Senior Coding for Good

Learn how computer programmers write code and how games and apps can help individuals and communities make positive changes.

Badge 1:
Coding Basics

Badge 2:
Digital Game Design

Badge 3:
App Development

This booklet gives girls an overview of the badge requirements and badge steps for all three Senior Coding for Good badges. It also includes interesting background information to spark girls' interest in coding. Volunteers can access the Volunteer Toolkit (VTK) to find complete meeting plans, including detailed activity instructions and handouts.

Welcome to the world of computer coding!

When you've earned these three badges, you'll know how computer programmers write code to make computers perform tasks and develop games and apps to teach people and mobilize communities.

- You'll know how computers make decisions.
- You'll know how game makers create choices for players.
- You'll know how data can make apps useful.

Volunteers can access the Volunteer Toolkit (VTK) to find complete meeting plans, including detailed activity instructions and handouts.

Badge 1: Coding Basics

Programmers write code for computers to solve all types of problems from telling the time to researching disease. But for computers to do anything, they need clear and precise instructions on how to perform a task.

Learn how to write code for computers and explore some of the incredible women of computer science.

Steps

1. **Use functions to create a self-portrait**
2. **Write code to create a portrait**
3. **Learn about computer logic**
4. **Explore "IF" statements**
5. **Use computer logic to create a quiz show**

Purpose

When I've earned this badge, I'll know about JavaScript syntax, functions and arguments, and x-y coordinates. I'll know how computers make decisions using Boolean logic, IF statements, and control flow—and how programming can make a positive difference in the world.

STEP
1 Use functions to create a self-portrait

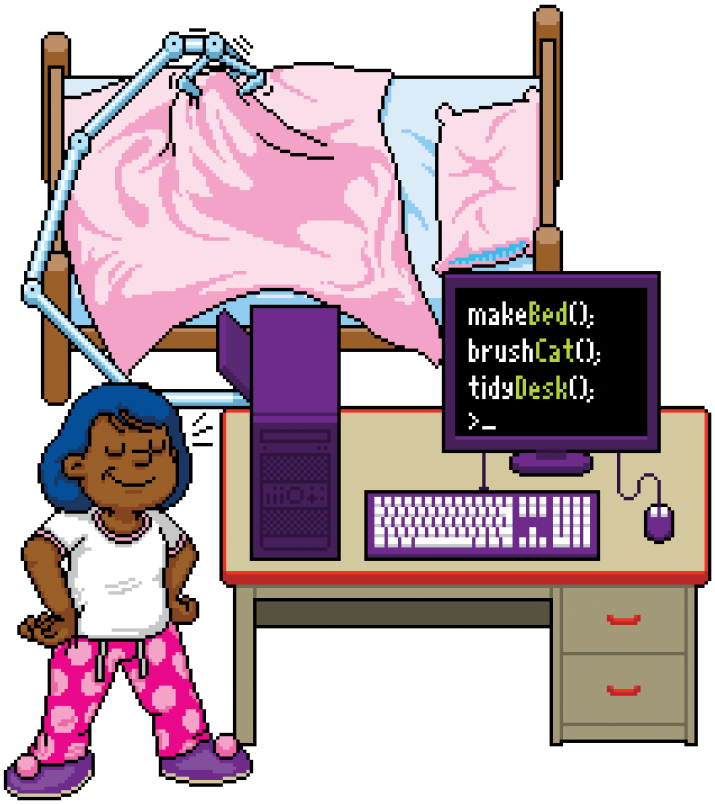
Computers are great at following directions. In fact, that's all they can do! When you write a computer program, you have to tell the computer exactly what to do, step-by-step, because it can't figure the steps out by itself.

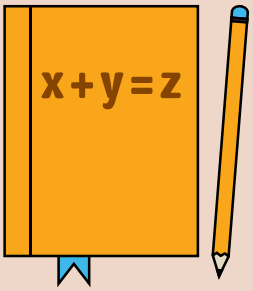
The instructions for a computer are called an **algorithm**. The part of the algorithm that tells the computer to do something specific are called **functions**. To make writing programs easier and more efficient, programmers write more general algorithms and then add details in the functions, which can be more easily switched out.

For example, you could write an algorithm for doing your chores that had sections for your bedroom and pet care. The functions could be makeBed(), tidyDesk(), putawayClothes() for the bedroom and feedCat(), scoopLitterbox(), and brushCat() for pet care.

Computers don't speak human languages. They have their own languages, like **JavaScript**, with their own grammar rules, called **syntax**. Computers can only understand algorithms written with correct syntax, so programmers have to learn the rules for each programming language. The chores functions above are written with JavaScript syntax.

What would it be like if you had to give detailed directions for people to do something, the way you have to for computers? When would that be a good thing? When would that be bad?





Where Did "Algorithm" Come From?

Algorithm is the Latin translation of a man's name: Muhammad ibn Musa al-Khwarizmi, which means "native of Khwarizmi" in Persian. He was a 9th century scholar, geographer, astronomer, and mathematician known mostly as the father of algebra.

Around 825, he wrote a book about the Hindu-Arabic numeral system. When the book was translated into Latin, the title included his name. The Latinization of his name is Algorithmus. Now you know where both algebra and algorithms come from!

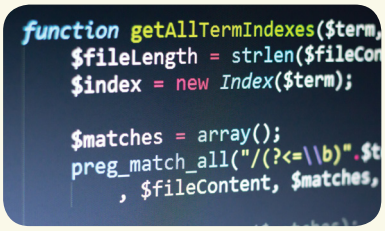
WORDS TO KNOW

Algorithm a series of specific instructions. By creating a sequence of instructions that can be applied to many circumstances, you're creating an algorithm.

Argument a part of a code that makes a function more specific and reusable in a number of different ways: it adds details to the function that are changeable. In many programming languages, arguments are represented as a list separated by commas inside the parentheses.

Boolean expression a statement that can either be true or false, and nothing else.

Code a series of instructions that make up a program directing a computer to do something.



Computer an electronic machine that can store and process data. A computer has hardware, which is the machine itself, and software, which is a set of instructions.

Control flow computers read programs line by line. But when a decision has to be made, the computer doesn't read all the statements—it makes a choice as to which statements to execute. The choosing of what to execute and what to ignore is control flow. An IF statement is one way to direct the control flow for a computer.

Function one of the basic building blocks of a program. It's a type of instruction similar to a verb: a function does something. In JavaScript, as in most coding languages, it has a special form: the name of the function followed by '()'. For example, turnLeft() and drawEye() are two examples of functions. The () tells the computer to "do" the named function. "Doing" a function is typically described as "calling" a function or a "function call."

IF statement this tests whether some "condition" is true or false. The program then executes the code in the true branch if the condition is true. Often, but not always, there is a second branch which executes when the condition is false.

JavaScript a computer programming language.

Programmer a person who writes algorithms to create programs or code for computers.

Pseudocode a way to plan a computer program using human-friendly language. It's not actual programming, but a written description of the key elements of an algorithm or program. It's used as a quick way of thinking about a program without completely writing it out in code.

Sequence the order in which the computer performs the steps the programmer writes.

Software the end product of written computer code.

Syntax the rules for how a program is written. These rules have a purpose similar to written grammar: it's a standard format for writing code that the computer understands. In programming, the syntax needs to be exactly correct for a computer to know what to do. For this reason, programmers often use pseudocode to help them flesh out ideas without the burden of being too exact.

X-Y coordinates when programmers put images on screens, they use a grid that represents the screen. Each square in the grid has a reference number like an address. This number is a square's location and can be written as a pair of numbers: the first number is the X (horizontal position), and the second one is the Y (vertical position). The grid's numbers start with the top left corner square as (0,0), which is called the origin.

Do You Love Wi-Fi? Thank a Woman!

Hedy Lamarr was a famous actress in the 1940s. She was also a self-taught engineer and inventor. During WWII she collaborated on a project to improve radio-controlled torpedoes. Her invention created the basis for Wi-Fi! Thanks, Hedy!

STEP 2

Write code to create a portrait

Functions make algorithms more specific, but sometimes that's not enough. Enter **arguments**! Functions tell the computer to do a specific task. Arguments tell the computer more about the task, like how or where to do it.

For example, in the chores function, `makeBed()` could be made even more specific by adding in arguments to tell you which rooms to tidy: `makeBed("guest room", "my room")`

Just like functions, for the computer to understand your arguments, you have to use correct syntax. In JavaScript, the arguments go inside the function's parentheses and are separated by commas.

Unlike real life, arguments can make things easier and better in coding!

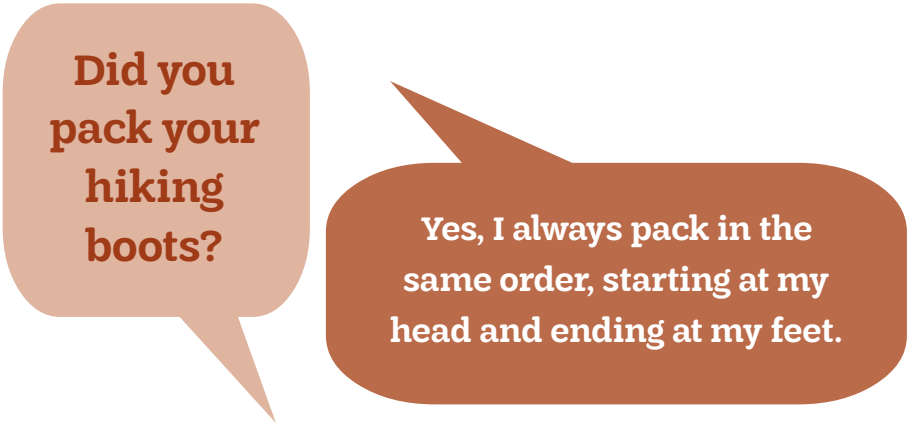
STEP 3

Learn about computer logic

Imagine you're given a set of tasks with each task on a different sheet of paper. Now, imagine the sheets of paper are shuffled before they're handed to you—how do you know where to begin?

Just like you'd need help to figure out what tasks to prioritize, computers need to be told in what order to do their tasks. This is called **control flow**.

Control flow is the sequence in which the algorithms are done and the order of the different decisions computers can make, usually by reading code from the top to bottom.



CODE THAT SAVES LIVES!

Advances in technology have changed the way first responders, like paramedics, search and rescue teams, and firefighters, help people. For example:

- Cars have onboard emergency systems that use digital location and communication systems to call for emergency help after an accident,
- Firefighters use artificial intelligence programs to help them navigate their way through burning buildings,
- Paramedics diagnose heart attacks through programs that recognize speech patterns, and
- Search and rescue teams use drones and GPS to observe weather conditions and find exact locations in the wilderness.

Medicine has even started using artificial intelligence (AI), or computers that can learn from experience, to help diagnose illnesses.

- Computers can read mammograms in more detail than human doctors and detect breast cancer at early stages.
- AI is better than people are at diagnosing an eye disease that affects people with diabetes!
- Robots help people remember to take their medications after heart surgery and ask questions to monitor their recovery.

All of these technologies use computers that have been programmed by people like you!

How to Code an IF Statement in JavaScript

When you're writing an IF statement in JavaScript, make sure you follow the correct syntax:

- The keyword "if" is lowercase.
- The Boolean expression goes in the parentheses.
- There's an opening curly bracket.
- Then, there's some code to execute (only when true).
- There's a closing curly bracket, the keyword "else", and another opening curly bracket.
- Then, there's some code to execute (only when false).
- Finally, there's the closing bracket.

STEP 4

Explore "IF" statements

But what if you're writing code and the computer has to make a choice? A computer uses Boolean logic to know how to react to different conditions, when a situation is either true or false. The program then executes the code in the true branch of the code if the condition is true. Often, but not always, there is a second branch of code which executes when the condition is false. The condition is a **Boolean expression**.

To code a condition or a question so a computer can answer it, you can use an **IF statement**. An IF statement in code tests whether some "condition" is true or false.

You can use an IF statement to turn a Boolean expression into a question like this in pseudocode:

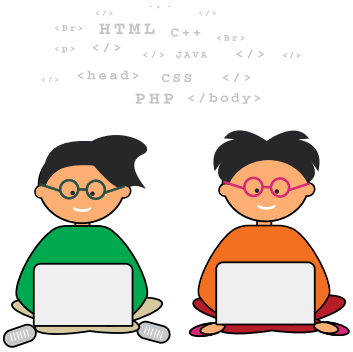
```
IF statement is TRUE
THEN say this
ELSE say that.
```

Here's what that looks like in JavaScript syntax:

```
if (Boolean expression) {
        do something when true
} else {
        or do something else when false
}
```

STEP 5

Use computer logic to create a quiz show



Now that you know how to code IF statements, you can write computer programs with all kinds of conditions!
For example, you can write a set of questions using Boolean logic to teach others about something you know a lot about!

You can use this same idea to create questions about anything that can be answered with true or false. Depending upon the answer to your question, you can then change the control flow for a computer to give different responses.

This is control flow in action! One branch of code is executed, or the other is: Either the TRUE branch or the FALSE branch.

Now that I've earned this badge, I can give service by:

- Playing the Coding Role Models quiz with schoolmates to teach others about women in computer science.
- Teaching a workshop about the fundamentals of coding.
- Organizing a "meet the programmer" event for younger girls and inviting female programmers to talk about what they do.

I'm inspired to:

Badge 2: Digital Game Design

Most days, simple choices, like which shoes you wear, might not matter, but sometimes a simple choice can have a big impact. What if you wore sandals and there was an unexpected snow storm? What if you forgot your cleats and had a big soccer game?

When designers create video games, they develop every possible decision and consequence. Game makers build in choices to make a game fun and challenging.

Steps

1. **Brainstorm your game "for good" scenario**
2. **Create a character for your game**
3. **Learn about decision trees in game design**
4. **Design your game**
5. **Playtest and iterate your game**

Purpose

When I've earned this badge, I'll know about narrative game design, how to create the elements and mechanics of a game, and how games can make a positive change in the world.

STEP

1 Brainstorm your game "for good" scenario

What if playing a game could change the world? What if you could help scientists understand the brain or feed hungry people by playing a video game? You can!

Digital game designers try to create games that are fun, but they can also design games that have a positive impact. Games can teach new skills, raise awareness about important issues, or help scientists do research.

An important part of making a great game is creating a good scenario. The **scenario** in a game is the set up: the environment, the sequence of events, and the main challenges the characters face.

What issues are important to you? What kind of game could you design to create positive action for your cause? How would the scenario you create affect your game?

WORDS TO KNOW

Avatar an electronic image that represents a person or character. Avatars can be manipulated by a computer user, like the player of a video game.

Condition a type of statement or test which will result in the condition being either true or false. It's used in the 'decision-making' part of an IF-ELSE statement.

Consequence in decision trees, a consequence refers to the result of a decision that has been made.

Decision tree a tool often used to design video games. It has a flowchart, or tree structure, that helps game developers design the structure and logic of the player's choices and consequences.

Game mechanics the instructions given to the computer on how the game is played. They're specific to the type of game: for example, in chess, all the moves relate to the game pieces. In video games, the rules of the world created by the game's designers are game mechanics. This can include how avatars move and how players beat a level.

IF-ELSE statement tests whether a condition is true and then runs one piece of code if the condition is true, or another if it is false. They're used by computers to make decisions.

Narrative the story in a video game. It can have many different paths that are created by players making choices and facing consequences.

Node one element of a decision tree. This is the part of the decision tree where the question lives. The first node of the decision tree is called the **root node**. The nodes that come after the decisions are called **child nodes**. Nodes that don't have children are called **leaves** or **leaf nodes** (like with real trees, the leaves are at the end of the branches). In decision trees for game design, leaf nodes represent the end of one possible game. Trees can have multiple levels of child nodes and many leaves.

Playtest playing a newly developed game to test it for flaws and to identify possible improvements.

Scenario the details of a situation, including settings and sequences of events for a game, scene, or plot. It's part of the setup in many types of games.

Computer Games that Make You Computer Savvy

What better way to learn the ins and outs of computers and digital news than to play a computer game!

- **Fakey** is a game that teaches people to recognize fake news and learn how to fact-check suspicious stories.

- **DigiZen** is a game that teaches about cyberbullying.
- **The Case of the Cyber Criminal** helps people learn about how to keep their personal information safe.

What other things could video games teach about computer science?

STEP 2

Create a character for your game

What's the first thing you notice about a video game character? Probably how she looks. Her look is important—it can tell you a lot about her. What she wears and how she moves are part of her look, and they can convey information about her life. Is she athletic? Does she have a job that requires her to wear a uniform?

There's more to a great video game character than just her look, though. How she interacts with other characters and the choices she makes tell a lot, too. Is she a leader or a follower? Does she take risks? How does she treat people? How does she solve problems?

When programmers write code to create characters in a video game, they need to write graphics algorithms for her look. They also need to write algorithms that present the challenges she'll face, the choices she can make, and the consequences of her choices. As users play the game, they'll learn about a character by seeing what kinds of choices they make.

How do people get to know you in the real world? Do they make assumptions about you based on your looks? What does how you interact with other people or the choices you make tell people about you?

STEP 3

Learn about decision trees in game design

Video games can have an open-ended story that lets players choose what they want to happen. Game makers use decision trees, which are like flowcharts, to create choices and consequences for characters.

Some decision trees offer just two choices for each situation. Some offer a lot more, but the more choices in the decision tree, the more code the programmer has to write. The only limits are really the designer's imagination and the time the team has to write the code.

Think about a decision tree like a hike in the woods:

- **As you're walking along the trail, you come to an intersection. You have two trails to choose from.**
- **You choose the one on the left, and so you continue down that path.**
- **The next time you have to choose which way to go, you have three choices: left, center, or right.**
- **You choose center and keep walking.**
- **Every time you make a choice, you are sent down a specific path. Being on that path has certain consequences. Is the trail steep or rocky? Is the trail leading you in the direction you want to go? Did that trail get washed out in a big storm?**

Decision trees work the same way. As the game designer, you get to think up all the choices and the consequences in your game. Then, you write the code for it. The more choices you want your players to have, the more code you write.

STEP 4

Design your game

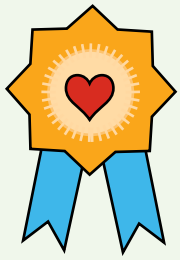
Scenarios, characters, and decision trees are the key ingredients in your video game. Game makers use scenarios and what they know about their characters to create the decision trees. These decision trees, with choices and consequences, will direct their player's experience.

You may find, as you're combining your scenario, characters, and choices and consequences, that you want to make changes. You may be inspired to offer some different options in your decision trees, give your characters a new skill or personality trait, or tweak the environment in your scenario. That's great! Many times, your best idea isn't your first or even second idea. Making a game is an like an open-ended story. Make creative choices!

Need to Learn a New Job Skill? Game ON!

When components of video games are applied to non-game situations, it's called **gamification**. In the world of medicine, gamification is changing the way professionals are trained and how patients experience their treatment.

For example, surgeons and nurses can use computer simulations of surgery or other medical situations to practice and improve their skills. For patients, programs are designed with a game-like "goal-action-reward." These games help motivate and guide patients through their treatment or rehabilitation.



Computer Pioneer: JEAN SAMMET

Jean Sammet was a mathematician who pioneered computer programming. As a high schooler, she was very good at math, but wasn't allowed to attend the prestigious Bronx High School of Science because it didn't accept girls at the time.

That didn't stop Jean. She went on to earn bachelor's and master's degrees, and worked on a doctorate. While she was working toward a PhD at the University of Illinois, she encountered her first computer. At first, she dismissed it as a big piece of hardware.

It was working for a life insurance company that introduced Jean to how useful computers could be. She first learned about punched card accounting machines (a kind of early computer). From there, she started writing computer programs and eventually helped to create two important computer languages: FORTRAN and COBOL.

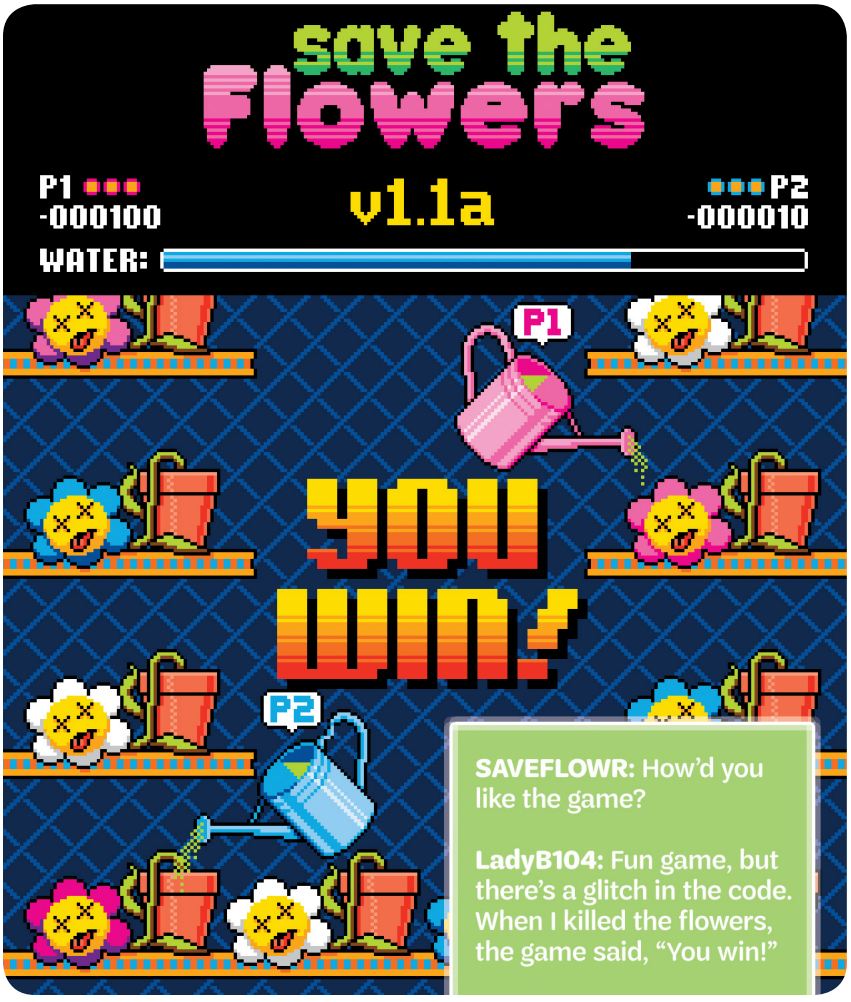
STEP 5

Playtest and iterate your game

So you designed a video game, but how do you know if it works? Once game makers have created a game, they **playtest** it. That means they have someone play the game to see if the scenario makes sense, the characters are fleshed-out enough, and the decision trees work as the maker intended.

Game designers may get all different kinds of feedback that they use to improve their games. For instance, testers could feel like the choices didn't make sense for the characters or the challenges were in the wrong order. Getting someone else's perspective on your design is important. You have an image in your mind of what the game could be like, but building what you've imagined in code is hard.

Playtesting your game lets you know if you've been successful or if you need to make changes. Each time you make changes to improve your game, you create an **iteration**, or improved version. Iteration is a crucial part of good game design.



Now that I've earned this badge, I can give service by:

- Inviting other people to play my game and explaining how decision trees work in game design.
- Hosting a Bring-Your-Own-Device (BYOD) game night, where kids can come together and play video games.
- Creating a video game review blog to highlight games that make a positive impact on the world.

I'm inspired to:

Badge 3: App Development

Some apps are all about storing and displaying data. Calendar apps keep track of your appointments. Photo apps take and store pictures for you. Fitness apps might keep track of your steps, sleep, water intake, and food choices.

Apps aren't just for individuals, though. Some apps, like the ones school systems use to let families know about snow days or ones that help people find and adopt pets, are useful ways to connect communities. Apps can let you collect and use data for community engagement.

Steps

1. **Learn to collect and visualize community data**
2. **Write objects to organize and store data**
3. **Design a community data collection plan**
4. **Analyze your community data**
5. **Develop a prototype for a social app**

Purpose

When I've earned this badge, I'll understand JavaScript syntax, how to use objects to store community and social data, and how to display data with visualizations.

STEP

1 Learn to collect and visualize community data

Collecting data from a community, organizing it, analyzing it, and presenting it in an easy-to-understand way is a big job. **Data** means any set of facts or statistics collected for analysis. **Data visualization** is a way of presenting data in a graphic way. Sometimes people use charts, graphs, or maps to convey data. Visualization lets the person looking at the data see the results of the analysis, like patterns or relationships.

In data collection and visualization, it's important to make sure the data you collect and share is accurate. It's also important that you find an effective way to share your findings, so that people can understand and use the information you've gathered.

What community issue is important to you? How would you gather data on that issue to inform any action you might take? And how would the type of data you gather impact your choice of data visualization tools?

Everybody Counts, So Count Everybody!

Every ten years, the federal government conducts a census. That means they collect information about all the people who live in the country by conducting a survey. What they learn affects how seats in the US House of Representatives are apportioned and how the government decides where and what to spend money on, like highways, medical research, or education. The US Census also collects facts about people, like where they live, how much money they make, and if they're in school or are veterans.

Collecting community data can also help leaders make informed policy choices. For example, community surveys can be used to measure **sentiments** (feelings or opinions) about issues such as the need for a new park. They can gather data and opinions from community members about issues like whether a student representative should be added to the school board.

Friendly Neighborhood Apps

Apps can strengthen communities by helping people connect or share information.

- **Freecycle** lets people give away unwanted items to others who need them.
- **Meetup** helps people with common interests connect.
- **Facebook Safety Check** lets people experiencing an emergency check in with friends and family.
- Some communities use apps to report graffiti, or potholes that need filling
- School districts use apps to notify families about snow days or emergencies.

What other apps do you know of that solve community issues?

Is Your Data Safe?

Before you install an app, make sure you know what kind of data it will collect from your phone *and* what it will do with it. Many apps ask for access to your camera and microphone, to track your location, or read your emails and texts. Some even ask for access to your contacts.

Before you install an app, ask yourself:

- What kind of information is the app requesting access to?
- Am I comfortable sharing my information with this app developer?
- Does the app really need all the permissions it's asking for?

Then, read the privacy policy. It will tell you what the developer plans on doing with your private data. After that you can go to the privacy settings and turn off access to things like your contacts, calendar, photos, or location sharing. Some apps even want to make posts to social media for you. Say "no" to that, if you aren't comfortable with the app making posts.

STEP 2 Write objects to organize and store data

Computers are great "number crunchers." That means that a computer can sort or analyze large sets of data very easily.

Data objects are one way app developers can store and use the data they collect. They're a kind of container that can store lots of different types of information.

When coding an object, it's important to know that objects are made out of **key-value pairs**. Each pair is separated by a comma.

You start with the object, then include data information inside the { and close with }. These are called **curly brackets**—they show the computer the beginning and end of the object.

Objects should have one name, but they can be made of more than one word without spaces. The name of an object must not have spaces and cannot start with a number or use special characters (*, %, &, etc.). Many coders create names from two words that relate to the data the object contains; the first word is lowercase, and the second word is capitalized. This is the typical way of writing an object in JavaScript, but it's not necessary.

For example, this code shows how a computer programmer might represent community data that answers the question, "Is the environment important?" in a data object:

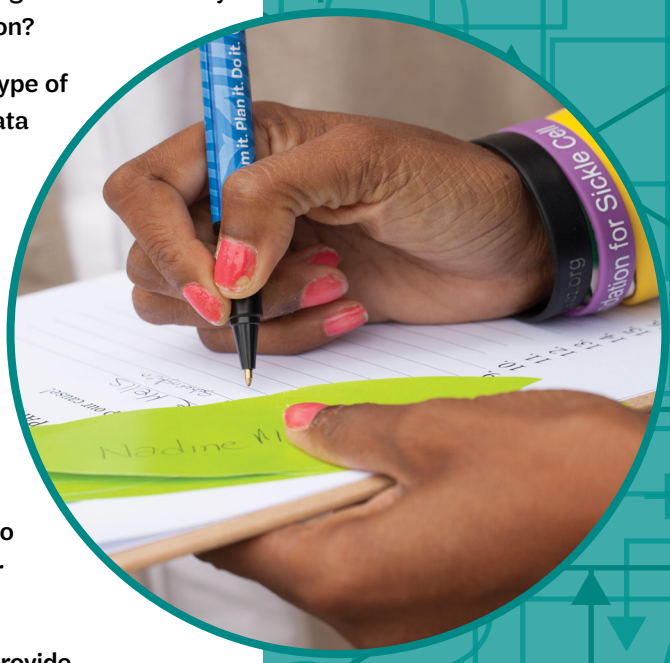
```
var theEnvironmentIsImportant = {
        No: 0,
        Somewhat: 3,
        Yes: 13
}
```

The "var" at the beginning stands for **variable**. You have to add 'var' before the name because JavaScript syntax rules require you to call out your object as a variable.

STEP

3 Design a community data collection plan

A great app hits a "sweet spot," providing just the right service to a certain group of people. When designing a great app for a community, developers have to do some research first. To gather data from a community, app teams think through the process before they start asking questions.



- **Define the community and goal**. Who are they asking and what do they want to know? What issue or issues will they focus on?
- **Decide how to organize or quantify findings**. The type of questions researchers ask will shape the kinds of data they gather.

  For example, will they ask, "On a scale of one to five, with 1 being not important and 5 being important, how do you feel about the issue?" or will they ask, "Which current community issue is the most important to you? A, B, C, or D?"
- **Decide what questions to ask.** Researchers need to decide what kinds of questions and how many to ask. They need to decide if they're going to ask only questions about issues or if they're also going to ask questions about the person answering, like their age, gender, or zip code.

  Information about the people taking the survey can provide interesting ways to analyze the data. For example, researchers might find that people living in a certain zip code think school overcrowding is most important, or that women are most concerned about housing costs.
- **Share the data collection plan and ask for feedback.** Researchers will share their plan and surveys with coworkers before they use it with the public. For example, they'll ask if their questions make sense and if they've identified the right community in which to collect data.

The more carefully you think through the process of data collection before you start to gather data, the easier it will be to organize, analyze, and present it when you're done.

WORDS TO KNOW

App stands for application, used to refer to self-contained software that a user interacts with on different devices. Apps can be used for different purposes like organizing information (such as research notes or to-do lists), providing a service (such as giving map directions or translating between languages), or providing entertainment (such as playing a game or a video).

Bar chart a graphical way to display data using bars of different heights.

Data any set of facts or statistics collected and analyzed or used for reference. Data can be in many forms and include information like steps taken, photos shared, or emails and messages sent. It can also include information collected from a cell phone or other device, such as the location history, internet browsing history, or login names and passwords.

Data visualization a way data scientists, computer programmers, designers, and others communicate information clearly and efficiently. Data visualization uses statistical graphics, plots, information graphics, and other tools. Effective visualization helps users analyze and think about data. It makes complex data more accessible, understandable, and usable.

Empathy the ability to understand how someone else feels.

Leading question a question that prompts a particular answer and will likely result in skewed answers. For example, "Since the lunch food is unhealthy, how likely are you to buy lunch outside of school?" is a leading question because it implies that you SHOULD buy your lunch outside of school because the food is unhealthy.

Object a way of storing lots of different types of data. In most programming languages, objects are represented with curly braces **{}**, with the content of the object between them. Objects are based on the idea of **properties** and **values**. When you code an object, each property-value pair is separated with a comma. Think of objects as being like a dictionary: a word to look up and a definition of it. Properties are the word to look up, and values are the definition.

Pie chart a type of data visualization that represents data as slices of a pie. Pie charts are best used to show the relationship between a data point and the whole. In other words, they're good at showing the percentage.

Pitch a business presentation seeking support from people to invest in a new or buy a new product. It can be an email, letter, or even a conversation. Sometimes the presentation can be a "sales pitch," where the goal is to get a user to buy a product.

Prototype a first version of a product which is built to be tested so that changes can be made before production.

Sentiment a view or attitude toward a situation or event; an opinion. Programmers have many ways to collect and analyze sentiment data from social media.

Social media (Social apps) technology based on the creation and sharing of information, ideas, interests, and other forms of expression. They're built using the internet and can work on their own or be part of another website or web-based service. Social media users usually create profiles and share content like text posts, comments, digital photos, and videos. Social apps can help the development of communities by connecting a user's profile with those of other individuals or groups.

User interface the visual elements of a program through which a user controls or communicates with an app. Often abbreviated UI.

STEP

4 Analyze your community data

You've gathered great data. Now what? Once app developers have collected data, they need to organize and analyze it before they can make a great app. That means they'll look at their data to see if they can find any patterns or relationships. They may use data visualizations to show their main findings. All the information they've gathered will inform their app design.

When trying to analyze their data though, they might find they don't have enough data, or that their questions weren't specific enough, to draw any conclusions. Or they might find that the question they asked influenced the answers they got. That's called a **leading question**.

For example, if you asked, "On a scale of 1 to 10, how do you feel about the dangerous intersection in front of the school?" you might have influenced people to think the intersection is more dangerous than they would have otherwise. This leading question may have skewed, or influenced, their answers, but if you asked, "On a scale of 1 to 10, rank the safety of the intersection in front of the school, with 1 being very safe and 10 being very dangerous," you might get less biased responses.

STEP

5 Develop a prototype for a social app

You've got the data, now go for it and design your app!

Once developers have gathered data from the people who would use an app, they use the data to help shape their app design.

When developers start working on their app, they need to identify

- **what they want the app to do,**
- **the data they will collect, how they will collect it, and what kind of data visualization they'll create for users to see and understand the data,**
- **the user interface, or the visual elements of the app that the user controls, like the design of the landing page, icons, or buttons, and**
- **any additional features like ways to connect with friends or community members and access to GPS or calendars.**

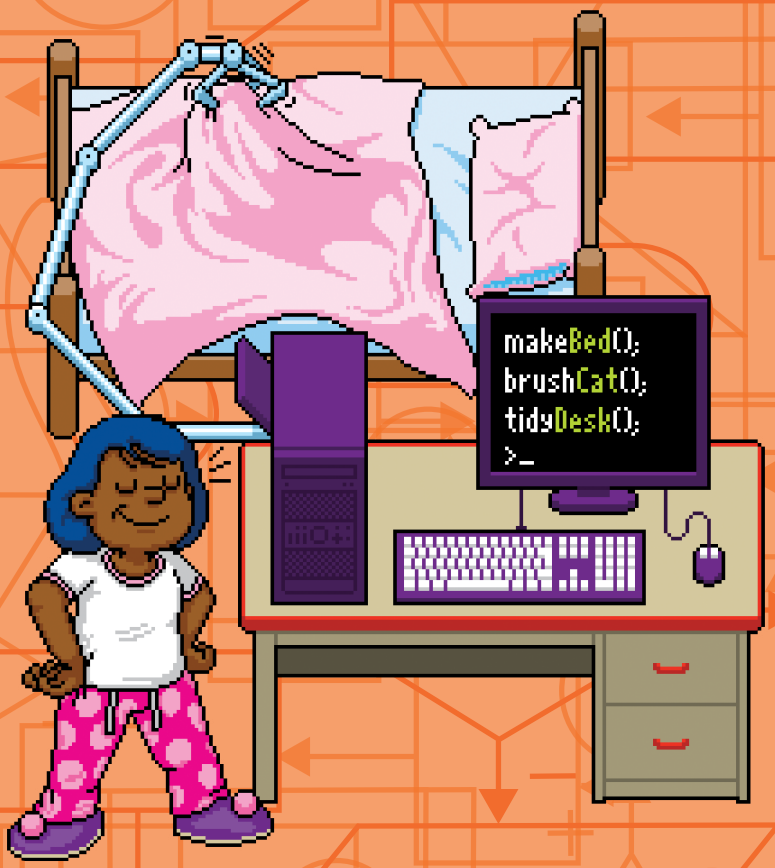
When they've identified all these elements, they can start writing the computer code to create their app.

What kind of app could you create to address a community issue? How could you use data visualization to help people understand an issue better? What features could you include to design your app to build awareness about an issue or build support for positive change?

Now that I've earned this badge, I can give service by:

- Creating a video that shows people how to limit the data collected by apps by using the privacy settings.
- Using social apps to share positive messages and raise awareness of issues that are important to me.
- Organizing an "Apps for Good Hackathon" where students can develop apps to address community issues and learn more about coding from local programming professionals.

I'm inspired to:



Made possible by a generous grant from At&T

First published in 2019 by Girl Scouts of the USA
420 Fifth Avenue, New York, NY 10018-2798
www.girlscouts.org

Printed in the United States

Jean Sammet courtesy Ben Barnhart

UPC 64059

7 31955 64059 0