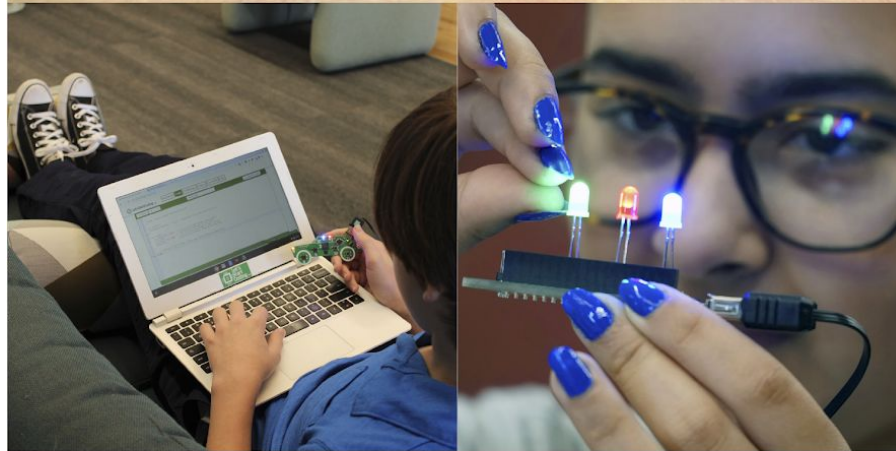


The Let's Start Coding Parents Guide

Helping you help your kids be successful with a Let's Start Coding Kit



INTRODUCTION	3
WHAT'S MY ROLE?	4
1) Help kids get the coding software on your computer	4
2) Review the parts of the kit	4
3) Do a project with them	4
4) Model the Process	5
WHAT IS CODING?	6
HOW DO THESE SKILLS APPLY?	7
WHAT WILL MY CHILD DO WITH THIS KIT?	8
HOW CAN I HELP MY CHILD?	9
1) Tech troubles	9
2) Process (and patience!)	9
3) Help define the pace (or lack thereof!)	10
4) Persistence	10
HOW CAN I ANSWER THEIR QUESTIONS IF I DON'T KNOW THE ANSWERS?	12
"It doesn't work and I don't know why!"	12
"I want it to do X but I can't. Why not?"	12
"How can I make my kit do X?"	13
WHAT OUTCOMES SHOULD I EXPECT?	14
"Hey! Look at this!"	14
Going off script	14
Real-world ties	15
TIES TO FORMAL EDUCATION	16
Scientific Method	16
Engineering Method	16
Variables	16
Electronics Concepts	16
Troubleshooting, Logic, Sequencing	16

INTRODUCTION

WELCOME TO CODING

For many of our customers, a Let's Start Coding kit is the first time coding has been brought into their home. This guide is designed to *help parents help their kids!* If you're not familiar with coding, it can be intimidating to start your child down a path where they may soon outpace you.

Here's the good news: You don't have to know code to help your child learn it!

With a little time, you can understand broadly what your child is doing and use your 'adulting' skills to help facilitate it without touching the keyboard.

We have answered the most common questions that we hear from parents, kids, and teachers in this document. Feel free to skim this guide, skipping to the chapters that are most relevant right now, and revisit it later.

- The Let's Start Coding Team

WHAT'S MY ROLE?

As a parent, you may find that most of your role in getting kids started with a new toy or item is to supply scissors to open the package and batteries to turn the thing on. With a Let's Start Coding kit, you'll go just a couple steps further.

1) Help kids get the coding software on your computer

Let's Start Coding kits have free software for Windows, Mac, and Chromebook computers. The installation method is simple, but if kids aren't familiar with installing software on a computer, you can provide the extra boost they need. This is a one-time-task and takes 5-15 minutes. Get started at www.letsstartcoding.com/start .

TIP: Do this *before* you get your child started with the kit. They will be excited and ready to get started, so saving a few minutes of patience will help keep their enthusiasm high.

2) Review the parts of the kit

Let's Start Coding kits go far beyond what is inside the physical box you and your child will open. There is a free software for your computer that enables your child to program the kit and there are helpful reference cards in the box that explain the circuits themselves and some coding concepts.

Kids are often so excited by the tactile nature of the electronics that they completely pass over the instruction 'cards' that are tucked in the box. These are critical to kids' understanding of what they're doing; the cards help kids learn important vocabulary that will help them learn and play.

Go through the cards that are tucked into the box with your kit. All of them may not make sense yet, but at least get familiar with the different parts of the green circuit board you have and the electronic accessories that go with it.

3) Do a project with them

If you've never coded before (especially with a hands-on circuit), this will be a new process to learn. Luckily, it's straightforward and once the process is understood, kids can repeat it hundreds of times on their own.

Let's Start Coding projects include:

- A guide for connecting the circuit board to the computer (depending on the kit)
- An example computer program for kids to load onto the circuit board from the computer
- An explanation in video or text format about that example program and how it works
- Challenges for the kids to modify the code on their own to create something new

Of course, the first project will take a little bit more time than subsequent projects. You'll need to make sure the circuit board is plugged in correctly so that it is powered on, navigate to the example program, make sure the components are plugged in correctly (Base and Ultimate Kit only), and get that first successful upload of code from the computer to the circuit board.

Once you have a program uploaded to the circuit board, try making changes according to the 'Challenges' section of the software page you're on. Learn how the code is just text that can be edited with the keyboard.

Get stuck? Watch the walkthrough video just below the example code program. It will explain, step-by-step, how the program works and also reveal some coding topics covered there.

By modeling patience and how the software program is used, you'll help set the stage for your child to be able to independently modify the code to their will without your help.

After your child gets started, the process of connecting the board to the computer, uploading the example code, and trying the challenges is repeated for every project, so they'll have the hang of it quickly.

4) Model the Process

Whether you have done the first project with your child or by yourself before giving the kit to them, you can now model the process of coding for them.

What does this mean? A coding kit is a little different than some other childrens' toys: coding requires concentration and creative thinking and the results the child wants to create may take some trial and error. You can help model that it's O.K. for kids to "fail" at their coding attempt and that they should keep puzzling on their project rather than give up.

WHAT IS CODING?

Coding can be defined in many ways, but one basic definition is *creating logical steps for a computer to execute*. Typically, the steps are designed and created ahead of time by the programmer or coder (words we use interchangeably) and then the code is 'run' without any more changes to the code as it runs.

The distinction here is between creating the code for a microwave to work (read our blog post at www.letsstartcoding.com/what-is-coding !) and someone using a microwave by pushing the buttons. Creating the sequence the microwave goes through is coding; pressing buttons to initiate the sequence is not.

Coding is creative by definition and there are very few 'right' or 'wrong' programs - coding is about making whatever you want to make! There are, of course, rules about *how* the steps you create are 'understood' by the computer. These rules, often called *syntax* are similar to punctuation in a book. Can you imagine reading this document without any punctuation, spacing, capitalization, or headings? If you could do it, it would be a testament to how great of a computer your brain is!

Most computers cannot interpret any kind of 'meaning' outside of their strict syntax rules. This is frustrating to many coders of all experience levels, but it also gives your child complete control over what the code can do if they 'play by the rules' of syntax.

Let's Start Coding kits use typed coding, as opposed to drag-and-drop coding that kids may be familiar with through Scratch or Blockly or Hour of Code. While block coding does mean the definition of coding, text-based or typed coding is the only way that real coding professionals create the programs that power the apps, websites, and devices we use. Kids can learn typed coding and the skills they learn by doing that will offer them more flexibility and creativity than other methods of coding.

HOW DO THESE SKILLS APPLY?

When children are exposed to coding through a Let's Start Coding kit, their minds are opened up to seeing how the world around them may be controlled or interacting with code. A few examples that can directly relate to Let's Start Coding projects:

- How is the tone of an alarm clock set? Could it be a higher pitch or a lower pitch?
- How does a thermostat trigger the heater in our house to run?
- What is happening when I press the 'volume up' or 'volume down' buttons on a smart phone?
- How does an ambulance siren work? What might the code look like behind that sound?
- How might a home security system detect someone?

While kids aren't building real-world tools like these when using their kit, they are modeling the processes and thinking that goes into them. Learning that code isn't scary, that 'failures' should be treated like puzzles to solve, and how the computer 'understands' logic are all fundamental skills that kids can use in a variety of future careers.

Particularly in the late-elementary to middle-school age range, there should be more focus on confidence-building, problem-solving, and persistence than the specifics of any single coding language or project. As kids grow older and find a 'niche' that interests them, these skills will be well-practiced!

WHAT WILL MY CHILD DO WITH THIS KIT?

A Let's Start Coding kit is a way for beginners to interact with and create code with hands-on electronics. It is both a toy and a learning device - edutainment, if you will - that encourages them to imagine and create their own fun with code.

Every kit is accompanied by free online projects that guide children through basic coding. These include different ways to blink lights, beep the speaker, react to button presses, and more. **There are two modes of play with Let's Start Coding: making the projects through code and playing with the circuit board after it has been coded.** For example, a child may write a program that makes each button on the Code Piano beep a different tone from the speaker when pressed. The act of creating and modifying that code is valuable learning and play time. *Then*, the child will have coded their own custom piano and they can play around with it, making songs and sound effects! Both modes are the 'right' way to use the kit and each engages their creative brains.

If you have an Ultimate or Base Kit (the green 'toolbox' packaging), children will plug in individual electronic components to ports on the main circuit board called Maker Board. Code is loaded onto Maker Board and that code dictates what happens to the components plugged in to it. For example, kids may plug in a button to port 2 and a speaker to port 10. Then in the coding software on their computer, they may write some code so that 'if button 2 is pressed, play a tone of 550 hertz on port 10'. After this code is loaded onto Maker Board, pressing the button plugged into port 2 will result in a 'beep!' from the speaker!

If you have a Code Car, Code Rocket, Code Piano, or Code Speaker, all of the electronic components like LED lights, speaker, and buttons are already attached to the 'brain' of the circuit board. Kids won't have to plug in or rearrange the components; most of the ingenuity here is in the kids' coding via the free downloaded software. Using Code Car as an example, kids will first learn how to turn on one of the LED lights that is in the 'headlight' position on the Car. Next, they'll be challenged to turn on a different light - the 'brakelight'. As kids progress through the projects, including 'police lights', 'siren', and 'backup alarm', what you will see is a series of different light and sound patterns. Behind the scenes, though, kids are learning a ton about how to create these with code - blinking a light versus fading one up and down is a very different task in code that requires knowledge and practice!

As you see what your child is creating with code, you can ask some questions about each project that will reveal how much they are actually learning. Asking something like "what was the most challenging thing about building this?" or "can you explain this code to me?" will reveal just how much kids are coding to make a seemingly simple outcome.

HOW CAN I HELP MY CHILD?

If you're not a coder, your child may surpass your understanding of programming within an hour of playing with their Let's Start Coding kit. This can feel a little intimidating, particularly when your child asks for help! **Fear not: there are some tried-and-true methods that will help you help your child without learning code.**

1) Tech troubles

Technology troubles are not always directly related to code. Especially when they are first getting started, kids are susceptible (like the rest of us!) to simple tech issues that may frustrate them or test their patience. If a child comes to you with an exclamation of 'it doesn't work!', the first things to check are the basics:

- Is the circuit board plugged in to the USB port?
- Is the steady green power light on?
- Is something else possibly at fault, like the internet connection has been lost?

2) Process (and patience!)

Let's Start Coding projects are designed to be self-paced and include a ton of support materials. Between the example programs, the embedded video walkthroughs, and the instructional cards in the kit, there are many resources to help kids learn code. Depending on the age and learning style of your child, though, those resources may not get utilized before frustration sets in. This is where an adult can help.

Ultimate and Base Kit (green box) parents:

If your child is having trouble getting an example program to work and the tech tips from step 1 are not fixing the issue, double-check that the circuit board connections are correct. Plugging a component into port 9 when you meant to plug it into port 10 is very easy to do and requires some care. The 'hardware guide' on every project page shows how the components should connect.

If your child is having trouble with a particular code challenge, encourage them to do the following:

- Watch the walkthrough video on the page where they are stuck. It will explain the example code in a line-by-line fashion.
- If they have already started working on a challenge, encourage them to start taking baby-steps backward, undoing their changes and checking if the code works along the way.
- If the code is not working how the child wants it to, encourage them to write down, with pencil and paper, what the program should do. Then they can use that plan instead of trying a series of things that may or may not push them in the right direction. This technique helps programmers of all ages think more clearly!
 - An example here might be that a child writes “I want 2 lights to blink on and off at the same time, but instead they are blinking back and forth”. Now a child has a clear idea of what they want to do and can start to think about *how* to accomplish the goal instead of being stuck on the idea that “it doesn’t work”. Even if this method doesn’t solve the issue, now you have a much better question to pose to the Let’s Start Coding customer support team! Our email address is info@letsstartcoding.com and we are happy to accept coding questions, though they need to be sent from someone 13 or older.

Every Let’s Start Coding project starts with working code and the code editor has a ‘Restore’ button that will reset whatever code your child has written *back* to the working code, so a ‘safety net’ is always in sight. Kids can undo every change that they have made to the example program and ensure they get back to something that works, starting their experiment over again.

Let’s Start Coding projects are ‘sequenced’ meaning that they do get more complex the further along a child goes. It’s totally O.K. for a child to jump around to the projects that interest them, but if they are stuck on a challenge, encourage them to look at earlier projects for clues or to watch walkthrough videos of earlier projects. This method may reveal the answer.

3) Help define the pace (or lack thereof!)

Let’s Start Coding kits are self-paced when used at home, and there isn’t a “right” or “wrong” speed to go through a project. Sometimes, kids think the goal of the kit is to “complete”, which often means build every circuit, upload every example, and move on. They wouldn’t learn much that way!

As a parent, you can encourage kids to keep playing with a project that has caught their curiosity. This may mean they tinker with blinking lights for three hours, which is great! Remember, as kids are playing and experimenting, they’re learning and strengthening their

creative coding skills. Engagement is more important than ‘project count’ or ‘number of lines of code written’.

4) Persistence

One of the key ways you can help your child is encouraging them to keep engaging with code and their Let’s Start Coding kit. Because coding is new to most kids with a kit, it may not be immediately obvious to them (or you!) what they can build. Encourage kids to spend some time coding each week as they get started, whether that is 30 minutes a day, one hour on a weekend, or something else. Celebrate their new creations and ask them to tell you how their code works. Coding takes practice and the skills kids learn build upon each other, so persistence is key.

TIP: Don’t take over the keyboard! Although it is tempting to jump into the ‘drivers seat’ when you think you see a typing error on the screen, we have seen time and again that when an adult takes over the keyboard to help solve a problem, kids ‘check out’ and now the bug becomes the adult’s problem!

HOW CAN I ANSWER THEIR QUESTIONS IF I DON'T KNOW THE ANSWERS?

Even if you've followed the methods laid out in the last chapter, your child may have coding questions that you have no idea how to answer. Don't panic! Even if you're used to having the answer, there are ways you can help your kids without learning a ton of coding knowledge.

Here are some questions - and exclamations - that kids may ask and how you might respond.

"It doesn't work and I don't know why!"

- *Has this exact program ever worked in the past?* If yes, that may point to a tech troubleshooting issue.
- *Have you created a program like this, but a simpler version?* If yes, stepping backward and 'undoing' changes may reveal where the issue lies.
- *Are there any example programs that you think are similar to this?* If yes, kids should start there, reviewing the Let's Start Coding example and building on it. If no, it's possible that what the child is wanting to do isn't possible with these basic kits. That's O.K. and means your child is still learning. Instead, *what would be a prototype or simpler version of this that you can build?*

"I want it to do X but I can't. Why not?"

- *Have you already done something similar to this?* If yes, the issue may be in finding a pattern that the child can replicate in their program. For example, if a child is blinking two LED lights and wants to blink three lights, they should look for patterns in the code that will enable that change. The code walkthrough videos are a good resource for this type of task.
- *What are the pieces required for the project you want to do? Are those included in the kit?* Imagination is a powerful thing, and kids may want to do something outside of what is contained in the kit. Having the Code Car actually drive across the floor is a good example. Asking the child to break down the project into its component parts helps them think about what is required to solve problems in the future.
 - Keep in mind that 'component parts' can also apply to the coding concepts here. If a child wants to control the volume on the speaker, the component parts would be a coding concept related to volume control. They can look through the projects for an example of that (though they will not find an answer - the speaker only has one volume!)

“How can I make my kit do X?”

- *Great question! How can we determine what parts are needed?* Coding is often an act of breaking a final product into its parts, then building each part to reach the end - much engineering and science could be described this way. By helping your child think through the different parts that make up their ideal finished project, there are four conclusions you may reach:
 - 1) This project is partially possible, but not completely. This is O.K.! Building a ‘prototype’ or a trimmed down version of a project is still fun and creative.
 - 2) This project is completely possible. Excellent! Now you have a list of its parts and can start building and testing them one at a time. Note that ‘parts’ can also be coding concepts, like a ‘a light that blinks every 3 seconds’, not just physical components.
 - 3) This project is not possible. No problem! You can either jump to conclusion #1 above and think about a prototype, or move on with the knowledge that you have now understood the challenge and *could* build it if the opportunity arises.
 - 4) We’re still not sure if this project is possible. If you have broken down the different parts of the project, but you and your child aren’t certain if they are possible, then treat each component part as its own question: *“How can I make my kit do X, which is required for Y?”*.

You may notice that these questions are nearly variations of the same thing. This is normal because coding is a whole new world and children are learning its boundaries. If a child asked “Why can’t my remote control truck drive on the ceiling?”, you’d feel more equipped to answer that based on your understanding of gravity. In the world of coding, different rules apply, so part of the learning process is learning the boundaries of the code and the circuits. Asking good questions *in response to your child’s questions* will help them learn.

Once, the Let’s Start Coding team was leading a summer coding camp when one of the campers came up with an idea to create BB-8, the Star Wars character. If you haven’t seen the film, this is a spherical robot that rolls across the ground with a ‘head’ that hovers above it - complicated, to say the least! By asking this camper about what parts would be needed to make a BB8, he realized it wasn’t something he could build with his kit. Instead, he used his skills to code up an LED light like BB-8’s “eye” and then used the speaker to create some robotic beeps-and-boops like BB-8. With a creatively crafted drawing to overlay on top of the circuit, our camper created his BB-8 prototype and was thrilled to show it off.

Through this process of exploring through questions, you’re modeling to kids problem-solving behavior and methods. Some parents and teachers have even evolved these conversations into checklists for kids to use to internalize these questions and answers. Learning and conversing with your child this way can actually be great fun and amplifies your successes when you reach them by thinking through everything together.

WHAT OUTCOMES SHOULD I EXPECT?

Our goal at Let's Start Coding is to get kids excited and engaged with coding through hands-on experience. They will not likely build an app for the Apple store or make your car drive itself by using one of the Let's Start Coding kits, but they will make huge leaps in their understanding and confidence around coding and technology. Here are some of the outcomes you can expect.

“Hey! Look at this!”

Coding something on your own is a challenge, so making it work is a huge accomplishment! When your kid shows off their newest code creation, ask them:

- *Cool! Where did you start to get to this creation?* Maybe your kid just uploaded an example program and now they are playing with it. Maybe they modified the program extensively with their knowledge. Either way, they'll understand the process of how they got to this result and can likely explain it in detail that will surprise you.
- *What were the challenges that you had to solve to do this?* Even experienced coders make mistakes on simple tasks - it really is inherent to the coding experience. Getting kids to remember and describe some of the hurdles they overcame not only helps solidify their knowledge, but helps them persevere next time.
- *What are some other ways you could remix or add to this?* Children often respond well to small prompts that spark their imagination about what else they are capable of doing. For example, if a child shows you three lights blinking on their project, you may say “*How would you blink FOUR lights on this project?*”. Even if you have no idea how to answer the question you asked, you may ignite their excitement to go and try.

Going off script

Writing code from a blank screen is tough. Although we often see in movies and TV someone sit down at a computer and start to create hundreds of lines of computer code, that's not very realistic.

As your child gets better and better at coding, you will find their ideas going further and further 'off script', incorporating more components and more lines of code and fewer small changes to the example program. This is a good sign! Kids generating their own ideas means they are learning the 'rules of the game' of coding and thinking about how to push its boundaries. Even if not all their projects are a success, they are venturing into new coding territory on their own.

Real-world ties

The Let's Start Coding kits make coding tangible and as kids play, they are likely to start making real-world connections themselves. Using the Maker Screen from the Ultimate Kit has prompted kids to joke about 'hacking' the gas price signs to make gas only \$0.01 per gallon! Traffic lights, alarm clocks, microwaves, TV remotes, thermometers, car remotes, and all kinds of other devices may get kids thinking "I wonder how the code inside that works?". This is a great outcome that keeps their curiosity lively and may inspire their next project (or their entire career!).

Asking for more

Once it 'clicks' to children that coding is done by normal humans just like them, you'll likely hear them asking about all kinds of coding opportunities. Video games and robotics are two very popular trends with kids, so don't be surprised if they suddenly want to 'hack' Minecraft or Roblox or join the school robotics team. It's up to you where to take them next, but opening the door with coding reveals many opportunities to kids.

TIES TO FORMAL EDUCATION

Although Let's Start Coding kits don't claim to be a strict academic curriculum, they certainly reflect some of the hard and soft skills that kids may learn in school. Here are a few topics that you may want to research further and observe how they are reflected in your child's coding activities.

Scientific Method

"A method consisting in systematic observation, measurement, and experiment, and the formulation, testing, and modification of hypotheses."

Engineering Method

"A systematic approach used to reach the desired solution to a problem. There are six steps (or phases): idea, concept, planning, design, development, and launch from problem definition to desired result."

Variables

"A quantity that may change within the context of a mathematical problem or experiment (or in our case, a computer program)."

Electronics Concepts

In particular, power, ground, and signals and how a circuit must be connected for electricity to flow through it correctly.

Troubleshooting, Logic, Sequencing

By playing with their kit, kids will learn the computer's strict logical rules and become attuned to how they must sequence their commands to get the desired program output.

Now, Let's Start Coding! Remember, if you have any questions or want to share your feedback or progress with us, we'd love to hear from you at info@letsstartcoding.com or team@letsstartcoding.com