Junior Coding for Good

Find out how to create computer programs, games, and apps that solve problems and help others by earning these three badges.

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 Junior 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.

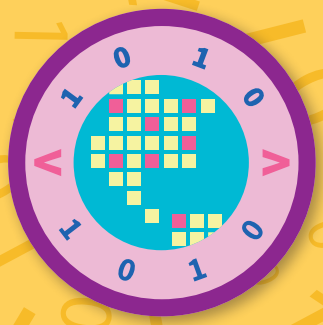
Welcome to the world of coding.

When you've earned these three badges, you'll know how programmers make computers do useful things and how computers can be used to help people.

- You'll know how to write a computer program.
- You'll know how video games are created.
- You'll know the steps app developers take to create apps people will like.

You'll also learn about important female computer scientists. You'll explore how to create technology that solves problems and helps people, too.

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



Badge 1: Coding Basics

From tablets and laptops to microwave ovens, cars, and even stoplights, we use computers every day to help us in all sorts of ways. But how do the computers know what to do? People write instructions for them! When you learn to "talk to computers" by writing code, you'll be able to tell computers what to do!

Steps

1. **Create algorithms for a computer that follow a sequence**
2. **Use loops to improve your algorithm**
3. **Keep your code interesting with conditionals**
4. **Create your own set of commands that use conditionals**
5. **Learn about women in computer science**

Purpose

When I've earned this badge, I'll know how programmers use sequence, loops, and conditionals to write computer programs and how people can use computers to help others.

What's a Programmer?

- A **computer** is a machine that can remember information and follow directions. At first, computers only did math and filled entire rooms. Today, they're much smaller. A laptop is a computer, and so is a smart watch.
- **Code** is a special language created by people to tell a computer what to do. For a computer to work, it needs instructions that have been written in a code it understands. There are hundreds of different computer languages, many created by women!
- **Programming** is writing a set of instructions in code. A program tells a computer to do something, like keep a calendar or search for a photo online.
- **Computer programmers** are sometimes called coders. Without coders, computers wouldn't know what to do.

STEP 1

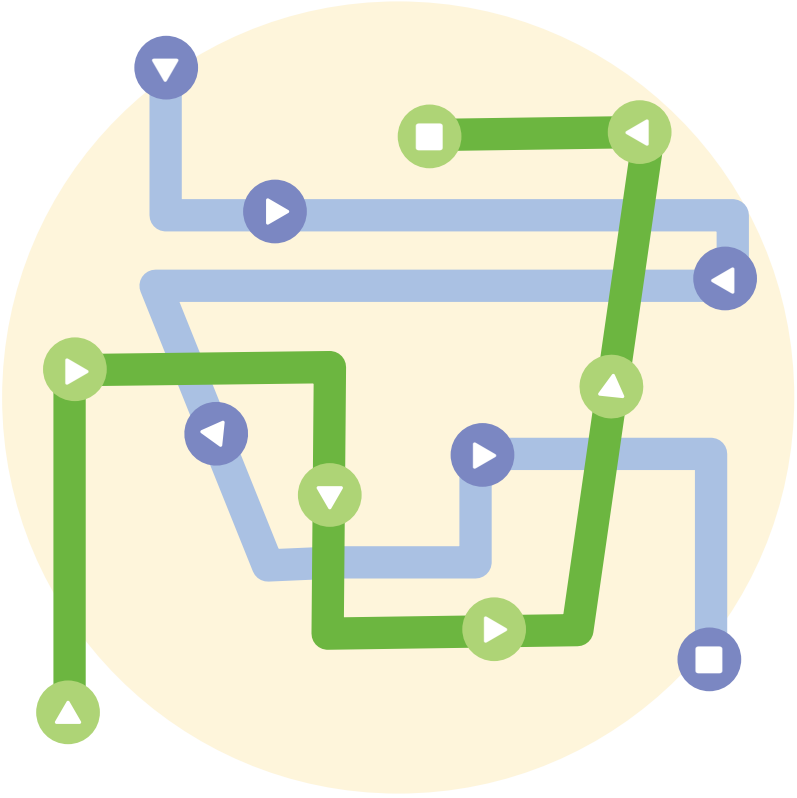
Create algorithms for a computer that follow a sequence

Giving directions to someone can be tricky. Giving directions to a computer is even trickier.

Why? A computer can only do exactly what you tell it to do, in exactly the order you tell it. A computer can't ask questions or figure out what to do on its own if your directions are unclear.

The directions that programmers write is called an **algorithm**. The order of the directions is called a **sequence**.

Sometimes the order of actions matters, and sometimes it doesn't. Imagine you're setting up a campground. You need to find a flat space and clear away rocks and sticks BEFORE you set up your tent. The sequence of your actions matters in this case.



WORDS TO KNOW

Algorithm This is a set of step-by-step instructions for how to do something. A recipe is an algorithm. It tells you all the steps you need to take to cook something. When a friend gives you directions to her house, that's an algorithm, too. She's telling you the steps you need to take to get to her house.

Conditional Programmers code conditional statements to get computers to react to different situations. They're written with IF/ELSE statements: IF something happens, THEN do this. ELSE, do something else.

Efficient programs These computer programs are written to respond quickly and take less memory and power.

ELSE statement In a conditional, when an IF action isn't met, the ELSE action will run.

IF statement In a conditional, something happens when the IF condition is met.

Loop This is when a set of instructions, or an algorithm, is repeated. When you take turns and follow the rules of a game over and over again, that's a loop.

Nested loop In coding, this is a loop within a loop, an inner loop within the body of an outer one. As long as the outer loop continues to run, it will trigger the inner loop each time until the outer loop finishes.

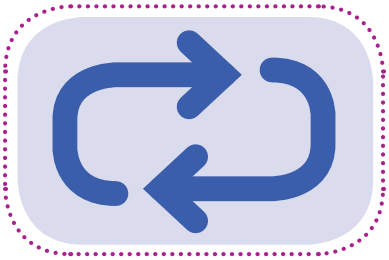
Sequence This is the order in which things happen. The routine you have for getting ready for school in the morning is a sequence. For example, you might write your sequence for getting ready like this: Wake up. Get dressed. Eat breakfast. Brush teeth. Walk to school bus.

STEP 2

Use loops to improve your algorithm

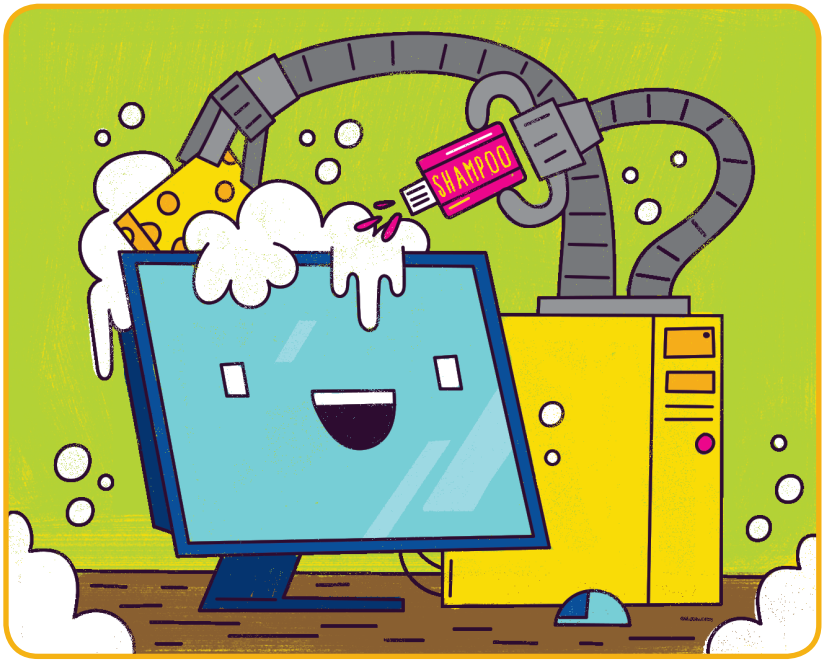
Have you ever read the directions on a shampoo bottle? They often say, "Apply shampoo to your hair. Lather, rinse, repeat." Lather means rub the shampoo around in your wet hair to make lots of bubbles. Rinse means use lots of water to wash the bubbles out. And repeat means—do it all again!

"Lather" and "rinse" are the steps in a hair washing **algorithm**. "Repeat" is a **loop**. It means do the same thing—lather and rinse—over and over again.



Lather is a **nested loop,** because you need to rub the shampoo around on your head in lots of different places—on the top, on both sides by your ears, and in the back. You repeat, or loop, lathering. That repeated action happens within the loop of washing your hair twice.

Programmers use loops to tell computers to repeat actions in their programs. Loops make programs shorter, easier to write, and easier for computers to understand.



STEP 3 Keep your code interesting with conditionals

Computers are great at doing the same thing over and over again. To make them even more useful, programmers have figured out how to get computers to react to different situations. They write programs that say if one thing happens, do this. If it doesn't, do that. These computer commands are called **conditionals**, and you write them using an **IF/ELSE statement**.

For example, if you didn't have the option to wear different jackets to match the weather, you'd always have to wear the same coat. But, with a conditional, you can give options: IF it's snowing, wear a warm coat. ELSE wear a light jacket. By writing code that includes conditionals, such as determining different coats depending on the weather, programmers can make code more flexible and more interesting.

STEP 4 Create your own set of commands that use conditionals

Being a leader means you have to make lots of decisions. So, when you're a leader, it's a good idea to prepare for what could happen and how you would react to all kinds of situations and surprises.

For example, if you're planning a camping trip, what will you do if it rains? Are any of your friends allergic to certain foods? What kinds of meals should the troop plan?

Programmers do the same thing when they write conditionals in their code. They think about different situations in the program and tell the computer what to do IF that situation comes up.

The Power of Imagination

Women have played a key part in the history of coding. For example:

Ada Lovelace wrote the first computer program when she was only 17 years old. She created the code for the first mechanical computer, invented by Charles Babbage in the early 1800s.

Grace Hopper and her team created some of the first electronic computers, like Mark 1, in the 1940s. The computers were huge—the size of whole rooms. Grace predicted that someday people would be able to hold a powerful computer in their hands.

In the 1970s, **Raye Montague** was the first person to figure out how to design a ship using a computer—then she designed one in less than 24 hours!

Computer Pioneers

In the 1960s, **Margaret Hamilton** worked with a team at MIT and NASA to create the computer programs that astronauts used in the Apollo 11 space flight.

Margaret's program told the computer what to do if something went wrong. The day the Apollo 11 astronauts were trying to land on the moon, the computer got confused when it had too many things to do at once. Margaret's code told the computer to land the spaceship and ignore error messages. That saved three astronauts and helped them to land on the moon.

She also came up with the idea of "computer software." "Hardware" means the machine—the keyboard, the screen, the different parts of the computer you can touch. "Software" means the programs that make the computer do things.

After working for NASA, Margaret started her own computer software business.

STEP 5

Learn about women in computer science

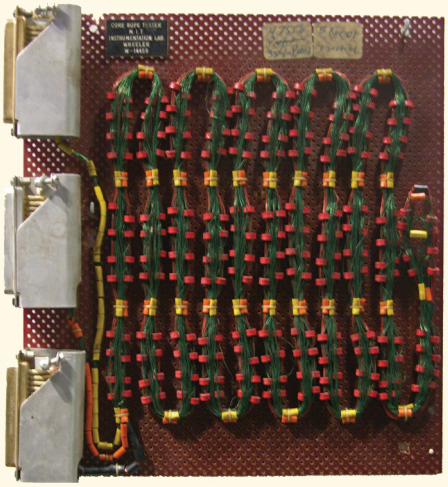
Part of being a leader is thinking ahead, imagining what problems might come up, and figuring out how to solve them. The first computer program was written by a woman, and women have been leading the way ever since! Their leadership and creativity have shaped the world of computer science in many ways. They've designed and built new kinds of computers, invented new programming languages, and even used computers to design ships and send people to the moon!

What kinds of problems would you like to solve with the help of computers?

Building Computers Is Women's Work

Part of what makes a computer a powerful tool is its memory. Today, we can store lots of code on tiny pieces of circuitry in phones and computers, but computer memory used to be much bigger.

For example, NASA needed people with very specific skills to build the Apollo 11 computer memory out of copper wires and magnets. Expert seamstresses carefully threaded the copper wires through and around the magnets. This created a core rope memory that contained Margaret Hamilton's "in case of emergency" programs.

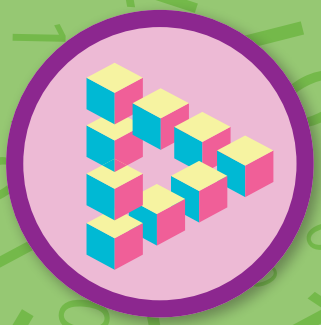


Core rope memory sample used by NASA in the 1960s.

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

- Coding a computer program that can help other people.
- Creating step-by-step algorithms to teach someone else a new skill.
- Sharing what I've learned about women in computer science at school.

I'm inspired to:



Badge 2: Digital Game Design

What makes your favorite video game fun? Have you ever wondered how the creators included all the different challenges and choices? They used algorithms and conditionals to tell the computer what to do. Use what you've learned about coding to create a game that is fun and helps solve a problem!

Steps

1. **Discover how game design can be used "for good"**
2. **Explore tools used to develop digital games**
3. **Plan a maze game**
4. **Build and test your maze game**
5. **Share and improve your maze game**

Purpose

When I've earned this badge, I'll know how video games are developed and how to plan, build, and improve a game by iteration.

STEP

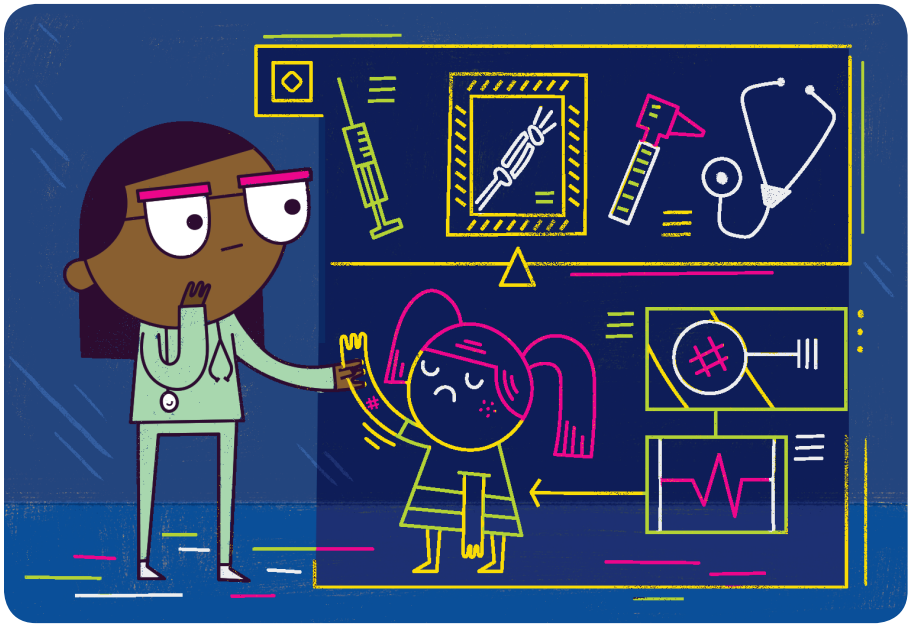
1 Discover how game design can be used “for good”

You’ve learned how computers helped astronauts land on the moon. Did you know that video games can help people, too?

Some jobs use video games to teach people new skills. Doctors and nurses use computer programs that show virtual medical situations to improve their skills. Pilots do the same thing using a simulated cockpit of an airplane.

Games can teach us new things and even help us see things we may never experience. We can find out about places, people, and situations we might not otherwise know about! Through video games, you can visit the International Space Station, dive to the deepest parts of the ocean, or visit ancient cities.

What kind of game could you create to teach someone a new skill?



Play Yourself Healthy

Is playing computer games good for you? It can be!

Some video games were created to help people with illnesses like cancer, diabetes, or asthma. A therapeutic video game was created to help kids with cancer feel more confident during treatment.

Other games get people to be more active. “Exergames” help patients do physical therapy by making them move in fun ways, like bowling or dance competitions.

Game developers have found that when patients enjoy a game, they feel more positive about getting well.

Words to Know

Digital games
These are games you can play on your phone, computer, TV, tablet, or digital gaming machine. They're also called video games.

Game design process
Video games are developed by programmers who imagine, plan, build, and test their designs. They iterate, or go through the steps many times, to improve their video game.

Iteration
This is when you do something many times to make it better. Think about drawing a picture of a flower. You may draw it once, then decide to add leaves. You may draw the flower many times. Each time you draw it, you'll make it a little better.

Perseverance
This is when something is challenging, but you don't give up. You keep trying. Learning to ride a bicycle is hard. You lose your balance a lot. You might fall down. When you keep trying to learn, even if it is hard, you have perseverance.

STEP 2 Explore tools used to develop digital games

What do you need to know to make a video game?
Computer programmers use the same ideas in every kind of program, including video games. You already know how to write **algorithms** with **sequence** and **loops**. Loops let players practice skills and get better.

You also know how conditionals create choices in the program. Game makers also use **conditionals** to make the game more exciting. Conditionals in games let players experience different things.

Think about where there are conditionals and loops in your favorite video game. Then, brainstorm how you can add or adapt them for your own game.

STEP 3 Plan a maze game

When you create a video game, you first need to decide what kind of game you want to make. Then, you plan, build, test, and improve it. When you're happy with the game you've created, you share it with others. These steps are parts of the game design process. You can use this process to create a level of a video game or to work on any big project.

BRAINSTORM

STEP

4 Build and test your maze game

Making anything new usually involves lots of trial and error. That means that the first version (or second, or third) probably won't work the way you want it to. You have to repeat the design process, practicing **perseverance**. Perseverance is when you keep working on a project, even though it's difficult.

Each time you repeat steps in the design process is an **iteration**. If you're trying to improve a cookie recipe, every time you make a batch, taste test, and change the recipe is an iteration.

Making a great video game also requires iteration and perseverance. As you test your game, you might find a mistake in your code that you need to fix. You might also think of a new feature you can add to make the game more fun.

Remember, if at first you don't succeed, try, try again!

CREATING COOL CHARACTERS

Do you have a favorite video game character? What do you like about her? Lots of things make a character special: her creativity and problem-solving skills, her ability to work with others, her leadership skills, her attitude, her history, and her look.

Video game designers have to make lots of decisions to develop and code their game characters.

☐ DESIGN CHOICES

Game designers make visual, audio (sound), and design choices about their characters. They have to make many decisions about their character: her body type, height, hair color and length, eye color and shape, clothing, facial expressions, the way she moves, the way she talks, and so much more. How a character looks, sounds, and moves all tell players about her personality.

☐ BACK STORY

A character's back story tells players the character's history and gives clues to how she behaves now. The back story is often told in dialogue or in text that can be read.

☐ CHARACTER INTERACTION

How other characters interact with a main character is important, too. Do other characters respect her? Is she a leader, or does she become one in the course of the game?

☐ DECISION MAKING

Conditionals like IF/ELSE statements create choices and consequences for a character. The choices she can make showcase her creativity, problem-solving skills, leadership, independence, and ability to work with others—or not.

STEP

5 Share and improve your maze game

The best part of finishing a big project is sharing it with other people. When you share your game with others, you get to see how it works and how other people enjoy it.

They might also give you ideas about how to make your game better.

Even after game makers share their game with the public, they might still find mistakes or ways to make the game better. They'll send an update to game users to correct or improve the computer program. Game makers are always learning what works, what their players like, and making improvements.

Brain Power

Your brain is the most powerful computer ever! Scientists are learning more and more about the brain, but there's still so much to discover.

To learn about the brain, scientists created a game called EyeWire. Ordinary people can play the game and solve 3D puzzles based on the shape of brain cells. Every time someone solves a puzzle by mapping the given cell data, the scientists learn more about the brain cells and how they interact.

Having lots of people working on the puzzles gives the scientists new information faster than if they had to map all of the cells by themselves. Since 2012, more than 200,000 people from 150 countries have played the game. Players have even charted new brain circuits and discovered six new kinds of brain cells by playing EyeWire!

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

- Sharing with friends and family how they can play games to help advance scientific or health research.
- Teaching others how to use the game design process.
- Encourage friends to practice perseverance when they're struggling with learning something new.

I'm inspired to:

Badge 3: App Development

Some programmers use their skills to help people by creating useful apps. Apps are software programs that run on computers. Some apps are just for fun—like games or programs to watch videos and listen to music. Other apps help people solve big problems. Can you develop an app that solves a problem for someone else?

Steps

1. **Discover the needs of others**
2. **Decompose the needs of your app user**
3. **Design your app screens**
4. **Include conditionals in your app design**
5. **Share and improve your app with user feedback**

Purpose

When I've earned this badge, I'll know how to use user-centered design to create an app.

STEP

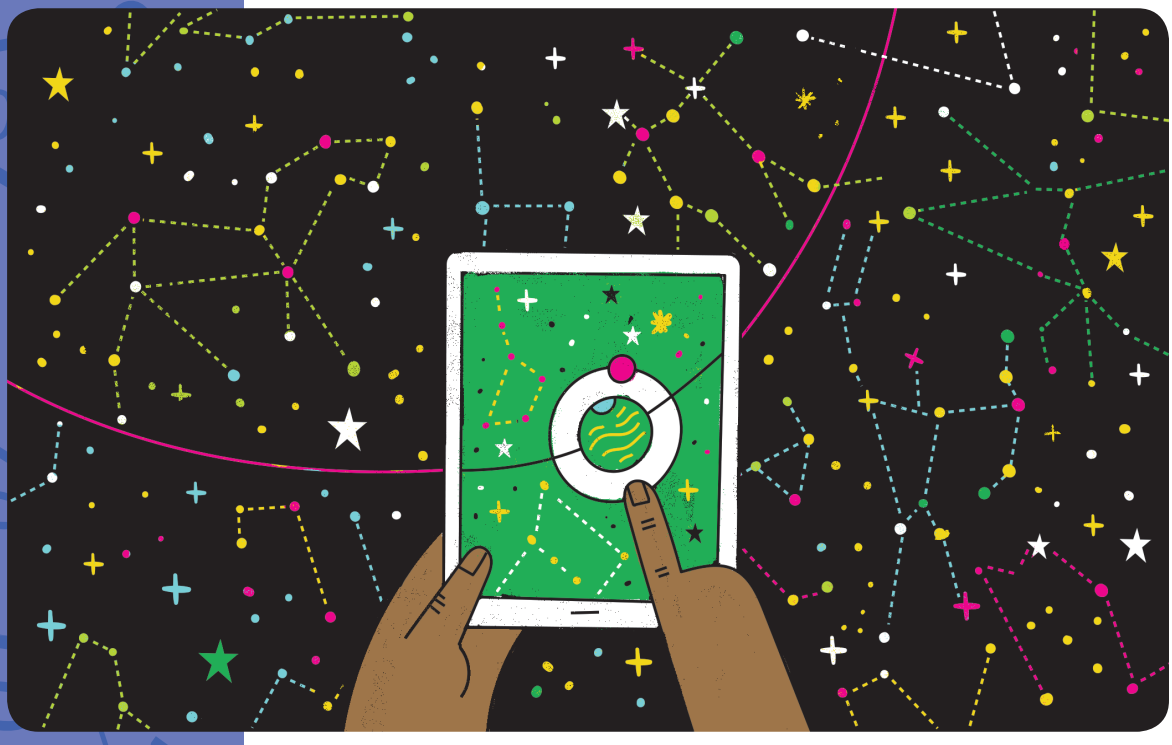
1 Discover the needs of others

Would you like an app that lets you check out ebooks from the library? What about an app that helps you identify stars and constellations in the night sky? And how about an app that lets you draw on your phone or tablet? These are all real apps!

Some apps are easy and fun to use. Others aren't so great. What's your favorite app? What features do you like? Which could you do without?

How do you think the computer programmers who make great apps decided what to include? Chances are they asked people lots of questions about what they needed the app to do. They listened carefully to their potential users and designed the app with those needs in mind. Understanding what people need is the first step in creating a great app.

When you start by thinking about what your app's users will need, you're creating your app with user-centered design.

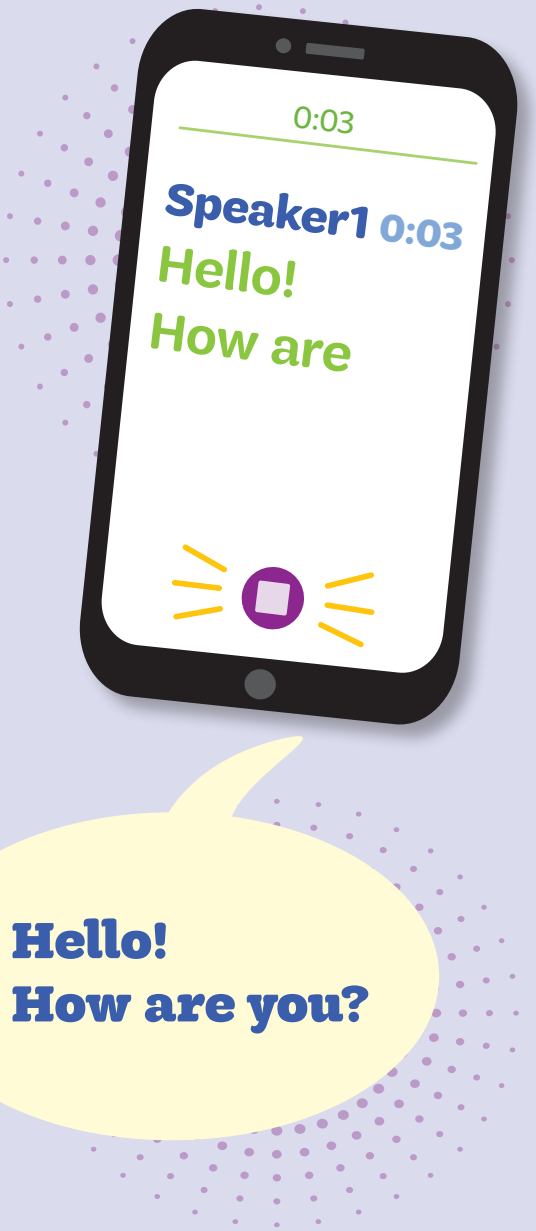


USER-CENTERED DESIGN

What makes a good app? Lots of things, but most importantly, it must work for the people using it. When designers think about what the user needs, that's called "user-centered design." App developers spend a lot of time learning what their app users want and need.

App teams learn what their users need by asking lots of questions. They do interviews and surveys with potential users. Then, they test the app with them and improve it based on their feedback.

Sometimes apps help people in very special ways. For example, Google has created apps for people with hearing impairment. One of the apps uses a smartphone's microphone to transcribe live conversation in real time, so people can read in 70 different languages! Another app turns a smartphone into a kind of hearing aid by improving the clarity of speech nearby and filtering out background noise.



Animal-Friendly Apps

Taking care of pets is hard work. Fortunately, there are all kinds of apps to help pet owners. Apps can help you:

- remember to feed your pets or give them medicine
- train your pet or track their walks
- help you become a pet owner

All over the world, animal shelters and rescue organizations use apps to help homeless pets find their forever homes. If you can't have a real pet, apps can let you have a virtual one including cats, horses, dogs, and even sheep.

STEP 2 Decompose the needs of your app user

Making a great app is complicated. The best way to get started is to think about the people who will use it (your **users**). What do they need in an app? What do they want?

The list of what users need in an app can be really long. Computer programmers break the big list of needs into smaller lists of to-dos to build their app. User needs might include:

- what the screens need to look like
- if the app needs to link to other apps, such as weather, calendar, map, or clock apps
- what kind of tracking users need, such as the number of steps taken, foods eaten, or time spent reading

Computer programmers call this **decomposition**. When you decompose the different needs a user has for an app, it makes it easier to design. Decomposing the problem lets you figure out which needs you can meet and which ones your app won't be able to help with. You can then work on each small part, one at a time. Breaking down big projects into smaller steps is a great problem-solving skill even when you aren't writing computer code!

STEP

3 Design your app screens

When you're building something new, it's always helpful to have a plan!

App developers start planning for a new app by drawing it on paper. This lets them share their new idea with others and gather feedback on how it could be better. Planning on paper lets programmers test out different ideas before they spend time coding.

Once you know what your user needs and what problem you want your app to solve, you can create a plan to build your app. First, you'll want to decide what app features you'll include. Then, you can draw different screens on paper to show how users will experience the app.

Words to Know

App App is short for application. This is a software program that runs on your computer, tablet, or phone. Apps can be entertaining, like when you play a game or watch a movie. They can be helpful, like giving you directions from your house to the soccer field. They can also teach you something new, like a new language.

App Features These are the parts of an app. They could be things like using the camera, a welcome video, a help page, or a way for app users to connect with friends.

Decomposition This is when you break down a problem into smaller steps or pieces to solve.

Development This is when you create something new. When you develop something, you create a plan before you begin building. Then, after you build it, you test to see how it works and find ways to make it even better!

User needs These are what potential or current users need to solve the problem your app is meant to solve.

User-centered design When programmers create an app, they include their users at many stages of the development process. For example, they'll talk to their users before they start planning or have the users test the app once it's built. This helps programmers develop apps that are easy to use and helpful to others.

STEP 4 Include conditionals in your app design

Conditionals make your app more flexible by giving the user choices. They let your app react to different situations, like changing weather, your location, or time of day.

For example, a cooking app could include a conditional that rewrites recipes for vegetarians or people with food allergies. A map app could include conditionals for giving directions if you're in a car, walking, riding a bike, or taking the bus. These app features make the apps more useful. What kinds of conditionals would you add to your app idea?

STEP 5 Share and improve your app with user feedback

When you've created something, you've imagined it and worked hard to make it real. Once you create something, it's always a good idea to show your creation to other people for feedback.

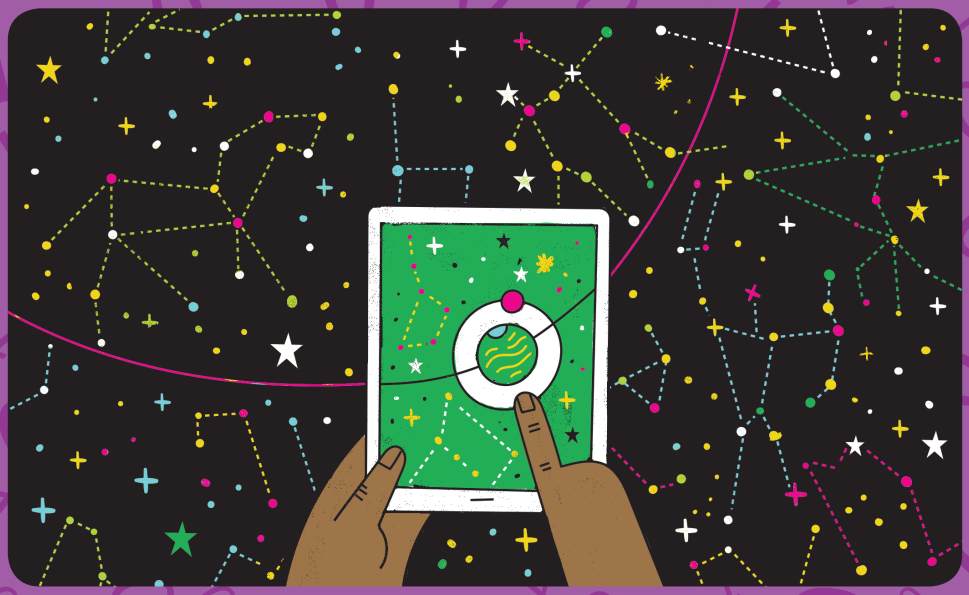
Take your app, for example. You thought of the idea and designed it, but you don't know if it works for your user until you test it. Sharing the app and asking for feedback is the best way to improve your app.

When someone uses your app, they can tell you what works well and what doesn't. Getting feedback and improving your design is an important step in programming. Programmers will have users test their programs over and over to find problems and look for ways to improve. Giving feedback on someone else's design is also a great way to get new ideas to improve your own computer program.

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

- Sharing what I've learned about coding with friends or my class at school.
- Using feedback to make my projects better—and giving others helpful feedback.
- Researching apps that help a cause I care about.

I'm inspired to:



Made possible by a generous grant from Dell

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

Photo of core rope memory: CC: Author: Nova13

UPC 64051

7 31955 64051 4