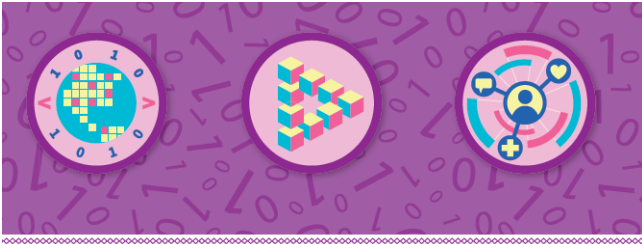


JUNIOR BADGES TO DO AT HOME



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



Junior Coding for Good

To earn this badge, please complete all steps.

STEPS	BADGE REQUIREMENTS	Badge 1: Coding Basics
<p>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?</p>		
Step 1	Discover the needs of others	<p>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.</p>
Step 2	Decompose the needs of your app user	<p>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!</p>
Step 3	Design your app screens	<p>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.</p>

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. Share your app virtually with friends or family. 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.