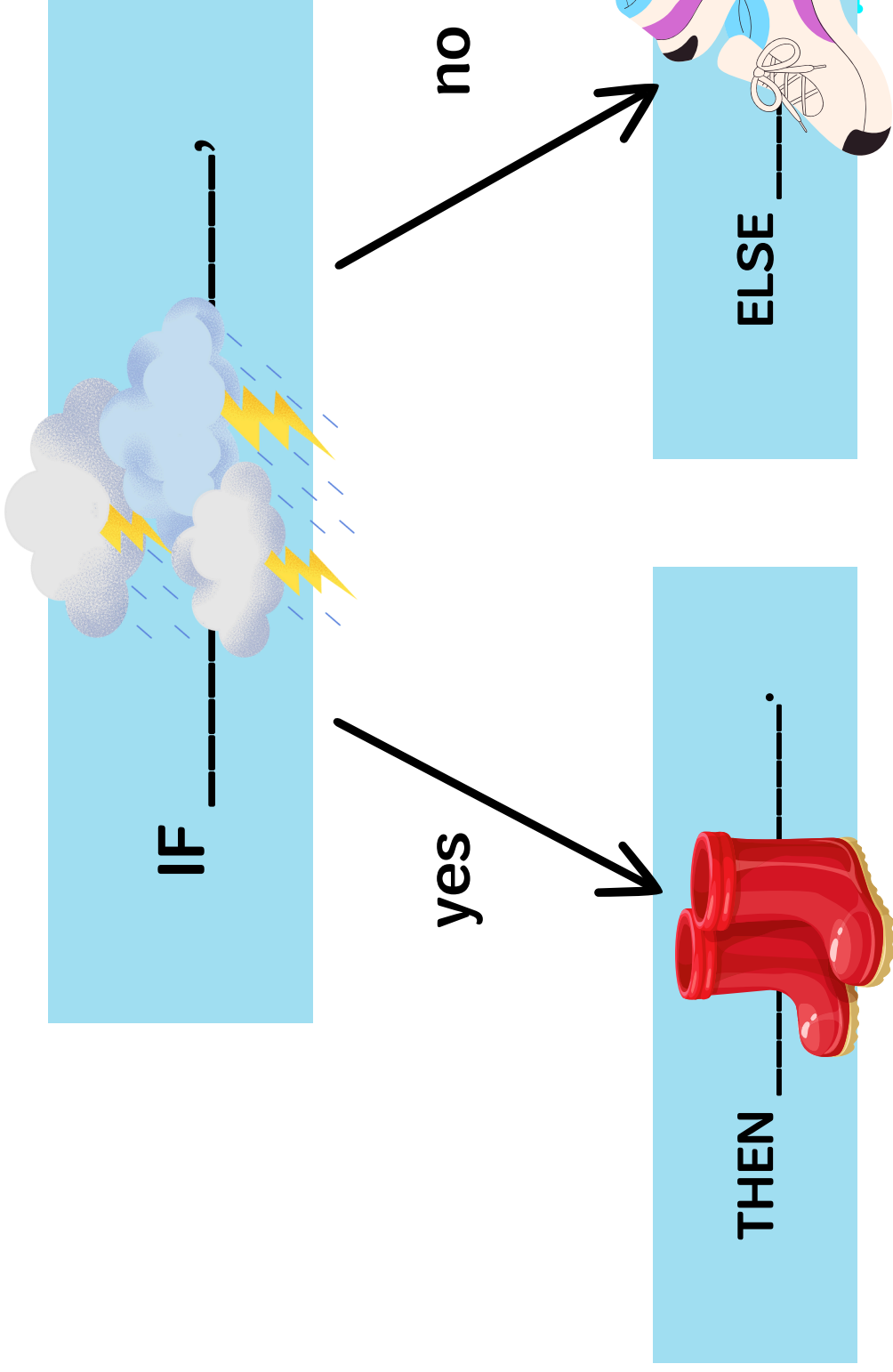
Programming - when a person creates a sequence of instructions, or an algorithm, that makes a computer or machine do something.

Sequence - the step-by-step order in which instructions or events should happen. For example: Think of the steps you do to get ready in the morning. You wouldn't get dressed before getting in the shower!

Conditional Statements!



Conditional Statements!



Trail Mix Recipe

Ingredients

- Chocolate chips
- Chex cereal
- Dried cranberries
- Pumpkin seeds
- White chocolate chips



Materials

- Ziploc bags
- Gloves

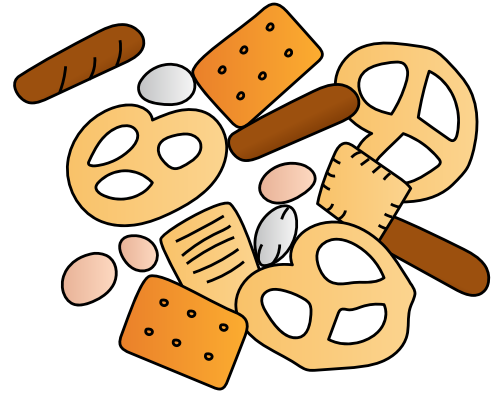
Instructions

1. Count out 10 chocolate chips and put them in your bag.
2. Count out 20 Chex cereal pieces and put them in your bag.
3. Count out 20 dried cranberries and put them in your bag.
4. Count out 15 pumpkin seeds and put them in your bag.
5. Count out 10 white chocolate chips and put them in your bag.
6. Close your ziploc bag tight!
7. Gently shake your bag so all of the items get mixed together.

Trail Mix Recipe

Ingredients

- 1 spoonful chocolate chips
- 1 cup full of Chex cereal
- 1 spoonful dried cranberries
- 1 spoonful of pumpkin seeds
- 1 spoonful of white chocolate chips



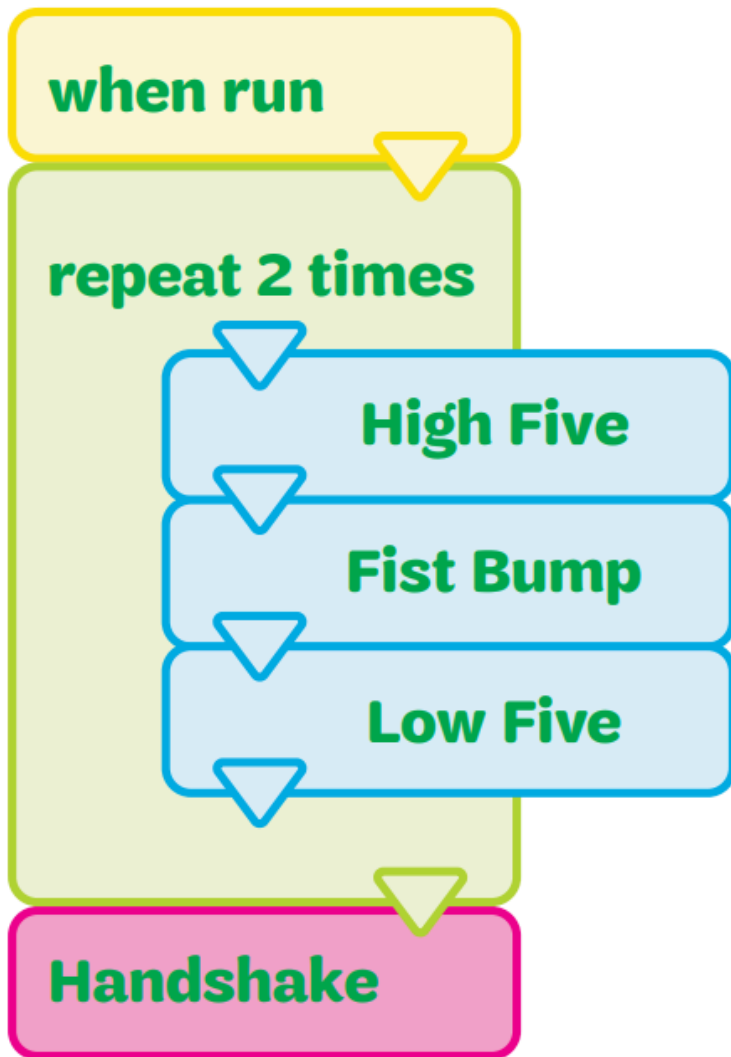
Materials

- Ziploc bags
- Gloves
- Spoon
- Cup

Instructions

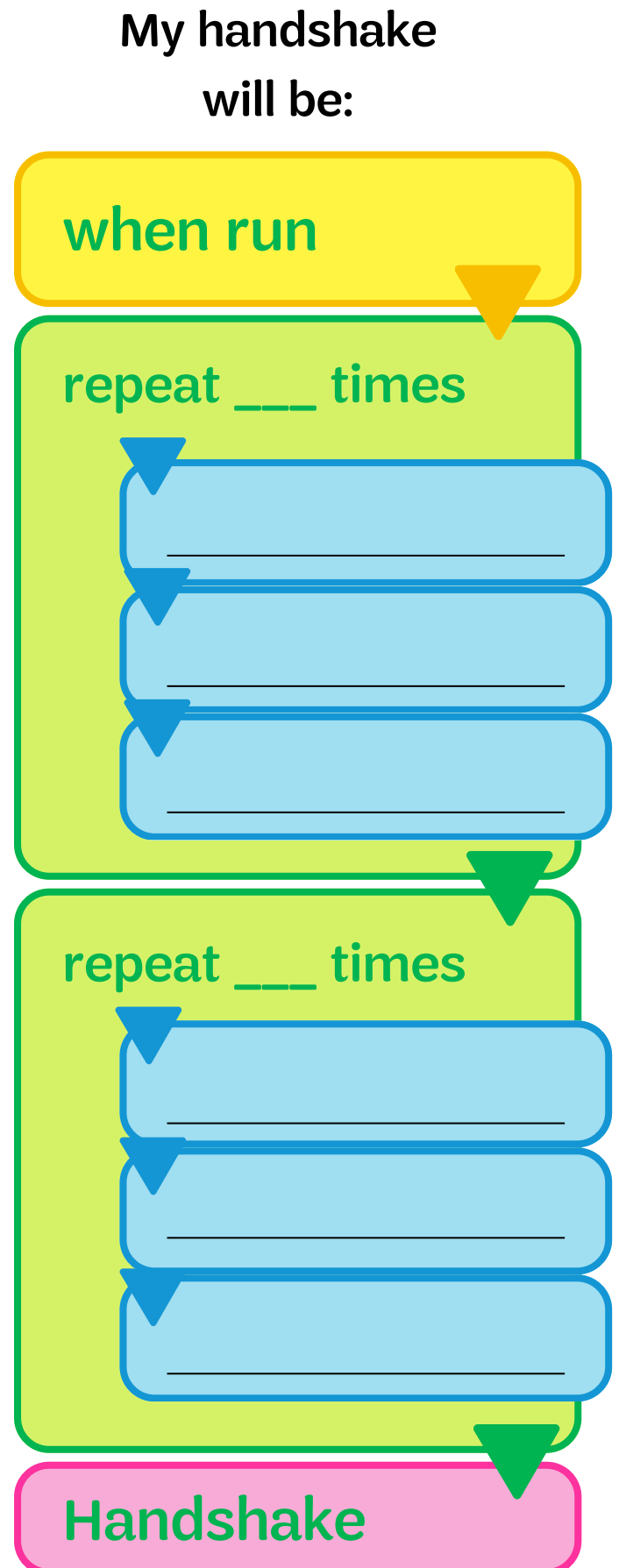
1. Put 1 spoonful of chocolate chips in your bag.
2. Put 1 cup full of Chex cereal in your bag.
3. Put 1 spoonful of dried cranberries in your bag.
4. Put 1 spoonful of pumpkin seeds in your bag.
5. Lastly, put 1 spoonful of white chocolate chips in your bag.
6. Close your ziploc bag tight!
7. Gently shake your bag so all of the items get mixed together.

Looping Handshake Example



This handshake
will look like:

High five
Fist bump
Low five
High five
Fist bump
Low five



Nested Loop Handshake

when run

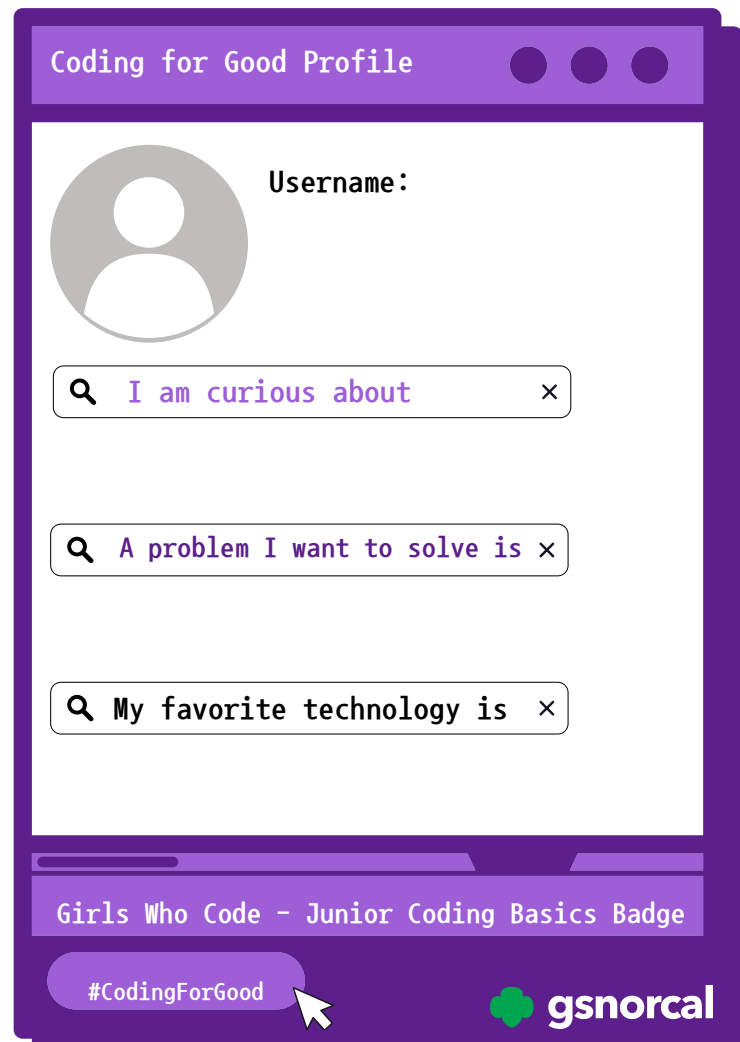
This handshake
will look like:

repeat ___ times

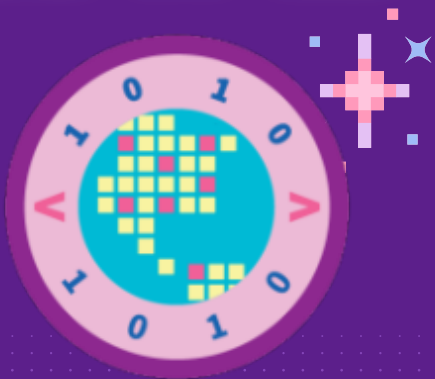
repeat ___ times

repeat ___ times

Handshake



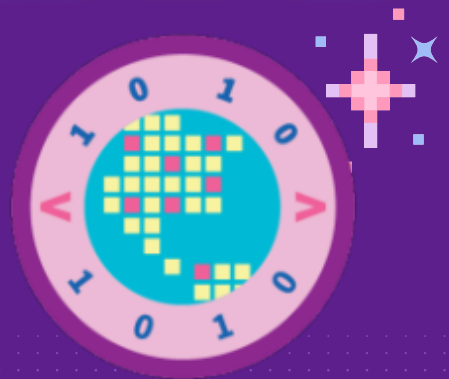
Girls Who Code



Junior Coding Basics Badge



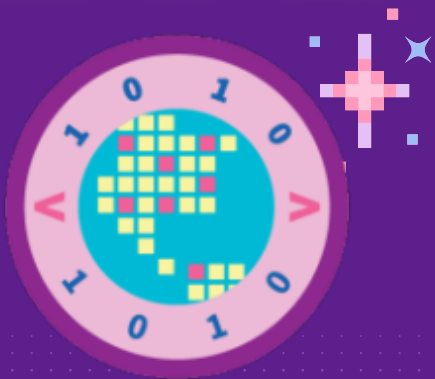
Girls Who Code



Junior Coding Basics Badge



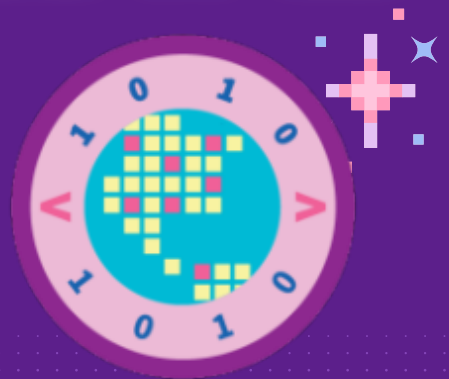
Girls Who Code



Junior Coding Basics Badge



Girls Who Code



Junior Coding Basics Badge

